TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Systém pro modelování a predikci sportovních výsledků

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika
Studijní obor: 1802T007 – Informační technologie

Autor práce: **Bc. Václav Slavík**
Vedoucí práce: Doc. RNDr. Miroslav Koucký, CSc.

Liberec 2016/17

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

System for modeling and predicting sports results

Master thesis

Study programme: N2612 – Electrotechnology and informatics
Study branch: 1802T007 – Information technology

Author: **Bc. Václav Slavík**
Supervisor: Doc. RNDr. Miroslav Koucký, CSc.

Liberec 2016/17

Technická univerzita v Liberci
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2016/2017**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Václav Slavík**
Osobní číslo:	**M15000185**
Studijní program:	**N2612 Elektrotechnika a informatika**
Studijní obor:	**Informační technologie**
Název tématu:	**Systém pro modelování a predikci sportovních výsledků**
Zadávající katedra:	**Ústav nových technologií a aplikované informatiky**

Zásady pro vypracování:

Cílem je vytvořit dostatečně obecný statistický model pro predikci výsledků v cyklistice. Model bude zahrnovat specifika nejenom disciplín, ale také typu soutěže, profilu tratě a bude využívat relevantní databáze (výsledky, profily tratí atd.). Praktická část bude zaměřena na softwarovou implementaci zmíněného statistického modelu.

1. Vytvořte statistické modely pro predikci cyklistických výsledků.
2. Navrhněte databázi pro cyklistické statistiky.
3. Implementujte grafické rozhraní k zobrazení výsledků z modelů a vyhodnocení jejich úspěšnosti.
4. Vyhodnoťte výsledky statistických modelů.

Rozsah grafických prací: **dle potřeby**

Rozsah pracovní zprávy: **40 - 50 stran**

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] Java Platform, Standard Edition 7 API Specification [online]. [cit. 2016-10-18]. Dostupné z: https://docs.oracle.com/javase/7/docs/api/
[2] H2 Database Engine [online]. [cit. 2016-10-18]. Dostupné z: http://www.h2database.com/html/main.html
[3] WEST, Mike a Jeff HARRISON. Bayesian forecasting and dynamic models. New York: Springer-Verlag, 1989. Springer series in statistics. ISBN 0-387-97025-8.
[4] COX, D.R a SNELL, E.J. 1971. Applied Statistics. Methuen, London.
[5] ANDERSON, T. W. An introduction to multivariate statistical analysis. 3rd ed. Hoboken, N.J.: Wiley-Interscience, c2003. Wiley series in probability and statistics. ISBN 0-471-36091-0.

Vedoucí diplomové práce: **doc. RNDr. Miroslav Koucký, CSc.**
Katedra aplikované matematiky

Datum zadání diplomové práce: **20. října 2016**

Termín odevzdání diplomové práce: **15. května 2017**

Technická univerzita v Liberci

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

prof. Dr. Ing. Jiří Maryška, CSc.
vedoucí ústavu

V Liberci dne 20. října 2016

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 4.9.2017

Podpis:

Abstrakt

Tato práce se zabývá predikcí a modelováním cyklistických výsledků. Důraz je kladen především na obecnost modelů a jejich použitelnost i v jiných sportovních odvětvích. Tyto modely jsou vytvořeny především pro individuální sporty, které jsou klasifikovány pomocí času. Vytvořené modely jsou založeny především na umístění závodníků, jejich dosažených časech, ale reflektují i výškový profil závodu, jeho úroveň či způsob startu. Výstupem celé práce je aplikace, která poskytuje rozhraní vhodné pro vývoj matematických modelů a zároveň implementuje statistiky z cyklistiky, biatlonu, běžeckého lyžování i závodů F1. Stejně podstatnou součástí jsou i vytvořené modely, které umožňují běžnému uživateli, respektive bookmakerovi, predikovat výsledky následujících závodů a pomocí grafického rozhraní mu zobrazit velkou škálu různých výsledků. Aplikace může být vhodným doplňkem pro sázkové kanceláře při vypisování kurzů.

Klíčová slova: predikce sportovních výsledků, pravděpodobnost, cyklistika, statistika

Abstract

This work focuses on prediction and modeling of cycling results. Emphasis is placed on the generality of models and on their applicability in other types of sports as well. The models are created primarily for individual sports that are classified by time. The models obtained are based primarily on the position of the racers, their times achieved, but also reflect the elevation profile of the race, its level or the method of starting. The output of this work is an application which provides an interface suitable for the development of mathematical models and also implements statistics from cycling, biathlon, cross-country skiing and F1 races. An equally important part of the thesis are models that allow a regular user or a bookmaker to predict the results of the upcoming races and use the graphical interface to display a wide variety of different results. The application can be used as a supplement for bookmakers in order to help them create the odds.

Keywords: predicting sports results, probability, cycling, statistics

Poděkování

Rád bych poděkoval vedoucímu diplomové práce doc. RNDr. Miroslavu Kouckému, CSc. za jeho rady, a čas, který mi věnoval při řešení dané problematiky. Dále děkuji i sázkové kanceláři Tipsport, za nastínění aktuálního stavu využití predikce v sázkovém průmyslu. V neposlední řadě děkuji i své rodině, za jejich podporu po celou dobu mého studia.

Obsah

Seznam zkratek

ER	Entity-Relationship
HTML	HyperText Markup Language
JSON	JavaScript Object Notation
RGB	Red, Green, Blue (aditivní barevný model)
SQL	Structured Query Language
UCI	Union Cycliste Internationale
URL	Uniform Resource Locator
XML	Extensible Markup Language

Seznam značení

$b_{start,j}$: způsob startovní procedury j-tého závodu (0 = individuální, 1 = hromadný)

$b_{stage,j}$: označuje, zda je j-tý závod součástí etapového závodu (0 = ne, 1 = ano)

C : součinitel odporu vzduchu

C_R : součinitel valivého odporu

$d(c_1, c_2)$: vzdálenost vektorů (c_1, c_2) v gamutu Lab

d_j : datum j-tého závodu vyjádřena počtem milisekund po 1.1.1970

e_j : úroveň j-tého závodu

$f_e(x, y, x_{last}, k)$: funkce detekující hranu výškového profilu

g : tíhové zrychlení

$h(x)$: funkce vyjadřující nadmořskou výšku ve vzdálenosti x od začátku závodu

h_{diff} : rozdíl mezi nejvyšším a nejnižším bodem závodu

$h_{height}(y)$: funkce pro vypočtení nadmořské výšky z obrázku

h_{max} : největší převýšení na jediném stoupání

h_{total} : celkové převýšení závodu

k_{add} : počet po sobě jdoucích pixelů, které musí být rozpoznány

k_{color} : prah pro vyhledávání barev

k_{detect} : počet detekčních pixelů nutných k správné identifikaci hledaného pixelu

k_{height} : výška obrázku uvedená v pixelech

$k_{multiple}$: pravidlo násobku mezi osou x a y při detekci

$k_{relative}$: maximální možná relativní chyba mezi metodou shora a zdola

k_{width} : šířka obrázku uvedená v pixelech

l_j : délka j-tého závodu

m : hmotnost

m_{eff} : hmota podléhající zrychlení

$p_{up}(x)$: funkce pro vypočtení výšky pixelu metodou zdola nahoru v x-tém sloupci

$p_{down}(x)$: funkce pro vypočtení výšky pixelu metodou shora dolů v x-tém sloupci

$p_i(x)$: pravděpodobnostní funkce pro i-tého závodníka v závislosti na jeho umístění

$P_i(x_1 \leq X \leq x_2)$: pravděpodobnost, že se i-tý závodník umístí mezi x_1 a x_2 místem

$p'_i(x)$: pravděpodobnostní funkce nezávislého rozdělení mezi závodníky pro i-tého závodníka

$p''_i(x)$: normalizovaná funkce $p'_i(x)$

$p_{s,j}$: součet úspěšně predikovaných pravděpodobností

$p_{r,j}$: vážený součet úspěšně predikovaných pravděpodobností

P : výkon podávaný cyklistou

P_A : výkon nutný k překonání odporu vzduchu

P_S : výkon nutný k překonání sklonu vozovky (gravitační přitažlivosti)

P_R : výkon nutný k překonání valivého odporu

P_B : výkon nutný k překonání odporu nerovností vozovky

P_{acc} : výkon plynoucí ze zrychlení

$r_{top,i,j}$: nejlepší dělené umístění i-tého závodníka v j-tém závodě

$r_{down,i,j}$: nejhorší dělené umístění i-tého závodníka v j-tém závodě

$r_{weight}(x,i,j)$: váha x-tého umístění i-tého závodníka v j-tém závodě pro případ dělených umístění

S_j : startovní listina j-tého závodu

s_l : sklon vozovky

$t_j(CS, m, \rho, P)$: funkce pro výpočet času v j-tém závodě (cyklistiky) při působení se stálým výkonem

$t_{total,i,j}$: celkový čas i-tého závodníka v j-tém závodě

$t_{loss,i,j}$: časová ztráta i-tého závodníka v j-tém závodě na vítěze závodu

T_i : množina x-ových souřadnic pixelů pro i-tý interval

v : rychlost

$v(CS, \rho, m, P, s_l)$: výpočet rychlosti z požadovaných parametrů pomocí řešení kubické rovnice

$w_{date,j}(k)$: funkce pro výpočet váhy data vzhledem k predikovanému závodu

w_j : váha j-tého závodu vzhledem k predikovanému závodu

$w_{level,j}(k)$: funkce pro výpočet váhy s ohledem na úroveň závodu

$w_{length,j}$: váha j-tého závodu vzhledem k délce predikovaného závodu

$w_{length2,j}$: váha j-tého závodu vzhledem k délce predikovaného závodu

$w_{power,j}$: váha j-tého závodu odvozena od výkonu cyklistů

$w_{profile,j}$: váha j-tého závodu vzhledem k převýšení a délce predikovaného závodu

$w_{start,j}(k)$: funkce pro výpočet váhy s ohledem na způsob startu

Z : množina závodů

ρ : hustota vzduchu

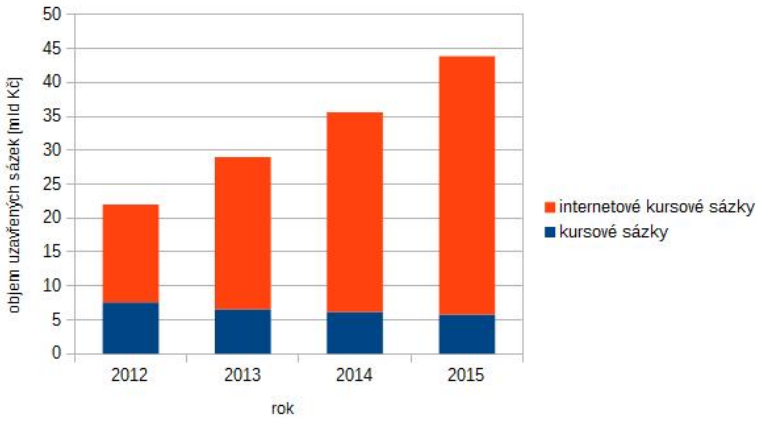
ϕ : aritmetická odchylka mezi součty pravděpodobností v jednotlivých intervalech

Seznam tabulek

Seznam obrázků

1 Úvod

Ve své diplomové práci jsem se rozhodl zaměřit na predikci sportovních výsledků v silniční cyklistice. Stanovení pravděpodobností na rozlišné události představuje poměrně zajímavou aplikaci matematické statistiky. Potřeba podobných aplikací je zřejmá s ohledem na neustále rostoucí objem kurzových sázek. Jen v České republice v roce 2015 činil obrat uzavřených kursových sázek 43,8 mld. Kč. Na obrázku 1.1 vidíme celkový obrat uzavřených kursových sázek v letech 2012-2015, který neustále roste. Zajímavý je především nárůst internetového sázení, naopak sázení na pobočkách spíše stagnuje. Lze rovněž předpokládat, že nastolený trend bude i nadále pokračovat. Obrat kursového sázení v roce 2015 činil 28,8% z celkového podílu hazardních her v ČR a díky zvyšujícímu se obratu internetového sázení se i tento podíl za poslední roky pravidelně zvyšuje [13].



Obrázek 1.1: Celkový obrat uzavřených sázek v ČR

První statistické modely se objevily již v roce 1951, kdy Moroney přišel s myšlenkou, že počet vstřelených gólů ve fotbale vyplývá z vysokého počtu útočných šancí daného týmu, a proto navrhl využít Poissonovo rozdělení. Na jeho myšlenku navázal Ma-

her, který model aplikoval na několika nejvyšších anglických fotbalových soutěžích. Zároveň přidal koeficient obrany a útoku pro domácí i hostující tým [10]. Za dalším zajímavým posunem stál Dixon s Colesem, kteří korigují Poissonovo rozdělení pro některé počty vstřelených branek [11]. Všechny zmíněné modely lze nasadit nejen ve fotbale, ale i dalších sportech, které mohou využít stejného principu. Jedná se např. o hokej, házenou, basketbal. Zejména tedy sporty, které jsou omezeny časem a rozhodují v nich vstřelené branky.

Na poli individuálních disciplín mnoho zveřejněných modelů neexistuje. Výjimku představuje například biatlonový model, který přichází s myšlenkou predikce na základě umístění závodníků v předchozích závodech, nebo podle jejich dosažených časů [12]. Použito je i rozdělení na více nezávislých částí, ke kterým přistupujeme odděleně a na závěr se časy sečtou. Jednotlivé složky jsou vyjádřeny pomocí distribučních funkcí, výsledný čas se tedy spočte pomocí konvoluce. Nevýhodu vytvořeného modelu představuje především jeho úzké spojení s biatlonem. Často se lze setkat s články, které predikují vývoj nejlepších časů atletických či plaveckých disciplín. Tyto sporty se vyznačují především velmi obdobnými podmínkami na různých místech světa, navíc se jedná o predikci rekordních výsledků, kde je zřejmé, že funkce bude nerostoucí [1]. Nicméně pro podobné sporty by jistě bylo možné obdobným způsobem predikovat i výsledky vybraných závodníků. V cyklistice se však podmínky velmi liší a nemáme k dispozici dostatek informací, aby bylo možné obdobným způsobem predikovat čas závodníků.

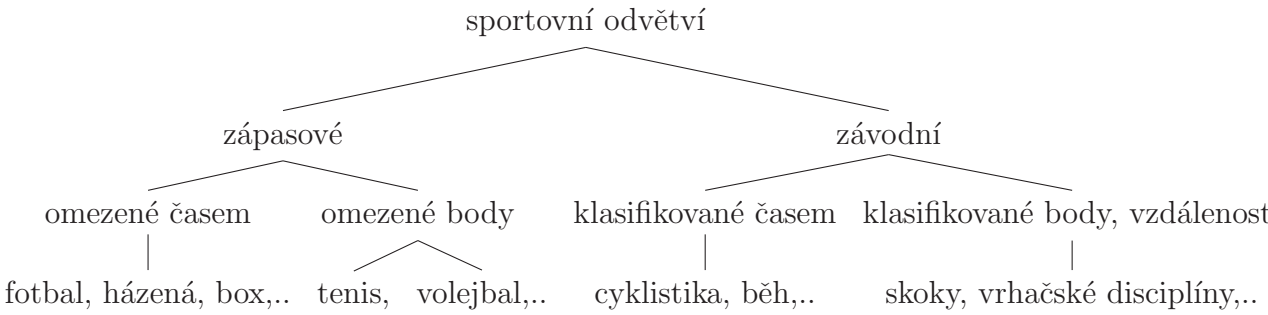
Aplikace je programována pomocí programovacího jazyka java, který je pro desktopovou aplikaci vhodný. Ve srovnání s jazykem C výkonnostně sice mírně zaostává, ale z programátorského hlediska obsahuje velké množství frameworků a výkonnostní rozdíl nepředstavuje zásadní problém. Aplikace využívá návrhového vzoru MVC, který ji rozdělí na 3 samostatně fungující části [19]. Základním motivem pro použití tohoto návrhového vzoru je oddělení programového modelu od uživatelského rozhraní, čímž se zvýší jeho přehlednost, ale i znovupoužitelnost. Model (M) obsahuje veškerou logiku, výpočty i práci s databází. Pohled (V) se stará jen o zobrazení výsledků uživateli. Model a pohled jsou úplně odděleny a komunikaci mezi nimi zajišťuje controller (C). Grafické rozhraní bude tvořeno s pomocí knihovny $JavaFX$, která je připravena na použití spolu s návrhovým vzorem MVC. Model bude vy-

tvořen pomocí běžných tříd, contoller pak jen implementuje rozhraní *Initializable* a zároveň umí komunikovat s pohledem, který představuje jazyk FXML. FXML je jazyk založený na xml a umožňuje vytvořit pohled v tomto formátu. Mimo psaní pohledu v xml podobě existuje i návrhář Scene Builder, který výrazně usnadní práci [20]. Pro ukládání dat bude využita databáze H2, která je založena na relačním modelu a může být používána bez instalace databázového serveru, což může být výhodné z hlediska přenositelnosti.

2 Specifika sportovních odvětví

Sportovní odvětví lze rozdělit na mnoho kategorií. Nás zajímá především hledisko predikce výsledků. Z tohoto pohledu nemusí být ani tak důležité, jestli se jedná o týmový nebo individuální sport, jelikož na výsledky vybraného týmu lze pohlížet podobně jako na výsledky jediného závodníka.

Na následujícím grafu je znázorněno rozdělení s ohledem na způsob predikce. Sportovní odvětví, která byla označena jako zápasová představují sporty, kde se utkají dva soutěžící týmy, či jednotlivci, a obvykle končí po předepsaném čase, nebo určitém počtu bodů.



Závodů se naopak může účastnit libovolný počet závodníků a vyhodnocují se na základě času, či dosažených bodů. Ve všech odvětvích, která byla označena jako závodní, lze určit umístění závodníků a na jeho základě i predikovat výsledky nadcházejících závodů. Rozdíl mezi klasifikací pomocí času a bodů (vzdálenosti) je zejména v uspořádání, kde v případě časové klasifikace vítězí závodník s nejnižším časem. V bodovém hodnocení naopak většinou vítězí závodník s vyšším počtem získaných bodů.

Diplomová práce je zaměřena zejména na cyklistiku a tak modely budou vytvořeny tak, aby mohly být implementovány ve všech závodních sportech, zejména pak těmi, které využívají časové klasifikace. Naopak kategorie zápasových sportů je s modely v této práci neslučitelná.

2.1 Silniční cyklistika

Silniční cyklistika patří mezi individuální sporty. Závodníci přesto často spolupracují a to zejména v rámci svého týmu. V rámci vybraného týmu mají závodníci obvykle přiděleny týmové pokyny a často odvádí práci pro lídra týmu, na úkor osobních ambicí. Mezi různými odnožemi profesionální cyklistiky je právě silniční královnou tohoto sportu, což se odráží v počtu profesionálních sportovců a jejich zázemí.

Používají se silniční kola, která jsou specifická především úzkými plášti, respektive galuskami. Pro všechna kola platí pravidlo minimální hmotnosti, závodní stroj nesmí být lehčí než 6,8 kg [2]. Prakticky všechny profesionální závodníci mají kola s hmotností velmi blízkou uvedenému limitu a tak můžeme pracovat při predikci výsledků výhradně s touto hmotností.

2.1.1 Jednorázové závody

Jednorázovým závodem, nebo také jednodenním závodem, rozumíme takový závod, který se startuje hromadně. Závod končí v předem stanoveném místě a po jeho průjezdu jsou závodníci klasifikováni. Jediným rozhodujícím kritériem pro vyhodnocení výsledků je čas, pokud nedojde k porušení pravidel a tím i k diskvalifikaci.

2.1.2 Časovka

Časovka, či jízda proti chronometru, je závod s individuálním startem. Trasa závodu je opět pevně dána. Startující zároveň nesmí využívat blízké jízdy za svým soupeřem (jízdu v háku). Váhový limit je stejný jako pro jiné závody, nicméně je povolené používat speciální vybavení, které je pro hromadné závody zakázané[2]. Zejména se jedná o aerodynamické helmy, nástavce na řidítka a případně i disková kola.

2.1.3 Etapové závody

Etapový závod se skládá z několika etap. Etapa může být jednorázovým závodem s hromadným startem, ale i týmovou či individuální časovkou. Obvykle se koná za jeden den pouze jedna etapa. Závody se výrazně liší i svou délkou, na českém území

se například koná 4-etapový Czech Cycling Tour. Nejdelší ale i nejznámější jsou třítýdenní etapové závody Tour de France, Giro d' Italia a Vuelta.

2.1.4 Problematika predikce výsledků

V silniční cyklistice nastává několik problémů z pohledu predikce výsledků, se kterými se v jiných sportech nesetkáme, nebo jen v malé míře. Závodníkům se, podobně jako v jiných odvětvích, v průběhu sezóny podstatně mění jejich kondice. Oproti ostatním sportům je však velmi složité odhadnout závodníkovu formu jednoduchým pohledem na výsledky. Výsledek závodu totiž může výrazně ovlivnit defekt v nevhodnou chvíli, ale zejména pozice závodníka v týmu. Pokud se jezdec v závodě vyskytuje v pozici domestika, tak i přes jeho možnou velice dobrou formu, výsledek nebude odpovídat skutečným možnostem závodníka. Někteří jezdci také směřují svou formu pouze k určitému závodu. Zejména se jedná o závodníky, kteří se specializují na třítýdenní etapové závody.

3 Výkon cyklisty

Výkon cyklisty hraje v silniční cyklistice zásadní roli. Na jeho základě se budeme snažit v dalších kapitolách zjistit náročnost terénu a jeho vhodnost pro určité typy závodníků. Při výpočtu výkonu zanedbáme ztráty při přenosu síly. Celkový výkon se skládá z několika základních složek a je dán jako

$$P = P_A + P_S + P_R + P_B + P_{acc}, \tag{3.1}$$

kde jednotlivé složky vyjadřují potřebný výkon k překonání odporu vduchu (P_A), odporu sklonu vozovky (P_S), valivého odporu (P_R), odporu nerovností (P_B) a zrychlení (P_{acc})[4].

Odpor vzduchu hraje v cyklistice zcela zásadní roli. Průměrné rychlosti se během závodů pohybují okolo 40 kmh^{-1}, takže se již nejedná o laminární, ale o turbulentní proudění. Složku tohoto výkonu spočteme jako

$$P_A = \frac{1}{2} C v^3 S \rho, \tag{3.2}$$

kde C je součinitel odporu vzduchu, S představuje průřez cyklisty, v jeho rychlost a hustota vzduchu je pak označena symbolem ρ.

P_{acc} označuje výkon potřebný ke zrychlení, nebo zpomalení, a je roven $P_{acc} = m_{eff} a v$, kde a představuje zrychlení ze závodníkovi rychlosti v a m_{eff} je hmota, která tomuto zrychlení podléhá.

Výkon, vynaložený na překonání gravitační síly, hraje roli zejména v kopcích, kde ho závodník musí překonávat, respektive jej využije při sjezdu, kdy ho naopak zrychluje a je dán

$$P_S = m g s_l v, s_l \in \langle -1, 1 \rangle, \tag{3.3}$$

kde m je hmotnost závodníka s kolem, s_l představuje okamžitý sklon vozovky. Průměrný sklon vozovky pak vypočteme jako podíl mezi nastoupanou výškou a ujetou vzdáleností.

Výkon, potřebný k překonání valivého odporu, spočteme jako $P_R = mgC_Rv$, kde m je opět hmotnost závodníka včetně jeho bicyklu, g gravitační zrychlení a v závodníkova rychlost. C_R závisí na tlaku a materiálu pneumatik, v silniční cyklistice se tato hodnota pohybuje okolo 0.003. Odpor, vznikající nerovnostmi povrchu, bývá ve většině silničních závodů velmi zanedbatelný. Výjimku by tvořily závody s pave sektory. Výpočet by však byl zřejmě velmi náročný a nepřesný, jelikož velmi závisí na drobných detailech na trase a také na materiálu a vzorku pneumatik. Z těchto důvodu se tento faktor zanedbává a ani neexistuje přesně stanovený vzorec pro jeho výpočet [4].

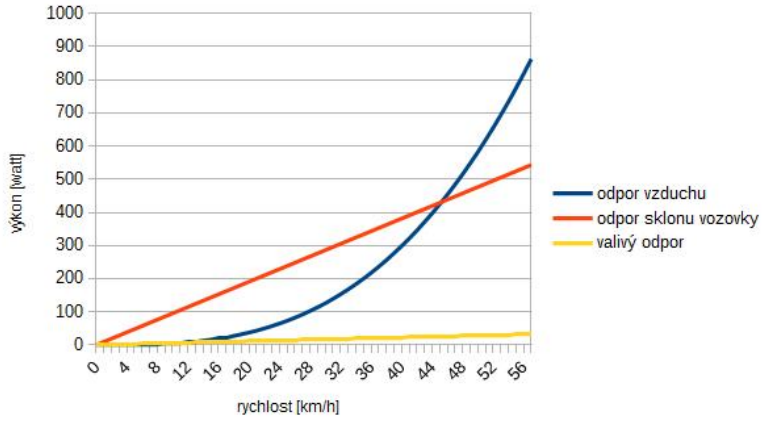
Po vyjádření všech složek a dosazení, získáme výsledný výkon cyklisty

$$P = \{\frac{1}{2}Cv^2S\rho + mg(s_l + 0,003) + m_{eff}a\}v. \tag{3.4}$$

Často se setkáme se situací, kdy je neznámou rychlost závodníka, naopak známe výkon a všechny ostatní koeficienty v rovnici. Získáme tedy kubickou rovnici ve tvaru

$$CS\rho v^3 + (mg(s_l + 0,003) + m_{eff}a)v - 2P = 0. \tag{3.5}$$

Jelikož nenastává žádná okolnost, která by usnadnila řešení této rovnice, musíme jí řešit obecně. V takovém případě se rovnice třetího řádu řeší pomocí Cardanových vzorců [8]. V programovacím jazyce JAVA, pak tuto implementaci nalezneme například v třídě *EquationManager* z knihovny *ixent* a je dostupná pod licencí GNU [9].



Obrázek 3.1: Složky výkonu cyklisty

V grafu 3.1 jsou zobrazeny jednotlivé složky výkonu cyklisty v závislosti na jeho rychlosti dle vzorce (3.4) s volbou koeficientů $CS = 0.356$, hustoty vzduchu $\rho = 1,22 kgm^{-3}$, hmotností jezdce včetně kola $m = 70kg$, normálním tíhovým zrychlením g a sklonem vozovky $s_l = 5\%$. Můžeme názorně vidět, že při nižších rychlostech je hlavní složkou výkonu právě odpor sklonu vozovky. Až při rychlosti nad 45 km/h začíná výkon nejvíce ovlivňovat aerodynamický odpor.

3.1 Stanovení koeficientů z naměřených hodnot

Mark Cote provedl zajímavý test silničního a časovkářského kola od firmy Specialized [3]. Testy byly provedeny na dráze a pak také na běžné silnici, bez výrazného převýšení. Oba testy vykazují velmi podobné výsledky, které potvrzují jejich správnost. Výkony na dráze a na silnici (v bezvětří) by měly být velmi blízké, jelikož jediný rozdíl je v odporu, který vzniká vlivem nerovností vozovky. A ten je při silničních závodech velmi malý a obecně se zanedbává. Na dráze však nehraje roli vítr, který na běžném prostranství může v průběhu času měnit svůj směr a proto budeme uvažovat naměřené hodnoty z dráhy. Při průměrné rychlosti $40,07 kmh^{-1}$ musel závodník na silničním kole vyprodukovat průměrný výkon 291,1 W. Na časovkářském kole pak rychlosti $39,92 kmh^{-1}$ odpovídal výkon jen 220,8 W.

Naměřené hodnoty použijeme pro stanovení koeficientů v rovnici (3.4). Máme tedy známé hodnoty v podobě průměrné rychlosti a výkonu. Hmotnost kola s jezdcem rovnici nikterak zásadně neovlivňuje, jelikož test byl proveden na rovinatém terénu. A tak hmotnost bude ovlivňovat pouze sílu z valivého odporu a ta je poměrně nízká. Hmotnost tedy stanovíme na 80 kg, během testu byla naměřena průměrná teplota 31 °C a test probíhal na dráze v Asheville, který se nachází v nadmořské výšce 650 metrů. Za této teploty a nadmořské výšky se hustota vzduchu $\rho = 1,078 kgm^{-3}$. Test probíhal za co možná nejkonstantnější rychlosti, takže zrychlení ze vzorce (3.4) zanedbáme. Dále použijeme normální tíhové zrychlení, jelikož se v závislosti na nadmořské výšce a poloze nikterak výrazně nemění. Dosazením těchto hodnot do vzorce (3.4) tedy snadno zjistíme hodnotu

$$CS = \frac{2P - 0,006mgv}{v^3\rho}. \tag{3.6}$$

Ačkoliv nejsme schopni určit obsah průmětu cyklisty ve směru působení větru (S)

a součinitel odporu vzduchu (C), umíme určit jejich součin a taková hodnota je postačující. Výška i hmotnost závodníků je rozdílná, ale jejich průmět na bicyklu je relativně podobný. CS pro silniční kolo je tedy 0,356, pro časovkářský speciál pak 0,265.

4 Výsledky závodů

4.1 Datové zdroje

Mezinárodní cyklistická unie (UCI) žádné výsledky ve formě databází, či jiných datových výstupů, jakými jsou XML či $JSON$, veřejně neposkytuje. Data sdílí pouze s národními federacemi a případně oficiálními sponzory, jak vyplynulo z proběhlé komunikace. Na jejich webových stránkách jsou výsledky pro návštěvníky samozřejmě dostupné, ale jsou plně generované pomocí javascriptu, což znemožňuje účinné strojové zpracování. Další alternativu z oficiálních zdrojů představují pořadatelé jednotlivých závodů. Je zřejmé, že jejich data budou pravděpodobně pod stejnými právy jako data UCI. Získávání dat, ze stránek jednotlivých závodů, je pak nemožné z hlediska obrovské časové náročnosti. Každý závod je jinak formátován a bylo by nutné vytvořit velké množství programů pro stahování potřebných dat. Jediný přijatelný zdroj tedy tvoří statistické servery. Opět sice neposkytují žádný přímý výstup s daty, ale je možné vytvořit vlastní program, který data ze serveru získá. Program však bude přímo závislý na formátu webových stránek a v případě jakékoliv změny ze strany provozovatele, bude nutno velkou část vytvořeného programu na stahování dat přepsat. K této situaci nakonec i došlo. Stejně tak tomu bylo i v případě biatlonových výsledků, které práce rovněž využívá.

4.2 Program pro stahování výsledků

Zvolen byl statistický server procyclingstats.com, který shromažďuje velké množství cyklistických výsledků ,ale i startovní listiny nadcházejících závodů. U mnoha závodů jsou k dispozici také výškové profily závodů ve formě obrázku. Server a jeho web je primárně určen pro návštěvníky a tak jeho výstup představuje běžná HTML stránka.

Již bylo předesláno, že se formát webu při psaní diplomové práce změnil, popsána tedy bude jeho aktuální podoba.

Program, stahující cyklistické výsledky, využívá třídy z balíčku *system.imports* (*ImportFunction*, *URLWorker*) a jeho konkrétní implementace, kterou představuje především třída *CyclingImport* se nachází v balíčku *individual.cycling.imports*. Stahování dat ze serveru daný server samozřejmě vytěžuje a navíc je časově náročné. Časová náročnost lze řešit využitím více vláken, ale v takovém případě lze předpokládat, že server bude blokovat IP adresu, kterou využívá náš program. Text konkrétní webové stránky je stahován metodou *getText* třídy *URLWorker*. Při oslovování serveru je nutné nastavit http hlavičku prohlížeče, jinak je požadavek ze strany serveru zamítnut. Zejména při vývoji, je nutné často stahovat stejné stránky. S ohledem na časovou náročnost, ale i vytížení serveru jsou stažené stránky ukládány, a v případě požadavku na stáhnutí stejného obsahu, jsou načteny z lokálního disku uživatele.

4.2.1 Stáhnutí výsledků závodu

Každý dostupný závod je umístěn na adrese *http* : *//www.procyclingstats.com/race/race_id*, kde *race_id* je identifikátor daného závodu. Server rovněž podporuje i alternativní způsob zadání závodu ve formátu *http* : *//www.procyclingstats.com/race.php?id* = *race_id*, což bude pravděpodobně pozůstatek staré struktury webu, ale není přesměrováván na novou adresu. Na tento fakt je třeba si dávat pozor, často bylo potřeba získat danou adresu závodu a na serveru se objevuje poměrně nepochopitelně v obou verzích, proto je v programu interně přepisována na první verzi, aby nedošlo k nějaké záměně. Obdobná situace se opakuje i u dalších stahovaných stránek.

Pokud známe *race_id* můžeme začít stahovat danou stránku a z ní začít získávat potřebné informace. K rozparserování stránky je použit HTML parser *jsoup*. K získání potřebných výsledků jsou také hojně využívány regulární výrazy. Z této stránky získáme mimo výsledků i délku a datum konání závodu. Dále lze zjistit o jaký typ závodu se jedná, místu konání, či číslo etapy, v případě etapového závodu. Pokud je k dispozici výškový profil závodu, je rovněž k dispozici na specifickém místě na této stránce, respektive jeho odkaz na přímou adresu s obrázkem. Zjištění, jestli se jedná o časovku, či hromadný závod není úplně spolehlivé, jelikož je založeno na zjišťování,

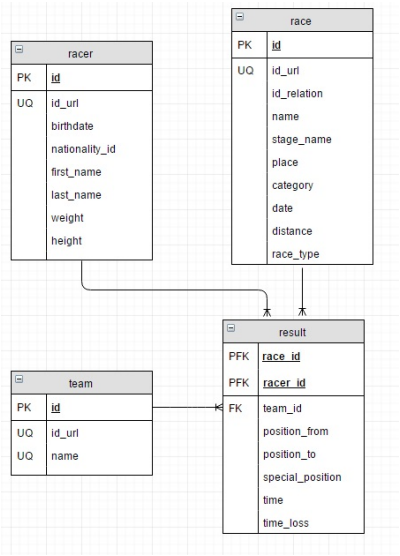
zda se v názvu etapy vyskytuje některé slovo, které je na těchto stránkách spojeno s časovkou (ITT, Time Trial a další). Výsledky závodníků se na stránce nachází ve formě tabulky, která je tvořena pomocí HTML značek typu div a span. U každého závodníka mohou být uvedeny údaje o umístění, času, název týmu, či bodech do celkového pořadí žebříčku UCI. Tyto hodnoty nejsou uvedeny vždy a mohou být v různém pořadí, proto je třeba zjisti, o jaká data se jedná nejdříve z prvního řádku tabulky a následně přizpůsobit získávání dat. Pokud bychom takto přímo ukládali výsledky, hrozilo by, že neidentifikujeme přesně závodníka, jelikož existuje možnost více závodníků stejného jména (včetně příjmení). Jméno závodníka však zároveň odkazuje i na stránku s jeho detaily, které má opět unikátní adresu a podle ní tedy jednoznačně rozpoznáme konkrétního závodníka. Z detailní stránky o závodníkovi ve tvaru $http://www.procyclingstats.com/rider/rider\_id$, kde $rider\_id$ je unikátní id závodníka, navíc zjistíme i datum narození závodníka a jeho národnost. Někdy bývá k dispozici i hmotnost a výška.

Aby mohl být uvedený postup efektivně použit, potřebujeme zjistit identifikátory závodů $race\_id$, což lze z kalendáře jednotlivých sezón, který je uveden na stejném serveru. Po zjištění závodů, již jen opakujeme výše popsaný algoritmus, dokud nejsou uloženy všechny požadované závody.

Ukládání dat

Pro ukládání potřebných dat byla vybrána databáze H2, důvody pro její výběr byly již zmíněny dříve. Všechna data jsou ukládána, po sesbírání všech výsledků o jednom závodu, v rámci jediné transakce. V případě jakékoliv chyby, budou všechna data z databáze vymazána a postup se bude znovu opakovat.

Z ER diagramu 4.1 můžeme vidět všechny detaily jednotlivých tabulek. U tabulek *racer* a *race* existuje přirozený primární klíč id_url, přesto je použit jako primární klíč speciální číselný identifikátor. K tomuto kroku bylo sáhnuto z důvodu zbytečného nárůstu objemu dat, jelikož ve výsledcích potřebujeme tuto vazbu zachovat. A zřejmě číselný identifikátor zabírá mnohem méně paměti, než řetězec o délce až 100 znaků. Zároveň si potřebujeme uchovat i vazbu na strukturu serveru, ze kterého data stahujeme.



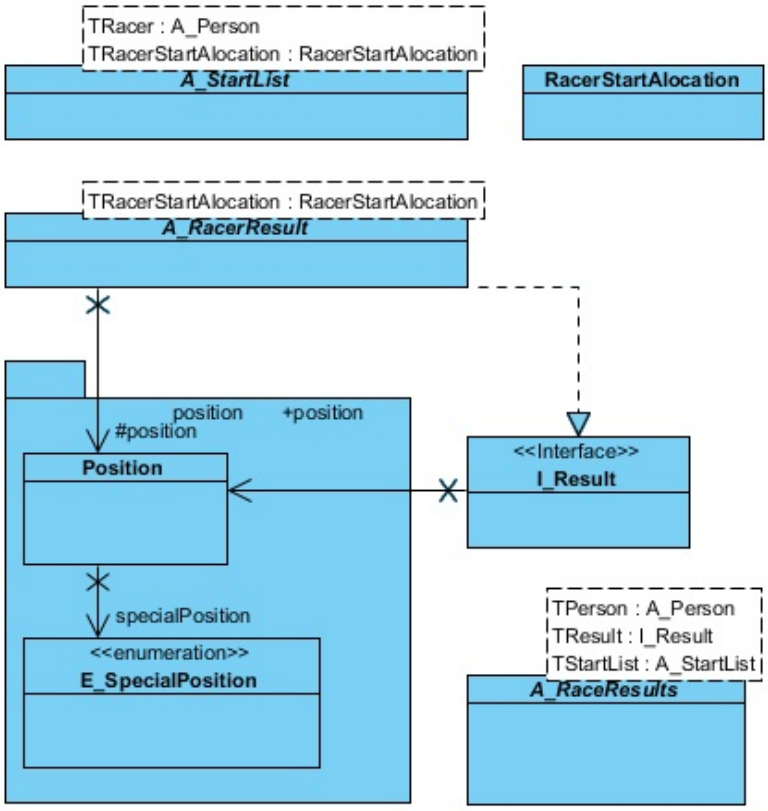
Obrázek 4.1: ER Diagram cyklistických dat

4.3 Objektový návrh statistik

Statistické modely potřebují k predikci výsledky předcházejících závodů. Výsledky se ukládají do databáze, ale pro práci s modely je třeba je mapovat do objektů. Návrh je udělán co nejobecněji, aby bylo možné využívat základní funkcionalitu společně pro různá sportovní odvětví a také vytvářet obecné modely pracující s obecnou výsledkovou vrstvou, společnou pro více sportovních odvětví. Obecný objektový návrh je umístěn v balíčku *system.statistics*, konkrétní implementace je následně u každého sportovního odvětví ve vlastním balíčku.

Na obrázku 4.2 vidíme diagram tříd, který je společný pro všechny sportovní odvětví z kategorie závodů. Abstraktní třída *A_Person* vytváří základ společný pro všechny závodníky, kteří od ní dědí, a obsahuje základní údaje, jakými jsou národnost, datum narození, pohlaví, jméno a nepovinné údaje o výšce a hmotnosti. Třída *RacerStartAlocation* obsahuje základní údaje o startovní pozici závodníka, kterými může být čas startu, či jeho pozice nebo ztráta na prvního startujícího. Startovní listina, kterou představuje abstraktní generická třída *A_StartList*, obsahuje u každého závodníka (libovolného potomka třídy *A_Person*) jeho startovní pozici (potomka *RacerStartAlocation*). Třída *A_RacerResult* představuje výsledky závodníka v rámci vybraného závodu a musí implementovat metody dané roz-

hraním *I_Result*. Umístění závodníka může být dělené, a závodník rovněž nemusí závod dokončit, být diskvalifikován, nebo do závodu neodstartuje. Tuto funkcionalitu zajišťuje třída *Position*. Výsledky celého závodu jsou pak obsaženy třídou *A_RaceResults*, která ke každému závodníkovi (potomek *A_Person*) uchovává jeho výsledky.



Obrázek 4.2: Objektový návrh výsledků závodu

Přístup ke všem výsledkům zprostředkovává abstraktní třída *A_Statistics*, která opět využívá genericity a pracuje s potomky tříd *A_RaceResults*, *A_Person* a rozhraní *I_Race*. Tato třída se statistikami umožňuje načíst výsledky podle zvoleného data z databáze. Dále vrací seřazená data závodů (sestupně i vzestupně), případně umí poskytnout závody dle data a samozřejmě výsledky po zadání konkrétního závodu. Statistiky, stejně jako některé další uvedené třídy, využívají hašovací mapy pro přístup k požadovaným datům.

4.4 Profil trasy

Profil trasy je velmi důležitý k zúžení favoritů na vítězství v závodě. Získán je ve formě obrázku a je ukládán do souborového adresáře, nyní je potřeba z něj získat co nejpřesnější data o sklonu vozovky v aktuálních úsecích.

Obrázek je načten pomocí knihovny *opencv*, která také poskytuje nástroje pro práci s ním. A je definován jako matice s rozměry k_{width}(šířka obrázku) x k_{height}(výška obrázku), jejíž prvky tvoří pixely. Tato matice je reprezentována třídou Mat. Každý pixel po načtení může být vyjádřen v různých barevných formátech.

Profil se ve většině případů vyznačuje určitou barvou, která je sice v každém obrázku různá, nicméně v rámci jednoho bývá obvykle pouze jedna. Úplná automatizace prakticky možná není, jelikož jsou potřeba zjistit údaje o nadmořské výšce a ty lze z obrázku přečíst jen velice obtížně. Nicméně graf si zachovává stejné měřítko, proto nám stačí zjistit 2 body, které nemají stejnou nadmořskou výšku a následně lze velice jednoduše ostatní body již dopočítat. Proto se uživateli zobrazí obrázek, z kterého následně zadá 2 požadované body. Jelikož je nutný tento zásah uživatele, který znemožní plně automatické zpracování, již nepředstavuje velký problém, pokud na obsluze necháme i jeden další krok, kterým je zadání barvy profilu trasy. Toto zadání provede velice jednoduše kliknutím na obrázek, ze kterého je barva vybrána podle pixelu, na kterém došlo k události kliknutí myší.

V programu je zabudované i automatické detekování, které vyhodnotí jako barvu profilu takovou, která je nejčetnější, přičemž bílá barva je vyřazena, jelikož pomocí ní profil obvykle vyznačen není. Úspěšnost této automatické detekce však není příliš vysoká a vzhledem k výše uvedenému je zadání ponecháno především na obsluze programu.

Nejsložitějším problémem je nalezení pixelů, které označují výškový profil závodu. K jejich nalezení bylo vytvořeno několik metod, které se zároveň i doplňují.

4.4.1 Sloupcová detekce

V každém sloupci pixelů, který představuje výškový profil, musíme označit jeden, který představuje nadmořskou výšku. Jelikož barevná plocha zvýrazněného profilu nemá naprosto stejnou barvu, není možné přímo porovnávat hodnoty z RGB modelu.

K vyhledání podobných barev využijeme gamut Lab, do kterého je nutné obrázek nejprve převést. Mějme vybranou barvu, která je dána vektorem $c_1 = (L_1, a_1, b_1)$. Tuto barvu potřebujeme porovnat s aktuálně vybraným pixelem z obrázku, což provedeme pomocí podmínky

$$d(c_1, c_2) \leq k_{color}, k_{color} \in \langle 0, 1 \rangle, \tag{4.1}$$

kde c_2 je vektor z Lab prostoru a označuje barvu aktuálně porovnávaného pixelu a $d(c_1, c_2)$ je vzdálenost barev daná rovnicí

$$d(c_1, c_2) = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2}. \tag{4.2}$$

Konstanta k_{color} je pak nastavena na hodnotu 0.1 ale uživatel jí může za běhu programu libovolně měnit. Cílem je nalézt v každém sloupci výšku vyznačeného profilu. Algoritmus pro stanovení hledaného pixelu, který znázorňuje námi hledanou nadmořskou výšku, je tedy nezávislý na ostatních sloupcích a proto bude nastíněn pouze pro vybraný sloupec.

Metoda zdola nahoru

Začneme od nejspodnějšího pixelu v rámci sloupce hledat první pixel, který splní podmínku ze vztahu (4.1). Tento pixel budeme také označovat jako detekční, v rámci této metody představuje nutnou, ale nikoliv postačující podmínku, pro nalezení výšky v sloupci. Pokud se nepodařilo najít žádný pixel, pak je hodnota v tomto sloupci neznámá. V opačném případě pokračujeme na vyšší, dokud je splněna podmínka (4.1). Při jejím prvním nesplnění jsme nalezli výšku profilu, která se nachází na předcházejícím pixelu. V tomto případě hovoříme o zastavovacích pixelech.

V případě ideálního obrázku by tento návrh měl být správný. Počet pixelů ve sloupci, které by měly být detekovány jako stejná barva, je obvykle vysoký. A jelikož nejsou obrázky zdaleka ideální, tak i jediný chybný pixel, může způsobit velké nepřesnosti. Tento problém lze úspěšně vyřešit pozměněním algoritmu tak, že po vyhledání pixelů s cílovou barvou nestačí ke stanovení výšky pouze první nesplnění námi popsané podmínky o hledání barev. Naopak budeme vyžadovat nesplnění v $k_{add} \in N, k_{add} > 2$ po sobě jdoucích pixelech. Při stanovení pozice pixelu, který

označuje výšku, je nutné odečíst nově zavedenou konstantu k_{add}, jelikož se jedná již o pixely jiné barvy. Funkci, která provede tento výpočet pro x-tý sloupec budeme značit $p_{up}(x) \in \{0, ..k_{height} - 1\}, x \in \{0, ..k_{width} - 1\}$

Metoda shora dolů

Pro vybraný sloupec stanovíme výšku, jako pixel, který jako první splní podmínku (4.1). Při tomto hledání postupujeme od horního pixelu. Obdobně jako u metody zdola nahoru využijeme větší počet zastavovacích pixelů, které musí následovat po sobě. Nicméně toto zlepšení zde není tak nutné jako u metody předcházející. Důvod je zřejmý, obvykle se jedná o chybu v obrázku a pravděpodobnost, že pixel obsahuje námi vybranou barvu, nebo podobnou dle nastavených kritérií, je menší, než že obsahuje jakoukoliv barvu jinou. Stejně jako u předcházející metody označíme výšku hledaného pixelu v x-tém sloupci $p_{down}(x) \in \{0, ..k_{height-1}\}, x \in \{0, ..k_{width-1}\}$

Kombinovaná metoda

Výše uvedené metody jsou navržené tak, že by měly být ekvivalentní a v případě ideálního obrázku výškového profilu také jsou. Nicméně profily jsou velmi různorodé a každá z metod má své výhody.

Obě strategie jsou problematické v případě, že se v obrázku vyskytuje stejná barva, která označuje výškový profil i pro jiné účely. Může jím být textový popis, či zvýraznění nějaké prémie. Vhodnou kombinací těchto metod můžeme problém odstranit.

Za věrohodně stanovenou výšku metodou shora $p_{up}(x)$ a metodou zdola $p_{down}(x)$ pro x-tý sloupec prohlásíme x-tý sloupec, který splní podmínku

$$|p_{down}(x) - p_{up}(x)| \leq k_{relative} k_{height}, x \in \{0, ..k_{width-1}\}, k_{relative} \in \langle 0, 1 \rangle, \tag{4.3}$$

a pro usnadnění si také nadefinujeme funkci

$$p(x) = \left\{ \begin{array}{ll} 1, & |p_{down}(x) - p_{up}(x)| \leq k_{relative} k_{height} \\ 0, & \text{ostatní} \end{array} \right\} \tag{4.4}$$

kde $k_{relative}$ je relativní velikost možné chyby, a byla zvolena hodnota $k_{relative} = 0.05$. Porovnáním výsledků, z dvou rozdílných metod, prakticky vyloučíme možnost

nalezení podobné barvy, která však nepředstavuje výšku závodu. Tato metoda bude mít za důsledek častěji nestanovenou výšku v rámci vybraného sloupce, avšak spolehlivost správně stanoveného pixelu bude vyšší.

Stanovení souřadnice věrohodného pixelu

Kombinaci metod budeme využívat jen pro stanovení x-ové souřadnice věrohodného pixelu, což je taková souřadnice, o které lze s velkou pravděpodobností prohlásit, že její výšku lze velmi přesně určit základními metodami shora a zdola. U takové souřadnice pak lze stanovit těmito metodami, prakticky bez rizika chyby, hledanou výšku a tu pak využít v metodách s omezením. Pokud bychom stanovili věrohodnou souřadnici při první splněné podmínce (4.3) stále by existovalo riziko, že nebyla zvolena správně. Profil trasy se zejména nenachází na celé jeho šířce. Takže se pokusíme rizikovým oblastem obrázku, pokud to bude možné, vyhnout.

K docílení spolehlivých výsledků celý obrázek rozdělíme na t částí podle osy x, budeme tedy hledat v menších intervalech. Množiny x-ových souřadnic T_i jsou tedy dány jako

$$T_i = \{\lfloor \frac{(k_{width}-1)(i-1)}{t} \rfloor, \lceil \frac{(k_{width}-1)(i)}{t} \rceil\}, i \in \{1,..t\}, \forall |T_i| \geq k_2, k_2 \in 2n{+}1, n \in N, \tag{4.5}$$

kde k_2 je počet po sobě jdoucích x-ových souřadnic v intervalu T_i, pro které musí platit (4.3). Prostřední z těchto k_2 souřadnic představuje námi hledanou výchozí souřadnici, která je dále používána. Pro zvýšení důvěryhodnosti celé množiny T_i zavedeme podmínku

$$\frac{\sum_{j=1}^{|T_i|} p(T_{i,j})}{|T_i|} < k_3, k_3 \in \langle 0,1 \rangle \tag{4.6}$$

kde k_3 je relativní úspěšnost určující podobnost stanovení pixelu pomocí dvou rozdílných metod na dané množině x-ových souřadnic.

Počet souřadnic, které by splnily uvedené podmínky může být velmi rozsáhlý. K dalšímu postupu však potřebujeme pouze jedinou a tak vybereme první, která kritéria splní. Abychom nejprve vyhledávali mezi nejlepšími kandidáty na stanovení věrohodné souřadnice, budeme prohledávat množiny T_i v následujícím pořadí $\{T_{\lfloor \frac{|T|}{2} \rfloor}, T_{\lfloor \frac{|T|}{2} \rfloor+1}, T_{\lfloor \frac{|T|}{2} \rfloor-1}, .., T_t, T_1\}$.

Omezení metod

Metoda shora i zdola trpí zásadním problémem, který spočívá v možném zanesení podobné barvy do obrázku nesouvisející s profilem trasy. Výrazně vylepšit lze tak, že se omezí hledání, v sloupci pixelů, jen na určitou část. Pokud nebudeme prohledávat oblasti, které jistě nemohou obsahovat hledanou výšku, vyhneme se potenciálně možným chybám. Potřebujeme tedy co nejvíce zúžit množinu sloupce, kde hledáme. Zároveň se, ale nesmí množina zmenšit příliš, mohli bychom i úplně vynechat hledaný pixel.

U základních metod operujeme v každém sloupci s množinou $P' = \{0, 1, ..k_{height} - 1\}$. Tuto množinu, však bez jakýchkoliv dalších znalostí, zúžit nelze. Z praktického hlediska však víme, že vozovka vytváří prakticky spojitou funkci a její sklon, jistě nemůže být větší než 30%. Takové silnice se zkrátka obvykle vůbec nebudují, v silniční cyklistice se na tento údaj můžeme poměrně dobře spolehnout. V obrázku však pracujeme pouze s pixely a také se zde vyskytují určité chyby. Zároveň může být profil v různém poměru os x a y.

Pokud máme tedy danou pozici pixelu $P_1 = [x_1, y_1]$, můžeme podle našeho předpokladu tvrdit, že pixel $P_2 = [x_2, y_2]$ bude mít souřadnici y_2 omezenou podmínkou

$$y_1 - |x_1 - x_2|k_{multiple} \leq y_2 \leq y_1 + |x_1 - x_2|k_{multiple}, \tag{4.7}$$

kde $k_{multiple}$ je koeficient možné změny v ose y v závislosti na ose y. Zavádíme tedy omezení na jiném sloupci, ideálně s co možná nejbližší souřadnicí na ose x. Tato závislost je jediným rozdílem oproti metodám, které jsou popsány výše. Metoda shora s omezením zřejmě musí začínat na souřadnici $y_2 = y_1 + |x_1 - x_2|k_{multiple} + k_{detect}$, kde k_{detect} označuje počet detekčních pixelů. Metoda zdola pak začíná na souřadnici $y_2 = y_1 - |x_1 - x_2|k_{multiple} - k_{detect}$

Pokročilé metody s omezením nejprve stanoví výchozí(věrohodný) pixel $P_v = (x_v, y_v)$ a dále pokračují na pixel $P_{v+1} = [x_v + 1, y_v + 1]$, kde se pomocí omezené metody shora či zdola určí y_{v+1}. Obdobným způsobem se dále pokračuje na další pixely, dokud nenarazíme na poslední souřadnici na ose x. Omezující podmínka se aplikuje pro nejbližší zjištěný pixel. Takto jsme určili všechny hledané pixely pravostranné řady x-ových souřadnic $X_r = \{x_v, x_{v+1}, x_{v+2}, ..x_{k_{width}-1}\}$. Stejným

postupem určíme i postupně y-ové souřadnice k pixelům s x-ovými souřadnicemi $X_l = \{x_v, x_v - 1, x_v - 2, .., x_0\}$.

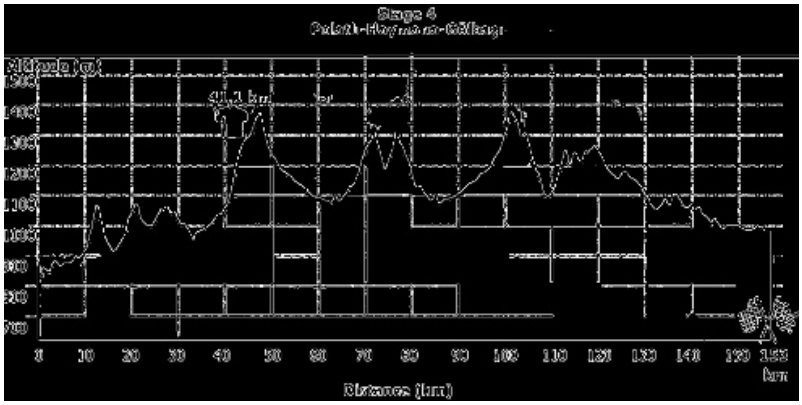
Ve výchozí implementaci jsou zvoleny konstanty $k_{detect} = 10, k_{multiple} = 5$, které bezpečně zaručí možnost najít hledaný pixel. Metody s omezením nejprve potřebují najít věrohodný pixel, což je poměrně výpočetně náročný algoritmus. Přesto jsou díky omezení, celkově rychlejší než metody základní, a zároveň mají schopnost lépe rozpoznat hledaný profil trasy. Drobnou nevýhodou je nutnost přesnějšího nastavení limitu barev, které jsou rozpoznávány při stanovení věrohodného pixelu.

4.4.2 Hranová detekce

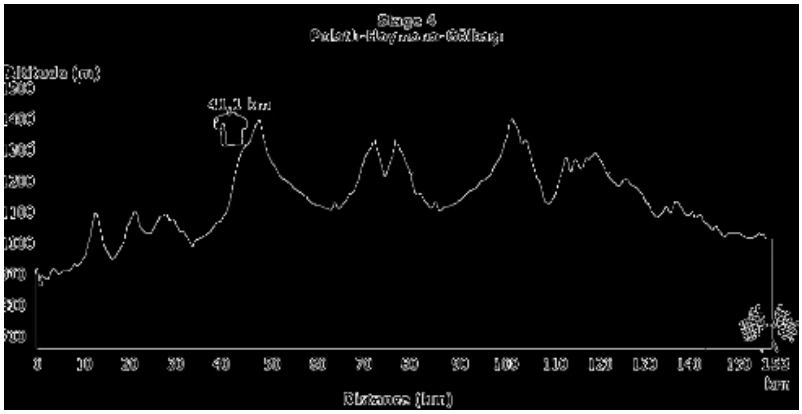
Navržená metoda sloupcové detekce vykazuje ve velkém množství případů velice dobré výsledky, v některých však selhává. Již u sloupcové detekce byla zavedena závislost na ostatních sloupcích, jelikož nadmořská výška silnice v obrázku představuje hranu. Nabízí se tedy možnost využít detekci hran, která je v oblasti rozpoznávání obrazu velmi dobře známa.

Potřebujeme získat hrany z obrázku, který znázorňuje profil závodu. Hrany stanovíme pomocí Cannyho detektoru, který se skládá ze 4 základních kroků. Nejprve eliminuje šum pomocí Gaussova filtru, následně se stanoví gradient, naleznou lokální maxima a nakonec se eliminují nevýznamné hrany. K realizaci v programovacím jazyce využijeme knihovnu *OpenCV*. Vstupní obrázek nejprve převedeme do odstínů šedi, pomocí funkce *cvtColor*. Následně jej pomocí funkce *blur* vyhladíme filtrem o velikosti 3 x 3. A na konec použijeme funkci *Canny*, která vytvoří hrany podle Cannyho detektoru. Uvedený postup je volen přesně dle tutoriálu využité knihovny [18]. Velmi důležitá je volba horního a dolního prahu pro detektor. Tato volba výrazně ovlivňuje množství detekovaných hran. Na obrázku 4.3 vidíme výsledek detekce v případě velmi nízkého prahu. Došlo tedy k nalezení velkého počtu hran, i takových, které profil vůbec nevyznačují. Následné odhalení správné hrany by tak bylo velmi složité.

Na obrázku 4.4 naopak vidíme vhodně zvolený prah, který počet hran velmi omezil. Přestože se jedná o ideální obrázek, s velmi vhodně zvoleným prahem pro detekci hran, stále byly nalezeny i takové hrany, které nepředstavují výškový profil trasy. K ideálnímu případu, kdy by byly nalezeny jen hrany představující hledaný



Obrázek 4.3: Příliš nízký prah pro detekci hran



Obrázek 4.4: Vhodně zvolený prah pro detekci hran

profil, však prakticky nikdy nedochází.

Detekované hrany máme nyní uloženy v matici, kterou představuje třída Mat a má rozměry identické vstupnímu obrázku, z něhož jsou hrany vytvořeny. Hodnoty v x-tém řádku a y-tém sloupci této matice jsou dány funkcí $m(x, y) \in \{0, 1\}$. V případě, že $m(x, y) = 1$ byla na y-tém sloupci a x-tém řádku detekována hrana.

Nalezení správné hrany zřejmě, bez dalších znalostí, není možné. Automatická detekce i vzhledem k dalším okolnostem, již byla dříve vyloučena. Necháme tedy uživatele vybrat správnou hranu manuálně. Uživatel bude vyzván, aby myší vybral pixel z obrázku, který je součástí hledané hrany. Pixel bude mít souřadnice $x_c \in \{0, ..k_{length} - 1\}, y_c \in \{0, ..k_{height} - 1\}$. Máme tedy předpoklad, že hledaná hrana by se měla nacházet velmi blízko souřadnicím x_c, y_c.

Nadefinujme si funkci, která hledá nejbližší hranu v rámci vybraného sloupce $f_e(x, y, x_{last}, k)$, kde (x, y) jsou souřadnice, kde předpokládáme hledanou hranu a

x_{last} je x-ová souřadnice posledně nalezené hrany a k je koeficient omezující hledání hrany. Funkce bude vracet y-ovou souřadnici y_f, kde y_f bude první souřadnice, která splní podmínku $m(x, y_h) = 1$, ve které je dosazováno v daném pořadí $y_h \in \{y, y+1, y-1, y+2, y-2, ..\}, 0 \leq y_h \leq k_{height} - 1 \& |y_h - y| \leq |x - x_{last}|k$.

Vyjděme tedy ze souřadnic (x_c, y_c) k nim nalezneme první skutečnou polohu hrany $(x_c, f_e(x_c, y_c, x_c - 1))$. Pokud funkce f_e úspěšně nalezla hranu, budeme pokračovat ve zjišťování další souřadnice a bude mít hodnotu $(x_c+1, f_e(x_c, f_e(x_c, y_c, x_c - 1), x_c))$, v opačném případě pokračujeme také, ale poslední úspěšně vyhledaná souřadnice zůstává původní, takže následující poloha bude mít souřadnice $(x_c+1, f_e(x_c, y_c, x_c - 1))$. Stejným způsobem vyhledáme i všechny následující souřadnice. Získáme tak všechny souřadnice vpravo od prvního vyhledaného pixelu. Od výchozího pixelu následně budeme postupovat i opačným směrem, postup je analogický, jen výchozí souřadnice $x_{last} = x_c + 1$. Zjednodušeně můžeme říct, že z výchozího pixelu pokračujeme oběma směry po hledané hraně a detekujeme tak pouze ji. Problémy tak nastávají jen v případě, že se hrany navzájem protínají.

4.4.3 Stanovení výškového profilu

Pro přepočtení nalezeného profilu z pixelů, na nadmořskou výšku a vzdálenost od startu, nám postačí záznam s nalezenými pixely $(x_i, y_i), 0 \leq i \in N \leq k_{width} - 1$. Dále z těchto hodnot spočteme nejmenší, respektive největší hodnoty na obou osách, které označíme $x_{min}, x_{max}, y_{min}, y_{max}$. Rovněž musíme znát vzdálenost závodu l, nejvyšší (h_{max}) a nejnižší (h_{min}) nadmořskou výšku. Na základě těchto hodnot jsme již schopni stanovit libovolnou nadmořskou výšku a vzdálenost u každého pixelu.

Nadmořskou výšku ve vybraném pixelu vypočteme jako

$$h_{height}(y) = y_{zero} + y_{coef} y; y_{coef} = \frac{h_{max} - h_{min}}{y_{max} - y_{min}}; y_{zero} = h_{min} - y_{min} y_{coef}, \quad (4.8)$$

kde y je y-ová souřadnice pixelu v obrázku, y_{zero} je nadmořská výška odpovídající nejnižšímu pixelu a y_{coef} je příbytek nadmořské výšky při posunu o jeden pixel nahoru. Obrázky jsou indexované vždy od levého horního rohu, který má tedy souřadnice $(0, 0)$. Je tedy velmi důležité s tím při implementaci počítat.

$$l(x) = (x - x_{min}) l_{pp}; l_{pp} = \frac{l}{x_{max} - x_{min}}, \quad (4.9)$$

kde x je x-ová souřadnice pixelu v obrázku, l_{pp} je přírůstek vzdálenosti při pohybu o 1 pixel dále.

4.4.4 Grafické rozhraní

Uvedené metody a implementované algoritmy, vzhledem k různorodým obrázkům, nemohou být zcela univerzální a bezchybné. Proto bylo vytvořeno grafické rozhraní, pomocí kterého uživatel může jednoduše zvolit vhodná řešení a vykonává i funkci kontrolora.

Pomocí Scene Builderu byl vytvořen základní vzhled grafického rozhraní. Model z návrhového vzoru MVC představuje třída $ImageToProfileConvertor$, controller pak $ImageToProfileController$. Aby mohlo grafické rozhraní i model běžet zdánlivě současně (pseudoparalelně) nebo v případě vícejádrového počítače úplně paralelně, musí rozhraní i model běžet v jiném vlákně. Proto třída $ImageToProfileConvertor$ implementuje rozhraní *Runnable*. Uživatel nejprve načte obrázek, ze kterého chce vytvořit výškový profil. Realizace je velmi jednoduchá s využitím třídy $FileChooser$ a její metody *showOpenDialog*, pomocí niž můžeme do naší aplikace nahrát libovolný obrázek. Třída $FileChooser$ umožňuje nabízené soubory filtrovat. Uživateli tedy nabídneme možnost výběru pouze ze souborů, které jsou obrázky pomocí třídy $ExtensionFilter$. Také přidáme speciální filtr, který nabídne pouze takové obrázky, jejichž profily zatím nebyly přidány do databáze. Obrázky mají před koncovkou jméno odpovídající atributu *url_id* z tabulky *cycling_race*. Sql dotaz, který vybereme všechny závody, které nemají záznam v tabulce $cycling\_prifile$ vypadá následovně

```
SELECT url_id FROM CYCLING_RACE
WHERE id NOT IN (SELECT DISTINCT(race_id) FROM cycling_profile)
AND id_relation IS NULL
```

Atribut *id_relation* musí být NULL, jinak by se jednalo o celkové výsledky etapových závodů, u kterých však žádný profil nemůžeme nalézt. Takový obrázek se samozřejmě ani nikde vyskytovat nemůže, takže by nebylo chybou tuto podmínku vynechat, ale zbytečně bychom přidávali filtry, které nejsou potřeba.

Po načtení se obrázek zobrazí v aplikaci a zároveň se automaticky doplní délka

Obrázek 4.5: Grafická aplikace pro hledání výškového profilu

závodu podle údaje, který je uložen v databázi. Pokud by údaj neodpovídal, může jej uživatel přepsat a posléze bude společně s profilem uložen. Obsluha programu následně musí vyplnit informace o nadmořské výšce. Buď vyplní minimální a maximální nadmořskou výšku, nebo nadmořskou výšku na startu a v cíli závodu za podmínky, že se nerovnají. Pomocí těchto údajů následně může být vypočten skutečný výškový profil. Nyní přichází na řadu samotné rozpoznávání profilu, uživatel si může vybrat ze 4 základních funkcí, které jsou výše popsány. V případě sloupcové metody shora, zdola, nebo její kombinace, je třeba zadat práh pro označení barvy za shodnou pomocí slideru (posuvného tlačítka) a kliknutím myši vybrat barvu profilu z

obrázku. Po kliknutí se začne okamžitě požadavek zpracovávat. Výsledek je následně zobrazen pod obrázkem originálního obrázku.

V případě výběru hranové detekce je třeba nejprve pomocí slideru vybrat vhodný práh pro vytvoření hran. Detekce je velmi rychlá a tak se uživateli okamžitě při pohybu sliderem zobrazuje. Poté, co je uživatel s detekovanými hranami spokojen, klikne na spodní obrázek a ten nahradí obrázek původní. Dále již stačí jen kliknout myší poblíž hledané hrany a program opět zobrazí v dolním obrázku nalezenou hranu.

Na obrázku 4.5 vidíme výslednou aplikaci, kde bylo použito kombinované sloupcové detekce. Na detekovaném obrázku je červenou barvou znázorněn profil vyhledaný vlevo od věrohodného pixelu a černou profil vpravo. Uživatel může měnit metody detekce a prahy, než je spokojen s výsledkem. Po zobrazení obrázku s detekovaným profilem se aktivuje tlačítko pro uložení profilu.

5 Statistické modely

Stěžejním cílem modelů je stanovit pravděpodobnost všech možných výsledků, k čemuž využívají výsledky z předchozích závodů. Cílem práce je vytvoření co nejobecnějšího modelu, který zahrne především silniční cyklistiku. Plně obecný model pro všechny sporty by mohl velmi snížit kvalitu výsledného modelu, jelikož bychom velmi obtížně hledali spojitosti mezi příliš rozdílnými sporty. Je zřejmé, že např. fotbal a cyklistika mají jen velmi málo společných faktorů. Z těchto důvodů omezíme modely na práci s individuálními sporty, které ke klasifikaci používají dosažený čas závodníků.

Model pro svou práci nezbytně potřebuje startovní listinu S predikovaného závodu, která obsahuje množinu závodníků. Druhým nezbytným elementem, pro fungování modelu, jsou výsledky předchozích závodů. Pro j-tý závod a i-tého závodníka musíme mít k dispozici jeho umístění, které může být i dělené. K tomu dojde v případě, že několik závodníků dokončilo závod v naprosto shodném čase. Proto zavedeme značení pro nejlepší dělené umístění ($r_{top,i,j}$) a nejhorší dělené umístění $r_{down,i,j}$. V případě, že závodník neskončí na děleném umístění, tak $r_{top,i,j} = r_{down,i,j}$. V mnoha případech se stane, že závodník do závodu nenastoupí, je diskvalifikován, či jej nedokončí. V takovém případě závodník s ohledem na pravidla nemusí být ani klasifikován a není mu tak přiřazeno žádné konečné umístění. Pro zjednodušení tyto speciální druhy umístění nebudeme v modelech rozlišovat, ale závodníka automaticky zařadíme na poslední místo, o které se bude dělit společně s dalšími neklasifikovanými závodníky. Dále známe celkový čas $t_{total,i,j} \in R$, pro zjednodušení si označme i ztrátu závodníka na vítěze $t_{loss,i,j} \in R$. V případě, že závodníci závod nedokončí, jim opět přiřadíme takový čas, který by je řadil na poslední místo. Společně s těmito výsledky je třeba mít k dispozici i startovní listiny pro j-tý závod S_j.

Na základě popsaných statistik a startovní listiny k predikovanému závodu mo-

del stanoví pro i-tého závodníka pravděpodobnostní funkci $p_i(x)$, kde $x \in N_+$ označuje umístění závodníka. Se znalostí této funkce pak již velmi snadno zjistíme pravděpodobnost, že se i-tý závodník umístí od x_1. do x_2. místa pomocí funkce $P_i(x_1 \leq X \leq x_2) = \sum_{x=x_1}^{x_2} p_i(x)$.

Všechny modely musí nutně splňovat následující podmínky, aby mohly být považovány za správně navržené.

$$P_i(1 \leq X \leq |S|) = 1; i \in S \tag{5.1}$$

$$\sum_{i \in S} p_i(x) = 1; x \in \{1, 2, ..|S|\} \tag{5.2}$$

První podmínka (5.1) říká, že závodník musí jistě dokončit závod mezi 1. a posledním místem. Další podmínka (5.2) se stará o správné rozdělení pravděpodobnosti mezi jednotlivé pozice, napříč celým startovním polem.

5.1 Parametry závodu

K větší obecnosti výsledných modelů je vhodné stanovit základní parametry závodů, které budou reflektovat výsledné modely. Zaměříme se na cyklistiku a navrhneme takové kategorie, které by ji mohly co nejlépe reprezentovat.

Prvním parametrem je délka závodu $l \in R$. Trasu však neudává jen její vzdálenost, ale i výškový profil $h(x), x \in X = \{x_1 = 0, x_2, x_3, ..x_{n-1}, x_n = l\}$. Tato funkce, která vyjadřuje nadmořskou výšku ve vzdálenosti x od začátku závodu, zřejmě není spojitá. Abychom získali spojitou funkci, dodefinujeme funkci $h(x)$ v neznámých vzdálenostech $y \notin X$, pomocí lineární interpolace

$$h(y) = h(k_1) + \frac{y - k_1}{k_2 - k_1}(h(k_2) - h(k_1)); k_1, k_2 \in X, 0 < y < l, \tag{5.3}$$

kde $k_1(k_2)$ je nejbližší menší (větší) hodnota k y, na které je definovaná funkce $h(x)$. Takže nyní máme funkci $h(x)$ definovanou na celém intervalu $\langle 0, l \rangle$.

Profil je poměrně náročné získat a také často není k dispozici. U sportů jako biatlon, či běžecké lyžování, bývá k dispozici údaj o celkovém (akumulovaném) převýšení (h_{total}), maximálním výškovém rozdílu na celé trase (h_{diff}), či největším převýšení

na jediném kopci (h_{max}). Pokud je k dispozici funkce $h(x)$, lze z ní samozřejmě velmi jednoduše uvedené hodnoty jednoduše spočítat.

Do parametrů lze zahrnout i datum závodu $d \in N$, pomocí něhož lze zohlednit aktuální formu závodníků, a bude vyjádřen ve dnech.

Závod dále můžeme dělit podle typu závodu (b_{start}) a tato hodnota bude nabývat hodnoty 0 pro hromadný start a 1 pro závod individuální. Jako hromadný závod vnímáme i Gundersenovu metodu, kde závodníci startují v časových odstupech, které si vytvořili v předcházejících závodech. Pokud je závod součástí etapového závodu, pak $b_{stage} = 1$. V opačném případě $b_{stage} = 0$.

Dále známe u každého závodu jeho úroveň, kterou označíme $e \in N$. Nabývá sice celých kladných čísel ale jedná se pouze o identifikační čísla.

5.2 Základní model

Mějme k dispozici výsledky předešlých n závodů a predikovaný označme jako $n+1$. Cílem modelu je získání pravděpodobnostní funkce pro i-tého závodníka $p_i(x)$.

V základním modelu zanedbáme všechny parametry závodů a vytvoříme pouze takový model, který splňuje uvedené podmínky (5.2) a (5.1). Model bude sloužit zejména pro srovnání s dalšími modely a vysvětlení základních prvků, které budou používat i pokročilejší modely.

Ke zpřehlednění práce si nadefinujeme funkci

$$r_{weight}(x,i,j) = \begin{cases} \frac{1}{r_{down,i,j} - r_{top,i,j} + 1}; r_{top,i,j} \leq x \leq r_{down,i,j} \\ 0; x < r_{top,i,j} \vee x > r_{down,i,j} \end{cases}, \tag{5.4}$$

která vrací váhu, x-tého místa i-tého závodníka v j-tém závodě, vzhledem k možnému dělení pozice. V případě, že se závodník dělí o některá umístění, je jejich váha rovnoměrně rozdělena mezi dané pozice.

Pro všechny závodníky ze startovního pole S, určíme četnosti jejich předchozích umístění, podle následujícího vzorce

$$p'_i(x) = \sum_{j=1}^{n} r_{weight}(x,i,j); x \in \{1,..m\}, \tag{5.5}$$

kde $i \in S$ a $m \in N$ označuje nejhorší umístění závodníků startovního pole S v předcházejících závodech. V případě, že se i-tý závodník ještě nezúčastnil žádného závodu, byla by celá funkce $p'_i(x) = 0$. Takovému závodníkovi tedy přiřadíme rovnoměrné četnosti od prvního až do posledního zjištěného umístění (m) u ostatních závodníků. Jeho funkce četností bude mít následující podobu $p'_i(x) = 1, x \in \{1, ..m\}$.

Funkce $p'_i(x)$ nyní představuje funkci četností a pro všechny závodníky je její součet nenulový, zároveň však nepředstavuje pravděpodobnostní funkci, je ji tedy třeba znormalizovat pomocí následujícího vzorce

$$p''_i(x) = \frac{p'_i(x)}{\sum_{i=1}^{m} p'_i(x)}, \tag{5.6}$$

kde m je opět nejvyšší možné historické umístění kteréhokoliv závodníka (poslední možná nenulová hodnota $p'_i(x)$). Pro každého závodníka nyní zřejmě součet všech hodnot funkce $p''_i(x)$ je roven jedné. Přesto nesplňuje podmínku (5.1), jelikož funkce může nabývat hodnoty $p''_i(x) > 0$ i v případě, že $x > |S|$. Z toho zřejmě plyne, že $P_i(1 \leq X \leq |S|) < 1$. Zmíněnou podmínku lze splnit normalizací na menší množinu možných umístění, následně by však stejně nebyla splněna podmínka (5.2).

Normalizované četnosti pro každého závodníka $(p''_i(x))$ budeme považovat za nezávislé pravděpodobnostní funkce a vytvoříme z nich výslednou predikci, která již uvedené podmínky splňovat bude.

5.2.1 metoda Monte Carlo

Výpočetní náročnost metody je však příliš vysoká. Mohla by se skládat až z $|S|^{|S|}$ výpočetních kroků a proto bude použita metoda Monte Carlo.

Monte Carlo bude probíhat v n simulačních krocích. Nejprve si nadefinujeme funkci $u_i(r)$, která vrací poslední umístění (x_1), které bude použito tak, aby platilo $\sum_{x_1 \in X} p''_i(x) \leq r$, kde X je množina vzestupně seřazených umístění, které nabývají nenulových pravděpodobností ve funkci $p''_i(x)$.

V každé simulaci nejprve vygenerujeme pro každého závodníka náhodné číslo r_i, vytvoříme množinu umístění $U = \{u_i(r_i)\ \forall i \in |S|\}$. Nyní se podíváme na hodnotu i-tého závodníka v množině U a spočteme, kolik hodnot v množině nabývá menších hodnot (x_b) a kolik hodnot nabývá vyšších hodnot (x_a). Pro i-tého závodníka následně

Tabulka 5.1: Test dobré shody

Interval	$\langle 0, 0.1)$	$\langle 0.1, 0.2)$	$\langle 0.2, 0.3)$	$\langle 0.3, 0.4)$	$\langle 0.4, 0.5)$
Skutečné četnosti	10001220	9999114	9997473	9998714	10002553
χ^2	0.15	0.08	0.64	0.17	0.65
Interval	$\langle 5, 0.6)$	$\langle 0.6, 0.7)$	$\langle 0.7, 0.8)$	$\langle 0.8, 0.9)$	$\langle 0.9, 1\rangle$
Skutečné četnosti	9998984	10001536	9998766	9999148	10002492
χ^2	0.1	0.24	0.15	0.07	0.62

již jen přičteme do pravděpodobnostní funkce $p_i(x)$ na pozice $X = \{x_b+1, ..|S|-x_a\}$ pravděpodobnost $\frac{1}{n(|S|-x_b-x_a)}$. Tento krok opakujeme přesně n-krát. Metoda zřejmě mnohokrát generuje náhodné číslo, n-krát musí řadit množinu U a také používat $n|S|$ funkci $u_i(r_i)$. Výpočetní náročnost zřejmě závisí na velikosti startovního pole $|S|$ a počtu simulačních kroků n.

Správnost simulace pomocí metody Monte Carlo nutně závisí na generování náhodných čísel. S ohledem na pravděpodobnost generujeme čísla v intervalu $\langle 0, 1\rangle$ a předpokládáme, že jejich rozdělení by mělo být rovnoměrné. Tuto hypotézu ověříme pomocí testu dobré shody. Obor všech možných hodnot rozdělíme na 10 intervalů a vygenerujeme 100 000 000 náhodných čísel. U každého intervalu tedy předpokládáme 10 000 000 hodnot. Celkové výsledky vidíme v tabulce 5.1.

Sečteme - li hodnoty χ^2 pro jednotlivé intervaly získáme $\sum_{k=1}^{10} \chi^2 = 2.87$. Tuto hodnotu porovnáme s kritickou hodnotou chí-kvadrát s 9 stupni volnosti na 5% hladině významnosti, která je 16.919. Tím jsme dokázali, že generování náhodných čísel má skutečně rovnoměrné rozdělení a můžeme jej použít pro metodu Monte Carlo.

5.3 Obecné modely založené na vahách

Zmíněné parametry závodů by měly podstatně zlepšit výslednou predikci. Jejich začlenění proběhne pomocí systému vah. Každému výsledku ze statistik přiřadíme určitou váhu pomocí parametrů, které se k danému závodu, potažmo výsledku, vztahují. Každý závod tedy získá svou váhu w_j vzhledem k predikovanému $(n+1.)$

závodu. Mírnou úpravou vzorce 5.5 získáme

$$p'_i(x) = \sum_{j=1}^{n} r_{weight}(x, i, j) w_j; x \in \{1, ..m\}. \tag{5.7}$$

.

Pokud výsledná váha j-tého závodu závisí na více parametrech, od kterých jsou odvozeny dílčí váhy $W_j = \{w_{j,1}, w_{j,2}, ..w_{j,k}\}$, výslednou váhu spočteme pomocí vynásobení všech dílčích vah dle vztahu

$$w_j = \prod_{i=1}^{k} w_{j,i}; w_{j,i} \in \langle 0, 1 \rangle, \tag{5.8}$$

kde k odpovídá počtu dílčích vah.

5.3.1 Datum závodu

Závodníkům se v průběhu roku mění jejich aktuální kondice. Nabízí se možnost zavést váhu data konání předchozích závodů, vzhledem k predikovanému závodu. Čím blíže jsou si predikovaný závod s již proběhlým, tím větší váha bude daným výsledkům přiřazena. Nadefinujme si tedy funkci, která bude váhu stanovovat následovně

$$w_{ex\_date,j}(k) = e^{-k(d_{n+1} - d_j}; k \in \langle 0, 1 \rangle, \tag{5.9}$$

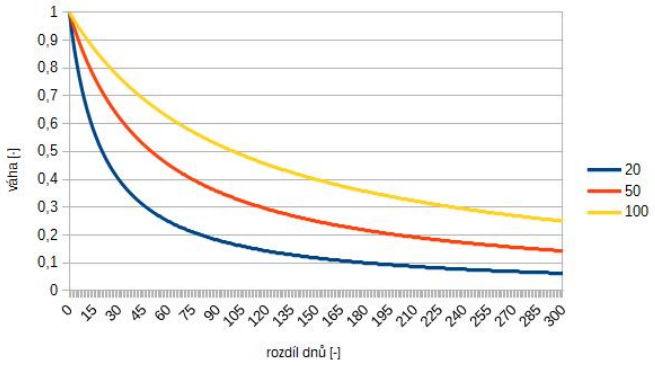
kde d_j je datum j-tého závodu, ke kterému hledáme váhu $w_{d,j}$. d_{n+1} představuje datum predikovaného závodu a k je koeficient určující rychlost poklesu exponenciály. Tato funkce se často používá při predikci fotbalových zápasů. Nevýhodou funkce jsou příliš nízké hodnoty pro příliš časově vzdálené závody.

Navrhněme ještě další funkci

$$w_{date,j}(k) = \frac{k}{k + d_{n+1} - d_j}; d_j < d_{n+1}, k > 0, \tag{5.10}$$

kde k je koeficient, který udává počet dnů po nichž se nejdříve váha zmenší na polovinu své původní hodnoty. Délka poklesu na další polovinu probíhá vždy na dvojnásobně dlouhém intervalu.

Na grafu 5.1 vidíme pokles váhy funkce (5.10) v závislosti na rozdílu počtu dní mezi vybraným a predikovaným závodem. Jednotlivé křivky pak popisují volbu koeficientu k. Čím nižší je tento koeficient, tím větší význam se přisuzuje nedávným výsledkům.



Obrázek 5.1: Funkce váhy pro datum

5.3.2 Druhy závodů

Závody se odlišují svou úrovní i druhem startu. Závody jsme si již dříve rozdělili podle startu (b_{start}) na individuální a hromadné. Tato proměnná nabývá jen 2 hodnot. Funkce, vyjadřující váhu v závislosti na typu startu, bude vypadat následovně

$$w_{start,j}(k) = \begin{cases} 1; b_{start,j} = b_{start,n+1} \\ k; b_{start,j} \neq b_{start,n+1}; k \in \langle 0, 1 \rangle \end{cases},$$

kde $b_{start,n+1}$ je typ startu predikovaného závodu a $b_{start,j}$ typ startu j-tého závodu, ke kterému stanovujeme váhu a k koeficient stanovení váhy v případě, že se typy startů neshodují. V případě, že $k = 1$ budou rozdílné způsoby startu úplně zanedbány.

Dále jsme si definovali úroveň závodu $e \in N$. Funkce určující váhy bude vypadat následovně

$$w_{level,j}(k) = \begin{cases} 1; e_j = e_{n+1} \\ k; e_j \neq e_{n+1}; k \in \langle 0, 1 \rangle \end{cases},$$

kde j je index aktuálně posuzovaného závodu, $n + 1$ index predikovaného závodu a k hodnota váhy pro případ, že se úrovně závodu neshodují.

5.3.3 Délka závodu

Délka závodu l může velmi výrazně ovlivnit průběh a výsledky závodu, proto představuje vhodného kandidáta na vytvoření funkce pro dílčí váhu. Cílem je vytvořit funkci,

která klade závodům s podobnou délkou predikovanému závodu, vyšší váhu. Funkce je navržena v několika následujících krocích

$$\begin{aligned} l_{max} &= max(l_{longest} - l_{n+1}, l_{n+1} - l_{shortest}); \\ w_{length,j}(k) &= (1 - \frac{|l_j - l_{n+1}|}{l_{max}})^k; k >= 0, \end{aligned} \tag{5.11}$$

kde l_j je délka j-tého závodu, l_{n+1} délka predikovaného závodu, $l_{longest}$ představuje délku nejdelšího závodu mezi všemi $n+1$ závody (tedy včetně predikovaného) a $l_{shortest}$ je naopak nejkratší závod na stejné množině závodů a koeficient k opět zvýrazňuje rozdíl mezi délkami závodů. Z uvedeného vztahu je zřejmé, že se nejprve vypočte největší možný rozdíl mezi délkou predikovaného závodu a všemi ostatními. Následně se použije při výpočtu výsledné váhy. Takto navržený postup výpočtu v případě velmi podobných délek závodů pomůže při dobrém rozlišení, nebo může být naopak až příliš citlivý. Další úskalí může představovat systém rozdílů mezi délkami závodů. Mezi predikovaným závodem na 100 km a závody na 10 km a 190 km totiž zavádí stejnou váhu, přičemž závod na 100 km je z hlediska predikce pravděpodobně bližší závodu na 190 km. Tuto vadu lze vyřešit jednoduchým vztahem

$$w_{length2,j}(k) = (\frac{min(l_j, l_{n+1})}{max(l_j, l_{n+1})})^k. \tag{5.12}$$

Tento výraz naopak může nedostatečně rozlišovat délkově podobné závody.

5.3.4 Členitost terénu

Členitost terénu může výrazně ovlivnit výsledky závodů. V kopcovitém terénu sportovci musejí vydat více energie a lehčí závodníci mohou mít často výhodu. Vliv v cyklistice jsme si již detailně popsali v kapitole o výkonu cyklistů. Ale význam má i pro mnohá další sportovní odvětví.

Náročnost trasy s ohledem na její členitost lze vyjádřit pomocí celkového převýšení $h_{total,j}$, která u některých sportů bývá přímo uvedena. Její hodnota je však zřejmě závislá na délce závodu a tak náročnost terénu vyjádříme poměrem mezi celkovým převýšením a délkou závodu l_j. Váhu pak stanovíme pomocí následující funkce

$$w_{profile,j}(k) = (\frac{min(\frac{h_{total,j}}{l_j}, \frac{h_{total,n+1}}{l_{n+1}})}{max(\frac{h_{total,j}}{l_j}, \frac{h_{total,n+1}}{l_{n+1}})})^k, \tag{5.13}$$

kde $h_{total,j}$ představuje převýšení v j-tém závodě, $h_{total,n+1}$ převýšení v predikovaném závodě, l_j délku j-tého závodu a l_{n+1} délku predikovaného závodu a $k > 1$ je koeficient zvyšující rozdíly mezi váhami.

Pokud u některých závodů není definováno celkové převýšení, ale máme k dispozici funkci s nadmořskou výškou v závislosti na vzdálenosti od startu $h_j(x)$, můžeme celkové převýšení $h_{total,j}$ samozřejmě vypočítat. Funkci $h_j(x)$ jsme již dříve dodefinovali, aby byla spojitá. Jelikož dodefinování spočívalo v lineární interpolaci známých nadmořských výšek ve vzdálenostech $X = \{0, x_2, ..x_{n-1}, l_j\}$, vystačíme si nyní jen s touto množinou bodů. Vzorce pro spojitou funkci by byla složitější, ale zároveň úplně zbytečná, protože programová implementace probíhá právě na diskrétní množině vzdáleností bodů X.

$$h_{total,j} = \sum_{i=2}^{n} h_j(x_i) - h_j(x_{i-1}); h_j(x_i) > h_j(x_{i-1}) \tag{5.14}$$

5.4 Model reflektující fyzikální zákonitosti

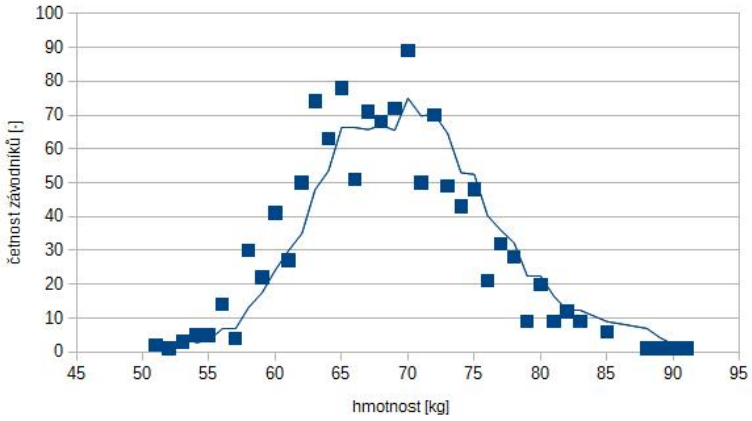
Funkce pro vypočtení obtížnosti terénu (5.13) sečte všechna stoupání a vyjádří je vzhledem k délce závodu l. Zanedbáváme tedy všechna klesání, respektive je považujeme za stejně náročné jako jízdu po rovině. Představme si dva závody, první po celou dobu vede do 1% stoupání, druhý po 9/10 závodu vede po rovině a zbylou 1/10 do kopce se sklonem 10%. Oba závody mají stejné převýšení, ale druhý závod bude, zejména v cyklistice, výrazně lépe vyhovovat závodníkům, kteří se cítí dobře v kopcovitém terénu, nežli závod první. Navržená funkce však nedokáže tyto případy rozlišit a pokud je k dispozici profil trasy, nevyužívá jeho potenciál dostatečně.

5.4.1 Cyklistický model

Nabízí se tedy možnost využít vzorec pro výpočet výkonu (3.4) v závislosti na profilu závodu. Tento postup však již platí pouze pro cyklistiku, v případě ostatních sportů by bylo potřeba zjistit obdobný vzorec a provést podobné kroky, které budou dále uvedeny. Nadále budeme pokračovat s již vytvořeným systémem vah, jen se pokusíme pro cyklistiku odvodit lepší váhu, s ohledem na výškový profil závodu.

Zmíněný vztah (3.4) však nelze přímo použít pro volbu této váhy. Vycházíme tedy stále z předpokladu, že dovednosti závodníků v závislosti na profilu závodu se mění. Schopnost zvládat těžké kopcovité terény mají především lehčí závodníci s vysokým poměrem mezi jejich výkonem a hmotností, naopak na rovinách vynikají závodníci s vysokým absolutním výkonem. Realita je samozřejmě o něco složitější a závodník může mít absolutní výkon vyšší jen v kopcích, nebo naopak na rovinách. Tento jev však nebývá nikterak zásadní, ale zejména na naše předpoklady nemá žádný vliv. Jednoduše vytvoříme prototyp závodníka, který zvládá dobře kopcovité závody a dalšího, který by měl být úspěšný především v rovinatých závodech.

Statistiky, které jsme získali obsahují 1180 cyklistů s uvedenou hmotností. Na obrázku 5.2 vidíme četnost závodníků s určitou hmotností. Četnost je zobrazena pomocí bodů a křivka představuje klouzavý průměr, vypočtený na základě 4 posledních hodnot. Nejlehčí závodník, z našich statistik, váží 51 kg, nejtěžší 91 kg a střední hodnota je rovna 68,35 kg. Hmotnosti na první pohled velmi dobře připomínají Gaussovo rozdělení. V rozpětí 58 až 78 kg se nachází hned 91,27% závodníků.



Obrázek 5.2: Četnost závodníků v závislosti na jejich hmotnosti

Nyní vytvoříme prototyp závodníka do profilově náročného terénu, který bude značen s indexy 1 a závodníka do lehkého terénu s indexem 2. Nejprve na základě rozdělení hmotností stanovíme celkovou hmotnost m, tedy včetně bicyklu (+7 kg), $m_1 = 65\ kg$ a $m_2 = 85\ kg$. Dále potřebujeme znát absolutní výkon závodníků P, se kterým jsou schopni závodit po dobu 20 minut až 1 hodiny. Výkon v tomto časovém horizontu se udává nejčastěji a označuje se jako výkon dlouhodobý. Průměrný výkon,

který jsou závodníci schopni vyvinout po dobu celého závodu, obvykle 4 hodin, je neznámý. Lze dohledat informace s výkony oscilujícími okolo 250 Wattů (zhruba 70% výkonu na 20 minut)[7]. Jenže tento údaj představuje spíše dolní mez, jelikož je obtížné najít závodníka, který celý závod vyvíjí maximální úsilí. Lze se domnívat, že bude u většiny závodníků výrazně vyšší a pro volbu koeficientů nám plně postačí hodinová hodnota. Vrchař s 58 kg by tedy mohl mít absolutní výkon $P_1 = 370\,W$, což i potvrzují naměřené hodnoty při zdolávání Alpe d'Huez [7], [15]. Takové parametry velmi přesně odpovídají vítězi Giro d'Italia a Vuelty Nairu Quintanovi.

Máme stanovené parametry pro prvního závodníka (vrchaře). Nyní nám zbývá stanovit výkon P_2. První prototyp cyklisty bude mít k-krát větší relativní výkon v poměru k hmotnosti, než druhý. A druhý naopak k-krát větší absolutní výkon, než první závodník. Potřebujeme tedy vyřešit 2 rovnice

$$P_1 k = P_2; \frac{P_1}{m_1} = k\frac{P_2}{m_2}; k > 1. \tag{5.15}$$

Zřejmě tedy $k = \sqrt{\frac{m_2}{m_1}} = 1,1435; P_2 = kP_1 = 423,11[W]$. Hodnoty druhého závodníka zase korespondují s naměřenými hodnotami Fabiana Cancellary při vítězné časovce na olympijských hrách v Brazílii. Cancellara projel trasu s průměrným výkonem 440 Wattů při hmotnosti 80kg (87 kg včetně kola)[16].

Předpokládejme rovnoměrný výkon těchto prototypů cyklistů po celou délku závodu a zároveň zanedbáme výkon potřebný na zrychlení (zpomalení). Další parametry, potřebné k výpočtům, použijeme z kapitoly o výkonu ($CS = 0,265$ pro časovku a $CS = 0,356$ pro hromadný závod). Hustotu vzduchu $\rho = 1,18$ zvolíme konstantní pro všechny nadmořské výšky. Zbývá stanovit celkový čas v j-tém závodě $t_{1,j}$ pro první prototyp závodníka a $t_{2,j}$ pro druhého. Tyto časy stanovíme podle vztahu pro čas závodníka v j-tém závodě

$$t_j(CS, m, \rho, P) = \sum_{k=2}^{n} \frac{x_k - x_{k-1}}{v(CS, \rho, m, P, s_l = \frac{h(x_k)-h(x_{k-1})}{x_k - x_{k-1}})}, \tag{5.16}$$

kde CS je součin potřebný k výpočtu odporu vzduchu, m je hmotnost závodníka včetně kola, ρ představuje hustotu vzduchu, P výkon závodníka, $h_j(x \in X_k)$ výškový profil pro j-tý závod a funkce $v(CS, \rho, m, P, s_l)$ počítá rychlost závodníka dle uvedených parametrů ve vybraném úseku s konstantním sklonem vozovky s_l dle vzorce

(3.5). Spočtení výsledného parametru obtížnosti pro vybraný závod provedeme dle vztahu

$$h_{power,j} = \begin{cases} \frac{t_{j,2}-t_{j,1}}{t_{j,1}}; t_{j,2} > t_{j,1} \\ -\frac{t_{j,1}-t_{j,2}}{t_{j,2}}; t_{j,1} \geq t_{j,2} \end{cases} \tag{5.17}$$

Pokud tedy prototyp vrchaře bude rychlejší $h_{power,j} > 0$, v opačném případě $h_{power,j} < 0$. Nabízela se i možnost použít prostý podíl $h_{power,j} = \frac{t_{j,2}}{t_{j,1}}$ pak by za podmínky $t_{j,1} < t_{j,2} h_{power,j} \in (0, 1)$ a opačně $t_{j,2} \geq t_1 h_{power,j} \in \langle 1, \infty)$. Hrozilo by tedy v extrémních případech, že by rozdělení obtížností terénů mohlo být špatně rozprostřeno a potlačovat význam profilově lehčích závodů. Při naší volbě hmotností a výkonů prototypů závodníků by však tento problém mohl nastat jen ve velmi omezeném měřítku, jelikož jejich rozdíly parametrů nejsou příliš vysoké.

Výslednou váhu vzhledem k predikovanému závodu následně stanovíme jako

$$\begin{aligned} h_{distance} &= h_{max} - h_{min} \\ w_{power,j}(k) &= \left(\frac{|h_{power,j} - h_{power,n+1}|}{h_{distance}}\right)^k, \end{aligned} \tag{5.18}$$

kde h_{max} je nejvyšší hodnota obtížnosti napříč všemi závody (včetně predikovaného) ze vzorce (5.17), h_{min} je naopak hodnota minimální ze stejného vzorce a stejné množiny závodů. k je opět koeficient pro zvýšení rozdílů mezi jednotlivými profily.

Navržená metoda by měla v případě dostatečného počtu různých závodů velmi dobře reflektovat náročnost terénu. Spoléhá však na rovnoměrný výkon po celou dobu závodu a také využívá vzorec pro závodníka, který jede osamocen. Tyto podmínky jsou velmi dobře splněny při časovce, kde závodníci nemohou využívat závětří za jiným závodníkem. A navíc výkon opravdu zůstává relativně stejný po celou dobu závodu [16]. U hromadných závodů by měl nastíněný postup přinést také zlepšení. Závodníci ovšem obvykle jedou většinu závodu ve skupině a i jejich výkon se v čase výrazně mění. Při hromadných závodech s ohledem na spolupráci závodníků v hlavní skupině jsou významnější stoupání až v závěru závodu, což však navržený postup rovněž nereflektuje.

5.5 Další typy modelů

Prozatím jsme vytvořili model, který reflektuje pouze předchozí umístění závodníků, a na jejich základě vytváří četnostní funkce. Tento přímý poziční model může být nepříznivě ovlivněn rozdílným počtem startujících v jednotlivých závodech. Základní myšlenkou je ponechat metodiku modelu, která byla dříve uvedena, jen pozměnit vstupní data četnostní funkce $p'_i(x)$.

5.5.1 Relativní poziční model

Poziční model může vykazovat chyby, pokud startovní listina často obsahuje rozdílný počet závodníků. Lze předpokládat, že získat dobré umístění mezi více závodníky, je obecně obtížnější. Mírně modifikujme četnostní funkci pro poziční model (5.7) a získáme

$$p'_i(\frac{x}{|S_j|}) = \sum_{j=1}^{n} r_{weight}(x, i, j); x \in \{1, ..m\}, \tag{5.19}$$

kde $i \in S$ a $m \in N$ označuje nejhorší umístění závodníků v předcházejících závodech a S_j je startovní pole j-tého závodu.

5.5.2 Modely založené na výsledném čase

Tyto modely využívají pouze výsledných časů závodníků, bez jakékoliv závislosti na výsledcích soupeřů. Z historických časů vybraného závodníka se určí četnostní funkce jeho predikovaného času a z něj se následně vypočte predikce jeho umístění.

V ideálním případě by výsledky v modelu měly pro jednoho závodníka a daný sport, či jeho oddělenou kategorii, být při jeho ustálené výkonnosti velmi podobné. V opačném případě bude model vykazovat velmi špatné výsledky. Pro tento přístup je tedy důležité nalézt takové sporty, které tomuto předpokladu vyhoví. Hledáme takové sporty, jejichž výsledky nejsou příliš ovlivněny taktikou ani počasím.

Bohužel se jedná o všechny sporty, které práce přímo implementuje a zejména pak silniční cyklistiku, při které ovlivňuje výsledný čas, počasí i taktika velmi výrazně. Cyklistika je tedy typickým představitelem sportů, který je pro tento přístup nevhodný.

Model je zaměřený na velmi úzký okruh sportů. Vhodnými kandidáty by mohli být sprinterské tratě. Dále i vrhačské disciplíny, při kterých navíc závodníci startují odděleně a pravděpodobně se tak ještě omezí taktické pojetí. Naopak v případě nalezení vhodného sportu model nemusí řešit závislosti mezi jednotlivými závody.

Tento typ modelů je naprosto identický s popsanou metodikou modelů pozičních, jen je třeba nahradit funkci (5.7) následující

$$p'_i(x) = \sum_{j=1}^{n} t_{total,i,j} w_j; x \in \{1, ..m\}, \tag{5.20}$$

kde $t_{total,i,j}$ je výsledný čas pro i-tého závodníka v j-tém závodě.

Modely využívající časový odstup

Další kategorii představují modely, které využívají časový odstup na vítěze závodu. Tento přístup je navržen tak, aby se pokusil zanedbat rozdíly způsobené počasím, a dalšími vlivy. Znalostí pouze odstupu na vítěze, klademe předpoklad, že vítězný čas by měl být ve všech závodech stejně kvalitní, ačkoliv může být rozdílný.

Představený koncept selhává zejména v případech, kdy časové rozestupy špatně reflektují výkonnost závodníků. K tomu dochází především při malých odstupech mezi závodníky. Model tedy nepřinese dobré výsledky pro cyklistické závody, které končí sprintem velké části pelotonu. Naopak lze použít zejména v případech, kde není možné použít model predikující výsledný čas kvůli změnám počasí. Lze předpokládat, že se uplatní především pro závody s individuálním startem.

Opět tedy nahradíme z původního pozičního modelu funkci (5.7)

$$p'_i(x) = \sum_{j=1}^{n} t_{loss,i,j} w_j; x \in \{1, ..m\}, \tag{5.21}$$

. kde $t_{loss,i,j}$ je časová ztráta na vítěze pro i-tého závodníka v j-tém závodě.

Relativní časový odstup vzhledem k délce závodu

Dále lze předpokládat, že se při větší délce závodu l_j zvětší i rozdíly mezi jednotlivými závodníky. Časovou ztrátu tedy budeme sledovat relativně vzhledem k délce závodu

$$p'_i(x) = \sum_{j=1}^{n} \frac{t_{loss,i,j}}{l_j} w_j; x \in \{1, ..m\}, \tag{5.22}$$

.

5.6 Programová implementace modelů

Základ pro modely je opět společný a nachází se v balíčku *system.models*. Třída *DiscreteDistribution* představuje diskrétní rozdělení a umí přidat pravděpodobnost na určitou pozici rozdělení, nebo normalizovat celé rozdělení, případně vypočíst konvoluci s jiným rozdělením. Třída *RaceResultsPrediction* pro každého závodníka nabízí vlastní distribuční funkci a také poskytuje funkcionalitu, která počítá výsledné pravděpodobnosti pomocí metody Monte Carlo.

Všechny modely musí implementovat rozhraní *I_Model*, které má jedinou metodu *predictResult*. Metoda vrací objekt třídy *RaceResultsPrediction* s generickými parametry, jež dědí od tříd *A_Statistics*, *A_StartList*, *I_Race*. Model tedy musí po předložení startovní listiny, závodu a statistik, umět predikovat výsledky pro každého závodníka ze startovní listiny.

5.6.1 Obecné modely

Obecné modely musí fungovat pro všechny sporty, splňující naše základní předpoklady. A je pro ně vytvořeno speciální rozhraní *I_GeneralModel*, které implementuje další rozhraní *I_Model* a generické parametry jsou nastaveny na nejvzdálenějšího možného předka. Všechna sportovní odvětví, jež jsou schopny být implementovány podle navrženého rozhraní, popsaného u objektového návrhu statistik, zároveň mohou být predikovány libovolným modelem implementujícím rozhraní *I_GeneralModel*.

Vytvořit výsledný model je již velice snadné. Pomocí následujícího pseudokódu si ukážeme, jak je metoda modelu implementována.

```
metoda predikujVysledky(statistiky, startovni_listina, predikovany_zavod){
    seznamRozdeleni = novy SeznamRozdeleni();
    pro všechny (zavodnik = startovni_listina.dejVsechnyZavodniky()){
        rozdeleni = nove rozdeleni();
        pro všechny (zavod = statistiky.dejVsechnyZavody()){
            vaha = vypocetVahy();
```

```
            pozice = statistiky.dejVysZavodu(zavod).dejUmisteni(zavodnik);
            rozdeleni.pridejNaPozici(pozice, vaha);
        }
        rozdeleni.normalizovat();
        seznamRozdeleni.pridat(rozdeleni);
    }

    vrat metodaMonteCarlo(seznamRozdeleni, pocetSimulaci);
}
```

V reálné implementaci se sice musí řešit ošetření výjimek a případy, kdy závodník nemá žádné výsledky, ale v zásadě se jinak příliš neliší. Model, vyvinutý přímo pro cyklistiku, pak využívá jen jiné rozhraní, respektive jiné generické parametry.

6 Vyhodnocení modelů

6.1 Metriky pro vyhodnocení modelů

Ke srovnání úspěšnosti vytvořených modelů je třeba navrhnout základní metriky, podle kterých zjistíme, jaké modely vykazují nejlepší výsledky. Porovnávat výsledky samozřejmě lze jen na již proběhlých závodech.

6.1.1 Vyhodnocení jednotlivých závodů

Pro zjednodušení značení, bude v této kapitole predikovaný závod označován jako j-tý s ohledem na již nadefinované funkce. U každého závodu po provedení predikce, získáme výsledky pro i-tého závodníka ze startovní listinu S_j v podobě pravděpodobnostní funkce $p_i(x)$. A zároveň máme k dispozici výsledky $r_{top,i,j}$ a $r_{down,i,j}$. Opět se jedná o nejlepší a nejhorší umístění závodníka, pro případ dělených pozic.

Měřítkem, které vystihuje úspěšnost predikce v j-tém závodě, může být součet pravděpodobností na správně predikovaném umístění

$$p_{s,j} = \sum_{i \in S} \sum_{r=r_{top,i,j}}^{r_{down,i,j}} r_{weight}(r,i,j) p_i(r), \tag{6.1}$$

přičemž funkce $r_{weight}(r,i,j)$ je definována vzorcem (5.4) Hodnota $p_{s,j}$ nabývá hodnot z intervalu $\langle 0, |S_j| \rangle$. Nicméně hodnoty 1, by dosáhl již model s rovnoměrným rozdělením mezi všechna možná umístění, stejnou hodnotu by tedy podle zákonu velkých čísel evidentně v průměru vykazoval i zcela náhodný model. Při větším počtu startujících může metrika dosahovat vyšších hodnot a je nutné s tím počítat při vyhodnocení výsledků. Nabízela by se tedy možnost vzorec dělit počtem závodníků, jenže ani tento postup by nebyl zcela objektivní. S rostoucím počtem závodníků ve startovní listině se obvykle velmi zvyšuje náročnost přesné predikce.

Tabulka 6.1: Závislost hodnoty $p_{r,j}$ na počtu závodníků

počet závodníků	$p_{r,j}$ (náhodný model)	$max(p_{r,j})$
10	0,29	2,93
20	0,18	3,6
80	0,062	4,97
160	0,035	5,66

Již bylo zmíněno, že v silniční cyklistice nejsou závodníci motivování bojovat o umístění na chvostu startovního pole. Význam umístění za elitní dvacítkou, lze většinou považovat za velmi nízký. Přesto se jedná zhruba o $\frac{7}{8}$ celého startovního pole. Tento jev nastává zejména v silniční cyklistice, ale může se vyskytovat i v jiných sportovních odvětvích. Upravíme funkci s ohledem na umístění a získáme

$$p_{r,j} = \sum_{i \in S} \sum_{k=r_{top,i,j}}^{r_{down,i,j}} \frac{r_{weight}(k,i,j)p_i(k)}{k}. \tag{6.2}$$

.

Opět se podívejme na obor hodnot, zřejmě $p_{r,j} \in \langle 0, \sum_{i=1}^{|S_j|} \frac{1}{i} \rangle$. V případě rovnoměrného rozdělení pravděpodobností pro všechny závodníky bude $p_{r,j} = \sum_{i=1}^{|S_j|} \frac{1}{i|S_j|}$.

Tabulka 6.1 zobrazuje závislost maximální hodnoty $p_{r,j}$ a průměrné $p_{r,j}$ pro náhodný model v závislosti na počtu závodníků. Na vývoji hodnot pro náhodný model vidíme, že s přibývajícím počtem závodů se $p_{r,j}$ snižuje.

Sledované hodnoty $p_{r,j}$ i $p_{s,j}$ jsou závislé na počtu závodníků ve startovním poli a je třeba s tím počítat. Tyto hodnoty tedy lze úspěšně porovnávat mezi modely, ale výhradně na stejných závodech.

6.1.2 Kompletní vyhodnocení závodů

Jsou vytvořeny metriky z jednotlivě predikovaných závodů. Nyní je třeba navrhnout postup při vyhodnocení nad všemi predikovanými závody.

Mějme tedy k dispozici množinu již proběhlých závodů $Z = \{z_1, z_2, ..z_n\}$, které jsou uspořádány podle data konání. Abychom získali co možná nejvíce výsledků z predikce, postupně budeme predikovat všechny závody z této množiny, s výjimkou prvního. První závod nemá význam predikovat, jelikož k němu nemůžeme získat

žádné výsledky, na jejichž základě bychom jej predikovali. Často však bude vynecháno větší množství závodů, aby model získal možnost standardně pracovat. Nyní, pokud budeme chtít predikovat závod $z_j, j \in \{2, ..n\}$ zúžíme trénovací množinu (závody, jejichž výsledky jsou k dispozici pro modely) závodů tak, že $Z' = \{z_1, ..z_{j-1}\}$.

První metrikou nad množinou predikovaných závodů je aritmetický průměr

$$\overline{x} = \frac{1}{n} \sum_{j=1}^{n} x_j, \tag{6.3}$$

kde n je počet predikovaných závodů a x_j sledovaná metrika j-tého predikovaného závodu, dle předchozí kapitoly. Aritmetický průměr však může být vychýlen hodnotami, které se výrazně odlišují od ostatních. Z tohoto důvodu budeme sledovat i medián

$$\hat{x} = \begin{cases} x_{\frac{n+1}{2}}; n \not| 2 \\ \frac{1}{2}(x_{\frac{n}{2}} + x_{\frac{n}{2}+1}); n \mid 2 \end{cases}, \tag{6.4}$$

kde $x_1 < x_2 < .. < x_n$.

Při testování budeme sledovat medián a aritmetický průměr nad metrikami j-tých závodů $p_{s,j}$, p_{r_j}. A budeme je značit $\overline{p_s}$, $\hat{p_s}$, $\overline{p_r}$, $\hat{p_r}$.

Mimo uvedený medián a aritmetický průměr pro vybraný model nad metrikou x může být přínosný i počet úspěšněji vyhodnocených závodů mezi jednotlivými modely. Nechť máme statistické modely $M = \{M_1, ..M_m\}$, které byly vyhodnoceny s metrikou $x_{k,j}$ pro j-tý závod a k-tý model. Pak definujeme funkci $m(M_i, M, y, x)$, která určí počet závodů, kdy se model M_i umístil na y-tém místě mezi modely M na základě sestupného seřazení metriky x.

Navržené metody vyhodnocení nemusí nutně zaručit, že model s dobrými výsledky je skutečně kvalitní. Základním předpokladem všech modelů je, že predikují pravděpodobnost na vybraná umístění. Je třeba tento předpoklad ověřit. Měli bychom tak odhalit zejména případy, kdy je na některá umístění přidělena příliš vysoká, respektive nízká, pravděpodobnost. Myšlenka je jednoduchá, pokud přidělíme umístění určitou pravděpodobnost, na velkém vzorku predikcí by i četnost správně predikovaných umístění měla odpovídat.

Rozdělme pravděpodobnost do n nepřekvrývajících se intervalů $I = \{I_1, ..I_n\}$, které zároveň pokryjí celý interval $\langle 0, 1 \rangle$. Podle intervalů rozdělíme predikované

pravděpodobnosti $p_{i,j}(x)$ tak, že bude splněno $p_{i,j}(x) \in I_k$ a koeficient k intervalu, do kterého spadá tato predikce bude určovat funkce $q(p_{i,j}(x))$. V k-tém intervalu tedy budeme očekávat akumulovanou pravděpodobnost

$$p_a(k) = \sum_{j \in Z} \sum_{i \in S_j} \sum_{x=1}^{|S_j|} p_{i,j}(x); q(p_{i,j}(x)) = k, \tag{6.5}$$

kde Z je množina všech závodů a S_j startovní listina j-tého závodu.

Následně ke každému intervalu určíme skutečný počet správně určených umístění závodníků dle vztahu

$$c_a(k) = \sum_{j \in Z} \sum_{i \in S_j} \sum_{x=1}^{|S_j|} r_{weight}(x, i, j); q(p_{i,j}(x)) = k, \tag{6.6}$$

kde Z je množina všech závodů, S_j startovní listina j-tého závodu a funkce $r_{weight}(x, i, j)$ definuje váhu případným děleným pozicím dle vzorce (5.4). Zajímat nás bude i celkový počet predikcí v daném intervalu, který spočteme lehkou modifikací

$$c_t(k) = \sum_{j \in Z} \sum_{i \in S_j} \sum_{x=1}^{|S_j|} 1; q(p_{i,j}(x)) = k. \tag{6.7}$$

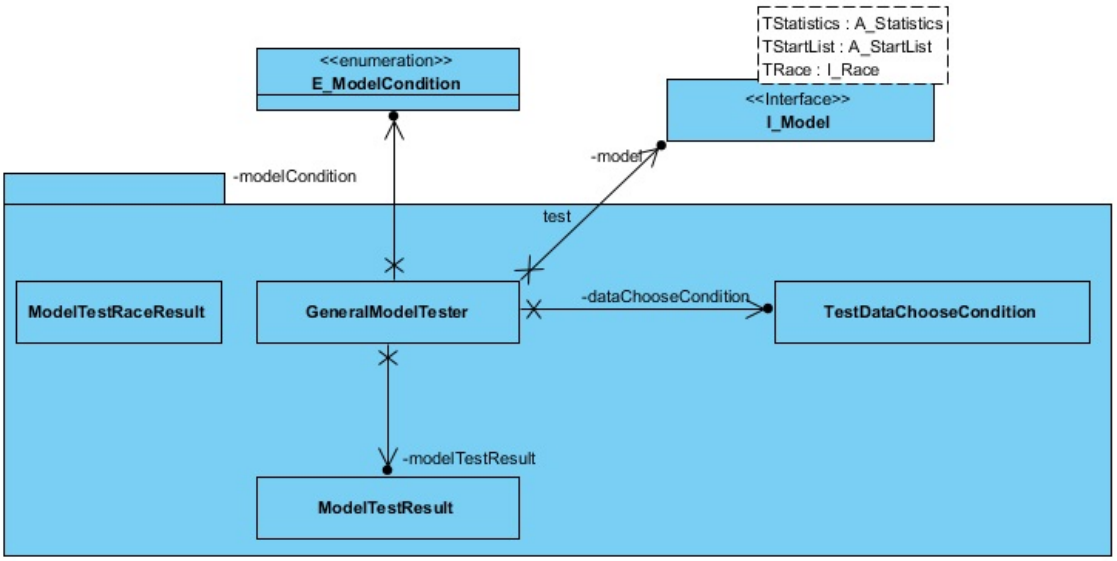
V některých případech budeme potřebovat 3 výše uvedené součty omezit umístěními v intervalu $\langle x_1, x_2 \rangle$. V takovém případě budou přidány 2 parametry pro každou funkci $(p_a(k, x_1, x_2), c_a(k, x_1, x_2), c_t(k, x_1, x_2))$. Výše uvedené vzorce pak změní svou poslední sumu na $\sum_{x=x_1}^{x_2}$.

Nyní máme k dispozici pro k-tý interval skutečné $(c_a(k))$ a teoretické četnosti $(p_a(k))$. Porovnávat zmíněné hodnoty pomocí testu dobré shody nepovede k dobré představě o úspěšnosti predikce, jelikož predikce není zdaleka tak přesná, aby bylo možné ji tímto způsobem porovnávat. Budeme potřebovat spíše porovnat rozdíly mezi těmito hodnotami.

6.2 Programová implementace

Na obrázku 6.1 je znázorněn diagram tříd pro vyhodnocení libovolného modelu. Všechny modely implementují stejnou metodu, proto zde není nutná tak vysoká míra abstrakce, jako u předešlých návrhů. Třída *ModelTestRaceResult* představuje

výsledky z jediného závodu, které jsou popsány v kapitole vyhodnocení závodu. Souhrnné výsledky o vybraném modelu napříč všemi otestovanými závody zpracovává třída *ModelTestResult*. *GeneralModelTester* pak zaštiťuje veškerou práci s vyhodnocením modelů. Nejprve mu je předána reference na konkrétní model, zadán sport, který má být testován a v jakém časovém období pomocí *TestDataChooseCondition*. Samotná třída *GeneralModelTester* implementuje jedinou metodu *run* z rozhraní *Runnable*, díky které ji lze následně použít ve spojení s vlákny. Metoda *run* načte potřebné statistiky, otestuje zadaný model a vypočte všechny zmíněné metriky pro jeho vyhodnocení.



Obrázek 6.1: Diagram tříd modelů

6.2.1 Uživatelské rozhraní pro testování modelů

Třída, která propojuje programátorský model a pohled, je u testování *TesterController*. Uživatel si může zvolit rozpětí data závodů, nad kterým má test probíhat. Toto rozpětí zajišťují komponenta *DateTimePicker* knihovny javafx. Uživateli je rovněž předložen seznam všech modelů a sportů, které může otestovat. Realizován je pomocí komponent *CheckBox*.

Po stisku tlačítka test, jsou požadované modely a sporty otestovány pomocí třídy *GeneralModelTester*. Každý model je přidělen do samostatného vlákna. Mezi všechna vlákna je sdílen objekt, kam každé vlákno zapíše výsledky svého testu.

Přístup k tomuto zdroji lze označit za kritickou sekci a je nutno ho synchronizovat, aby nedošlo k neočekávané chybě. Modely jsou vytvořeny pomocí třídy $ModelFactory$, která vytváří potřebné modely. Statistiky obdobně vytváří třída $StatisticsFactory$. Obě třídy jsou založeny na návrhovém vzoru factory [21].

Výsledky jsou prezentovány v tabulkách, komponenty $TableView$, které jsou data předávány ve formě seznamu podle návrhového vzoru observer [22]. V javě tento vzor implementuje třída $ObservableList$. Jsou zobrazovány jak výsledky pro jednotlivé závody, tak i výsledky celkové. Po vytvoření výsledků je můžeme filtrovat na základě sportovního odvětví, způsobu startu, pohlaví, rozpětí umístění, nebo vybírat jen některé modely. Filtrování probíhá ze seznamu všech výsledků a tabulkám jsou předány jen vyfiltrované seznamy. Pokud nejsou aktivní žádné filtry, je seznam překopírován v originální podobě.

Problematickým místem se ukázalo být testování výsledného rozdělení pravděpodobností, které probíhá dle vzorců (6.5), (6.6), (6.7). Vzhledem k požadavku filtrovat i tento test podle závodu a umístění byla nejprve navržena funkcionalita, která si ponechávala v operační paměti každou predikci (tzn. dvojici hodnot umístění a pravděpodobnost). Pro jediný cyklistický závod se vytvářelo až n^2 těchto dvojic pro n závodníků. Pokud je následně nutné testovat současně k modelů na m závodech, celkový počet uchovávaných dvojic se již rovná n^2km. U běžných počítačů může docházet operační paměť. Operační systém si s tímto problémem umí poradit a používá paměť pevného disku, který je však pomalý a dochází i k výraznému poklesu rychlosti výpočtů. Pokud bychom se chtěli vzdát možnosti filtrování, bylo by možné ukládat jen nasčítané hodnoty ze vzorců (6.5), (6.6), (6.7) u každého zkoumaného intervalu. Pro zastoupení těchto hodnot slouží třída $ProbabilityCounter$. Filtrovat výsledky však není potřeba na základě jednotlivých závodníků a tak je možné ukládat tyto součty hodnot pro každou možnou pozici v rámci každého závodu. Místo uchovávané dvojice, pracujeme s trojicí v podobě třídy $ProbabilityCounter$, ale již si vystačíme s nkm těchto tříd. Pokud u cyklistických závodů bývá počet závodníků $n = 200$, snížíme nároky na operační paměť až $\frac{2 \cdot 200}{3}$-krát.

6.2.2 Uživatelské rozhraní pro predikci výsledků

Grafické rozhraní k prezentaci výsledků je určené jak tvůrci modelů, tak ale i bookmakerovi, či běžnějšímu uživateli. Cílem je zobrazit uživateli pro vybraný sport a závod kompletní predikci ve srozumitelné podobě.

Uživatel si nejprve vybere jedno sportovní odvětví pomocí *ComboBoxu*, kam jsou načteny všechny sporty z výčtového typu *E_Sports*. Po zvolení sportu jsou načteny pomocí továrny *StatisticsFactory* veškeré statistiky. Prozatím statistiky nejsou nikterak velké, aby bylo nutné tento krok přehodnocovat a v nejbližším vývoji programu k tomu jistě nedojde.

Po dokončení načítání statistik se v pravé části obrazovky zobrazí, za pomocí komponenty *ListView*, podle data konání všechny dostupné závody. V prostřední části obrazovky jsou k dispozici všichni závodníci a jsou seřazeni abecedně, podle příjmení. V pravé části je startovní listina, která se objeví po kliknutí na libovolný závod. Startovní listinu lze rovněž libovolně měnit, pomocí nástrojů, které jsou implementovány. Startovní listinu lze jediným tlačítkem celou smazat, nebo smazat libovolného ze závodníků, kliknutím na jeho jméno a následně potvrzením tlačítka umístěného pod seznamem. Se seznamu všech závodníků pak lze vybrat libovolného závodníka a přesunout ho do startovní listiny. Tímto způsobem si uživatel může vytvořit libovolné startovní pole, pokud jsou potřební závodníci v databázi. Výchozí řazení je zajištěno databází, což je nejrychlejší možnost, případné další řazení obstarává standardní třída javy *TreeSet*.

Všech závodníků je velké množství, proto byly vytvořeny filtry. Uživatel si může zvolit národnost závodníka, v případě sportu, kde existují týmy i tým. Podle těchto kritérií se následně zobrazí jen vybraní závodníci. Výběr je realizován s pomocí hašovacích map, díky čemuž je výběr velmi rychlý a není tak nutné procházet všechny závodníky. Nakonec uživatel vybere konkrétní model a počet simulací pomocí metody Monte-Carlo.

Po odsimulování jsou uživateli nabídnuty souhrnné výsledky predikce. Základní přehled udává tabulka s pravděpodobnostmi umístění závodníků. V každém řádku je umístěn jeden závodník. Uživatel si může přát zobrazit libovolné umístění pro určitý sport a tak bylo vytvořeno pro každý sport zadávání požadovaných umístění, které

se mají ve výsledcích objevit. Pro každý sport si lze vybrat libovolné umístění, které se zadává ve formě od, do. Tyto údaje jsou ukládány a načítány z databáze. Proto je třeba definici tabulky s pravděpodobnostmi závodníků generovat dynamicky za běhu programu.

Kliknutím na libovolného závodníka, z tabulky s predikcí umístění, se na grafu zobrazí jeho distribuční funkce, v závislosti na predikovaných umístěních. Graf je vytvořen s pomocí třídy *LineChart* knihovny *javaFX* a v další tabulce se objeví pravděpodobnosti, že vybraný závodník porazí konkrétního závodníka. Všechny tabulky, tvořené pomocí třídy *TableView*, mohou být řazeny dle libovolného sloupce bez nutnosti psaní dalších funkcí.

6.3 Stanovení optimálního modelu a parametrů

Bylo nadefinováno několik základních typů modelů a systém vah závislý na parametrech závodů. Nyní je třeba stanovit vhodný typ modelu a ideální koeficienty pro systém vah. K tomu použijeme experimentální testování s vybranými koeficienty. Vhodné koeficienty pak vybereme s pomocí navržených metrik pro vyhodnocení modelů. K dispozici máme kompletní výsledky sezón 2013 až 2016.

6.3.1 Počet simulací

Model používá k predikci metodu Monte Carlo, je tedy nezbytné nastavit vhodný počet simulací. Nejlepší výsledky zřejmě vždy zaručí, co nejvyšší možný počet. Metoda je však poměrně výpočetně náročná a při simulování stovek závodů by trvala neúměrně dlouho. Časovou náročnost, mimo počtu simulací, ovlivňuje i velikost startovního pole. Potřebujeme tedy stanovit, co nejnižší možný počet simulací, který zároveň ještě zaručí dostatečně dobré výsledky.

Experimentálně otestujeme na 10ti závodech poziční model s váhou $w_j = 1$ a nastavíme různý počet simulací. Jako ideální můžeme označit výsledek vzorového modelu s nejvyšším počtem simulací a porovnáme, jak moc se mu ostatní modely přibližují. Budeme sledovat součty pravděpodobností a jejich rozdělení do jednotlivých intervalů podle vzorce (6.5). Tento údaj neříká nic o výsledcích modelu, ale ukazuje rozvrstvení pravděpodobností, a to postačí na zjištění podobnosti modelů,

Tabulka 6.2: Součet pravděpodobností dle vybraných intervalů

simulací	$\langle 0, 0.01)$	$\langle 0.01, 0.03)$	$\langle 0.03, 0.05)$	$\langle 0.05, 0.1)$	$\langle 0.1, 1\rangle$	ϕ
100	472.94	693.13	126.21	43.52	8.21	69.34%
300	602.15	644.43	64.78	33.1	5.54	21.01%
1000	697.59	565.28	52.22	29.57	5.34	6.96%
3000	734.72	531.72	49.97	28.72	4.88	2.99%
10000	749.74	518.28	48.65	28.52	4.8	1.96%
30000	752.41	515.6	48.77	28.23	4.98	0.82%
100000	755.21	512.67	48.97	28.06	5.09	0%

v závislosti na počtu simulací.

Ke zlepšení představy o rozdílech si nadefinujeme aritmetickou odchylku mezi součty pravděpodobností v jednotlivých intervalech. Máme množinu intervalů $I = \{I_1, ..I_k\}$, vzorový model se součtem pravděpodobnostní na i-tém intervalu $p_{a,v}(i)$ a testovaný model se součtem pravděpodobnostní na i-tém intervalu $p_{a,t}(i)$. Odchylku potom definujeme pomocí vzorce

$$\phi = \frac{\sum_{i=1}^{k} \frac{|p_{a,v}(i) - p_{a,t}(i)|}{p_{a,v}(i)}}{k}. \tag{6.8}$$

V tabulce 6.2 vidíme součty pravděpodobností dle intervalů jednoduchého pozičního modelu, v závislosti na počtu simulací metody Monte Carlo. Nadefinovaná odchylka ϕ tabulku zpřehledňuje a ukazuje, že zvyšující se počet simulací, dle očekávání, snižuje i tuto odchylku od vzorového modelu. 100 i 300 simulací zřejmě nepřináší dobré výsledky, naopak po 3000 simulacích se odchylka snižuje jen velmi málo. S ohledem na výpočetní náročnost zvolíme právě 3000 výpočetních kroků.

6.3.2 Základní typy modelů

Celkem bylo vytvořeno 5 typů modelů. Všechny otestujeme na sezónách 2015 a 2016, což představuje 285 závodů. V Tabulce 6.3 jsou zobrazeny výsledky tohoto testu. Sledovat v ní můžeme průměr a medián nad všemi závody metrik (6.1), (6.2). Podle metriky $p_{s,j}$ rovněž budeme sledovat pořadí modelů dle metodiky, která již byla dříve popsána. Tato umístění jsou v tabulce rovněž zaneseny.

typ modelu	$\overline{p_s}$	$\hat{p_s}$	$\overline{p_r}$	$\hat{p_r}$	1.	2.	3.	4.	5.
celkový čas	1.0026	1.0043	0.0174	0.0193	1	0	2	5	277
relativní čas k vzdálenosti	1.1542	1.1623	0.0224	0.02195	4	6	75	193	7
relativní čas	1.1747	1.1825	0.0258	0.0256	1	3	195	86	0
poziční	1.4155	1.4184	0.0769	0.07466	110	171	4	0	0
relativní poziční	1.424	1.4215	0.0779	0.0752	169	105	9	1	1

Tabulka 6.3: Test základních typů modelů na sezónách 2015,2016

Model založený na celkovém čase dosahuje v průměru součtu pravděpodobností, při správně určené pozici $\overline{p_s} = 1.0026$, medián je pak $\hat{p_s} = 1.0043$. Obě metriky jsou velice blízké 1, takže vykazují velmi podobnou úspěšnost, jakou by dosáhl náhodný model, respektive model s rovnoměrným rozdělením. Nelze s jistotou ani usuzovat, že je skutečně úspěšnější než náhodný model. Ve srovnání se všemi ostatními modely, byl v 277 z 285 závodů úplně nejhorší. Tento typ modelu zřejmě není pro silniční cyklistiku vhodný, což bylo již dříve předpokládáno, jelikož celkové časy jsou velmi různorodé.

Další 2 modely jsou založené na časovém odstupu od vítěze závodu a vykazují velmi podobné výsledky. Proti předpokladu je dokonce úspěšnější model, který nereflektuje v časových odstupech vzdálenost závodu. Pro oba modely platí $\overline{p_s} > 1$, $\hat{p_s} > 1$, a výrazněji, než u předchozího modelu, takže o nich s jistotou můžeme prohlásit, že oproti náhodným modelům již vykazují zlepšení.

Poziční modely dosahují nejlepších výsledků a ve velké většině případů ostatní typy modelů jasně převyšují. Rozdíl mezi pozičním a relativním pozičním je však tak malý, že nelze jednoznačně říci, který z nich se pro predikci v silniční cyklistice hodí více. Všechny zobrazené metriky v tabulce 6.3 velmi mírně favorizují relativní poziční, zejména v 169ti z 285 závodů měl nejlepší součet pravděpodobností $p_{s,j}$. S ohledem na ostatní metriky se lze domnívat, že druhý model byl často poražen jen velice těsně.

V reálném nasazení modelů nebude predikce výsledků závodníků, kteří končí na chvostu startovního pole tak zajímavá, použijeme tedy zabudované filtry programu a podíváme se na výsledky predikce jen pro předních $k \in N$ umístění. Výsledky jsou zobrazeny v tabulce 6.4 a opět jsou velmi vyrovnané. Všimněme si hodnot

model	k (pozice do)	$\overline{p_s}$	$\hat{p}_s$	1.	2.
relativní poziční	50	0.5353	0.5369	148	137
poziční	50	0.5354	0.5369	137	148
relativní poziční	10	0.1729	0.1801	139	146
poziční	10	0.174	0.1818	146	139
relativní poziční	3	0.0779	0.0753	149	136
poziční	3	0.0769	0.0747	136	149

Tabulka 6.4: Poziční a relativní poziční model filtrovaný podle umístění

$\overline{p_s}$ v závislosti na pozicích, do které jsou počítány. Zřejmě modely snadněji určují pravděpodobnosti pro přední umístění. U pozičního modelu připadá na první 3 místa $\overline{p_s} = 0.0769$, což je průměrně $0.0256\overline{3}$ na 1 pozici. Mezi 11. a 50. umístěním získáme $\overline{p_s} = 0.5353 - 0.174 = 0.3613$, což je průměrně jen 0.009 na 1 pozici. Lze se domnívat, že tento princip platí pro všechny modely a je složitější predikovat horší umístění.

Relativní poziční a poziční model, si jsou již v návrhu velmi podobné. Ani při detailnějším pohledu na výsledky testování těchto modelů, se nepodařilo jednoznačně prokázat, který model je úspěšnější. Data, která jsou testována, se týkají 2 nejvyšších kategorií cyklistických závodů, a počet startujících závodníků je velmi podobný, proto si pravděpodobně jsou i oba modely tak blízké.

Pro dokonalé pochopení specifik a nedostatků navržených modelů, se podíváme na poziční model a vyhodnocení pravděpodobností rozdělených do jednotlivých intervalů. V tabulce 6.5 jsou uvedeny pro jednotlivé intervaly dvojice hodnot: skutečný počet správně určených umístění $c_a(k)$ / očekávaný součet pravděpodobností $p_a(k)$. Tyto hodnoty by v ideálním případě, měly být prakticky totožné. V prvním řádku tabulky vidíme kompletní výsledky pro všechny pozice. Zejména první interval $\langle 0, 0.001)$ neodpovídá a očekávaný počet 279.84 úspěšných predikcí byl takřka 9ti násobně překonán. I většina dalších intervalů vykazuje velmi nepřesně určené pravděpodobnosti. Na dalších řádcích tabulky jsou postupně ukázány výsledky, omezené pozicemi a úspěšnost predikce se postupně zlepšuje. Predikce mezi 1. až 10. místem, s výjimkou prvního intervalu, vykazuje maximální odchylku mezi dvojicí hodnot 24,5%. Modelem stanovená pravděpodobnost zřejmě u těchto pozic je mnohem přesnější, než u horších umístění. Interval $\langle 0, 0.001)$ však zůstává problematický pro všechna možná

od	do	$\langle 0, 0.001)$	$\langle 0.001, 0.01)$	$\langle 0.01, 0.03)$	$\langle 0.03, 0.05)$	$\langle 0.05, 0.1)$	$\langle 0.1, 1\rangle$
vše	vše	2458/279.84	33047/30065	12452/15755	491/897.15	152/396.8	19/36.68
50	vše	1735/181.45	25703/23100	6868/9456.56	166/499.74	31/222.15	1/12.32
11	49	498/72	6453/6068.99	4272/4846.39	33/83.24	8/33.96	1/3.71
1	10	225/26.39	891/895.72	1312/1452.37	292/314.17	113/140.69	17/20.66

Tabulka 6.5: Rozdělení úspěšnosti pozičního modelu podle intervalů

umístění. Tento nepříznivý jev zřejmě plyne z umístění, která jsou predikována s nulovou pravděpodobností. Pokud model nemá obrovské množství historických výsledků, může některá umístění pro predikovaného závodníka označit za nemožná. Žádné umístění však jistě není nemožné a z toho pak plyne tato obrovská odchylka ve výsledcích predikce. Metoda Monte Carlo a relativně malý počet simulací tomuto jevu rovněž výrazně přispívá. Pokud by měla být predikovaná pravděpodobnost velmi nízká, metoda Monte Carlo při nízkém počtu simulací často dojde k nulové pravděpodobnosti.

6.3.3 Redukce pravděpodobností

Tabulka 6.5 zároveň poskytuje návod, jak je možné špatné rozdělení pravděpodobností řešit. Pokud jsou pravděpodobnosti na určitém intervalu k-krát menší (větší), nabízí se možnost každou pravděpodobnost, z daného intervalu, vynásobit (vydělit) právě koeficientem k. Modelu jsme však předepsali dvě nutné podmínky (5.1), (5.2), které by po navržené úpravě nemuseli být splněny. Proto využijeme základní funkčnost modelů a navržený postup mírně modifikujeme.

Mějme výsledky modelu a množinu intervalů $I = \{I_1, I_2, ..I_n\}$ a ke každému intervalu koeficient

$$k_l = \frac{p_{a,l}}{c_{a,l}}, l \in \{1, ..n\}, \tag{6.9}$$

kde $p_{a,l}$ je součet pravděpodobností pro l-tý interval definovaný v (6.5) a $c_{a,l}$ je počet skutečně úspěšných predikcí v l-tém intervalu dle (6.6). Pro i-tého závodníka máme stanovenou pravděpodobnostní funkci $p_i(x)$, kterou nyní pomocí daných koeficientů pozměníme tak, že

$$p'_i(x) = p(x)k_l, \tag{6.10}$$

$\overline{p_s}$	$\langle 0, 0.001)$	$\langle 0.001, 0.01)$	$\langle 0.01, 0.03)$	$\langle 0.03, 0.05)$	$\langle 0.05, 0.1)$	$\langle 0.1, 1\rangle$
1.291	58/43.42	38817/37128	9388/10202	78/92	18/12.18	0/0

Tabulka 6.6: Rozdělení a úspěšnost pravděpodobností po redukci

kde k_l je koeficient intervalu, který obsahuje $p(x)$. Násobit nulové pravděpodobnosti zřejmě k dobrým výsledkům nepovede, a proto pravděpodobnosti spadající do 1. intervalu budou mít přiřazenou pravděpodobnost $p'_i(x) = \frac{c_{a(1)}}{c_{t(1)}}$. Funkci $p_i(x)'$ pro každého závodníka normalizujeme, získáme tedy $p''_i(x)$ podle (5.6). Normalizované funkce opět považujeme za nezávislou a použijeme metodu Monte Carlo k predikci konečných výsledků.

Pro ověření, jestli navržená metoda může být úspěšná, vezmeme koeficienty z již proběhlého testu na sezónách 2015-2016. A opět otestujeme stejnou sadu závodů, ale s použitím redukce pravděpodobností. V tabulce 6.6 vidíme výsledky. Pravděpodobnost je již zřejmě mnohem lépe rozdělena, celková úspěšnost modelu se však výrazně snížila. Pravděpodobnosti totiž byly rozmělněny mezi různé pozice, bez hlubší znalosti souvislostí, zejména se jedná o původně nulové pravděpodobnosti. Po redukci žádné umístění nebylo predikováno s větší než 10% pravděpodobností.

Při reálném nasazení samozřejmě není možné použít koeficienty vypočítané ze závodů, jejichž výsledky ještě neznáme. Musíme se omezit na závody jen předcházející. Jelikož se i tyto koeficienty budou vyvíjet, bereme v potaz jen k závodů, které předchází predikovanému.

Tabulka 6.5 ukazuje, že úspěšnost predikce je závislá i na predikovaném umístění závodníků. Proto i redukce umístění může probíhat v závislosti na predikovaném umístění x.

$$p'_i(x) = p(x)k_l(x-\alpha, x+\alpha); k_l(x-\alpha, x+\alpha) = \frac{c_a(l, x-\alpha, x+\alpha)}{p_a(l, x-\alpha, x+\alpha)}; l \in \{1, ..n\}, \quad (6.11)$$

kde redukční koeficienty jsou nyní stanoveny i v závislosti na umístění a koeficient α říká, kolik okolních pozic má být zahrnuto při výpočtu redukčních koeficientů a pro pravděpodobnosti spadající do prvního intervalu opět $k_l(x-\alpha, x+\alpha) = \frac{c_{a(1,x-\alpha,x+\alpha)}}{c_{t(1,x-\alpha,x+\alpha)}}$.

Nyní se podívejme na reálné nasazení redukční metody s i bez závislosti na umístění závodníků. Opět byl použit základní poziční model s $w_j = 1$ a otestován na sezóně 2016, předchozí sezóna byla použita pro stanovení koeficientů. V tabulce

$\overline{p_s}$	$\langle 0, 0.001)$	$\langle 0.001, 0.01)$	$\langle 0.01, 0.03)$	$\langle 0.03, 0.05)$	$\langle 0.05, 0.1)$	$\langle 0.1, 1\rangle$
1.309	317/84.4	19475/18265	4905/5790	95/94.01	39/33.09	1 / 2.06
1.279	290/73.2	19543/18382	4947/5753.3	45/52.86	7/4.46	0/0

Tabulka 6.7: Úspěšnost redukčních metod

6.3.3 vidíme v 1. řádku výsledky redukce v závislosti na pozicích a volbou koeficientu $k_1 = 10$. V druhém řádku jsou zobrazeny výsledky běžné redukční metody, u obou metod byl koeficient redukce počítán na základě předchozích 90 závodů. Správné určení pravděpodobností je u obou metod velmi podobné, nicméně metoda závislá na pozicích je celkově úspěšnější, což potvrzuje metrika $\overline{p_s}$.

Důležitou otázkou zůstává, zde je důležitější, aby byl vysoký součet pravděpodobností správně určených pozic $\overline{p_s}$, nebo je potřebné, aby pravděpodobnosti odpovídaly v závislosti na jednotlivých intervalech, očekávání. My se dále zaměříme především na metriku $\overline{p_s}$, potažmo $\hat{p_s}$ pro prvních 10 závodníků a zároveň budeme požadovat, aby pravděpodobnosti alespoň přibližně odpovídaly.

Při vyhodnocení nás bude často zajímat odchylka mezi reálnými a předpovídanými pravděpodobnostmi. Mějme opět intervaly $I = \{I_1, ..I_n\}$ a součty pravděpodobností v k-tém intervalu $p_a(k)$, dále součty správných predikcí v k-tém intervalu $c_a(k)$. Odchylku pak spočteme dle vzorce

$$\epsilon = \frac{\sum_{k=2}^{n} \frac{max(c_{a(k),p_a k})}{min(p_a(k),c_a k)}}{n-1} - 1. \tag{6.12}$$

V odchylce tedy vynecháváme první interval, jelikož často velmi zkresluje situaci, zároveň pokud je velmi špatně určený, tato chyba se stejně promítne i v dalších intervalech.

Při základním testování parametrů a jejich koeficientů, používaných pro výpočet váhy, budeme používat vždy jen jeden parametr závodu, abychom mohli posoudit jeho přínos a neovlivnili srovnání jiným parametrem.

6.3.4 Typy závodů a úrovně

V silniční cyklistice lze závody dělit podle typu, na časovky a hromadné. Lze se domnívat, že nemá velký význam na základě výsledků časovek, predikovat výsledky

w_j	$\overline{p_s}$	$\hat{p}_s$	$\langle 0.001, 0.01)$	$\langle 0.01, 0.03)$	$\langle 0.03, 0.05)$	$\langle 0.05, 0.1)$	$\langle 0.1, 1\rangle$	ϵ
1	0.174	0.1795	376/382.98	561/627	234/253	110/130	18/18.2	8.2%
$w_{level}(0)$	0.18244	0.18422	362/326.36	514/586	222/276	136/171.2	28/53.5	33.2%
$w_{start}(0)$	0.2039	0.19377	343/338	519/591	228/259	130/164.6	55/59	12.6%

Tabulka 6.8: Vyhodnocení modelů vytvořených s parametry typu a úrovně závodu

hromadných závodů a opačně. Proto vytvoříme poziční model s parametrem $w_j = w_{start}(0)$. Stejně tak vytvoříme model, který předpokládá nulový vztah mezi výsledky napříč úrovněmi $w_j = w_{level}(0)$.

V tabulce 6.8 vidíme výsledky tohoto testu. Základní model s váhou $w_j = 1$ měl nejnižší odchylku, ale naopak nejhorší úspěšnost predikce. Jednoznačně nejlépe z tohoto testu vychází model oddělující závody podle způsobu startu, odchylka zůstává velmi podobná, ale jeho úspěšnost je výrazně vyšší. Naopak parametr, odlišující úrovně závodu, byl poměrně neúspěšný, zejména odchylka je poměrně vysoká. V databázi máme zaneseny jen závody nejvyšší kategorie World Tour, které se dále dělí na dvě kategorie. Nicméně jsou si zřejmě velmi podobné a nedostatek dat, při jejich oddělení mohl způsobit vyšší odchylku.

6.3.5 Délka závodu

Délka závodu se zdá, že nehraje příliš velkou roli v silniční cyklistice. Jak metoda $w_{length(k)}$ i $w_length2(k)$ vykazují poměrně špatné výsledky. U první metody při koeficientu $k = 2$ získáme $\overline{p_s} = 0.1829$, přitom odchylka $\epsilon = 32\%$. Ve srovnání s bezparametrovým modelem nezískáme takřka žádné zlepšení, přitom odchylka je již poměrně vysoká. Silniční cyklisté se obvykle nespecializují na nějakou vzdálenost. Často se první fáze závodu jede ve vytrvalostním tempu a rozhoduje se až v závěrečné. V takovém případě pravděpodobně nehraje velkou roli, jestli závodníci ujedou 150, či 200 km. Nejdelším silničním závodem je Milan-San Remo, které je dlouhé 272 km. Zde se podle ohlasů závodníků tato vzdálenost již skutečně výrazně promítá. Byl vytvořen model s koeficienty $K = \{1, 2, 5, 10, 20\}$, které se použily pro metodu $w_{length}(k)$. Simulovány byly 3 závody Milan-San Remo, úspěšnost $\overline{p_s}$ se sice zvyšuje ale zároveň se zvyšuje i odchylka ϵ. 3 závody jsou navíc příliš malý vzorek, abychom mohli udělat nějaký směrodatný závěr. Délka závodu z testů jistě nějaký význam

k	$\overline{p_s}$	$\hat{p_s}$	ϵ	1.	2.	3.	4.
0.1	0.2153	0.1797	80%	21	4	3	13
0.01	0.1833	0.1804	32%	15	15	5	6
0.001	0.1671	0.1783	9.7%	1	12	23	5
0	0.1649	0.1754	10%	4	10	10	17

Tabulka 6.9: Základní koeficienty data pro cyklistiku

má, pravděpodobně ale je nízký a výrazně zvyšuje výslednou odchylku ϵ.

6.3.6 Forma závodníků

Výkonnost závodníků se v závislosti na čase mění, proto jsme vytvořili funkce $w_{ex\_date}(k)$ a $w_{date}(k)$. Koeficient k pro každou funkci zvolíme tak, že nejprve vybereme několik koeficientů $K = \{k_1, k_2, ..k_n\}$, podíváme se na výsledky, určíme 2 nejlepší koeficienty a následně otestujeme další koeficienty, které se mezi nimi nacházejí. Budeme tak postupně zužovat interval s ideálními koeficienty. Ideální koeficienty však budou rozdílné pro různé druhy závodů. Dále se zaměříme na nejpopulárnější etapový závod, Tour de France. Predikovat budeme pouze jednotlivé etapy, nikoliv celý závod. Koeficienty by se pravděpodobně měly měnit i na jednotlivé etapy, budeme však hledat takové koeficienty, které budou shodné po celou dobu všech etap.

Nejprve vyzkoušíme koeficienty pro exponenciální funkci $w_{ex\_date}(k)$ a stanovíme parametr k . Použity byly koeficienty $K = \{0, 0.001, 0.01, 0.1\}$, výsledky jsou v tabulce 6.9. Krajní koeficienty zřejmě vykazují nejhorší výsledky a proto se dále podíváme do intervalu $(0.001, 0.01)$.

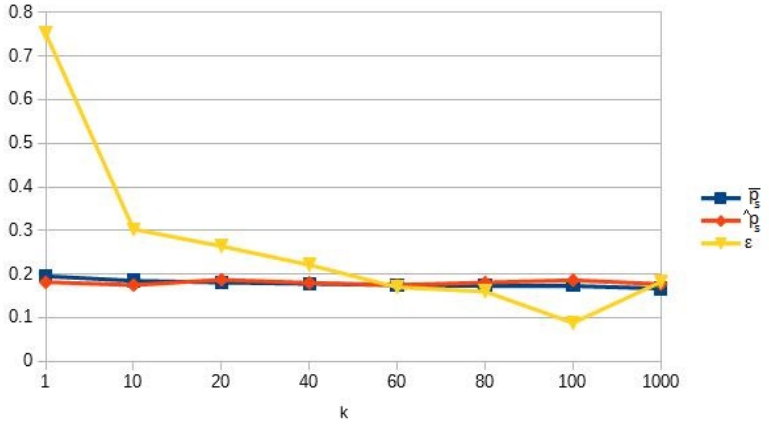
Opět byly stanoveny koeficienty $K = \{0.002, 0.004, 0.006, 0.008\}$ a vyhodnoceny, viz. tabulka 6.10. Nakonec jako ideální prohlásíme koeficient $k = 0.006$, jeho medián $\hat{p_s}$ je vůbec nejvyšší, který jsme v obou tabulkách měli, a odchylka je pro nás stále ještě vyhovující. Pokud bychom chtěli naopak co nejmenší odchylku, vhodný kandidát by byl koeficient $k = 0.001$.

Pro další váhovou funkci $w_{date}(k)$ opět stejným způsobem stanovíme vhodné k. Tentokrát jsou jednotlivé koeficienty, včetně výsledků, zobrazeny v grafu 6.2. Vybráno bylo $k = 20$, které mělo nejvyšší medián p_s, s výjimkou $k = 1$. Stanovení

k	$\overline{p_s}$	$\hat{p}_s$	ϵ	1.	2.	3.	4.
0.002	0.1687	0.1778	17.1%	5	9	11	16
0.004	0.1701	0.1784	31.2%	7	7	14	13
0.006	0.1776	0.181	34.3%	11	17	6	7
0.008	0.1808	0.1857	34.8%	18	8	10	5

Tabulka 6.10: Detailnější koeficienty data pro cyklistiku

těchto koeficientů je poměrně složité, zejména s ohledem na fakt, že simulovány byly jen 2 sezóny, za které se jelo na Tour de France 41 etap. Zároveň jsme data vyhodnotili jen na základě prvních 10 umístění.



Obrázek 6.2: Vyhodnocení koeficientů času pro Tour de France

6.3.7 Profil závodu

Výškovému profilu závodu byla věnována při tvorbě modelu velká pozornost. Opět se podíváme pouze na Tour de France. Tento závod obsahuje velmi rozdílné etapy, a proto bude vhodný k našemu pozorování.

Nejprve se podíváme na funkci $w_{profile}(k)$ (5.13) a stanovme koeficient k. Ve 2 krocích byly zvoleny koeficienty a ty jsou společně s výsledky zaneseny do tabulky 6.11. V prvním řádku je pro srovnání uveden koeficient $k = 0$, který znamená váhu 1 pro každý profil trasy, jedná se tedy o základní bezparametrový model. Můžeme si všimnout, že odchylka ϵ s rostoucím k podle očekávání vždy neroste. Pravděpodobně

k	$\overline{p_s}$	$\hat{p}_s$	ϵ
0	0.1649	0.1754	10%
1	0.2077	0.2226	21.77%
2	0.2365	0.2396	19.83%
3	0.2541	0.2467	23.9%
4	0.2615	0.2461	23.22%
5	0.2701	0.2515	25.3%
6	0.2734	0.2563	27.84%
7	0.2796	0.2472	34.8%
10	0.2872	0.2612	40.76%
20	0.2985	0.2644	89.15%
50	0.3045	0.2251	161.5%

Tabulka 6.11: Volba koeficientů pro parametr profilové náročnosti závodu

k tomuto jevu dochází s ohledem na nízký počet simulovaných závodů. Vhodným koeficientem se zdá být $k = 6$.

Dále jsme definovali funkci $w_{power}(k)$ v (5.18). Opět byla sada koeficientů otestována na každé etapě Tour de France za roky 2015, 2016. Velmi zajímavý se zdá již koeficient $k = 1$, který přináší velké zlepšení oproti základnímu modelu bez parametrů a zároveň mají stejnou odchylku ϵ. Ve srovnání s koeficienty podle předchozí funkce, které jsou uvedeny v tabulce 6.11, aktuálně zkoumaná funkce vykazuje vyšší medián $\hat{p}_s$. A při podobných výsledcích se zdá být odchylka menší. Vhodným kandidátem je koeficient $k = 16$.

Pro pochopení, proč se výsledky oproti základnímu modelu takto zlepšily se podíváme na predikci závodu Tour de France, který se jel 21.7.2016. Etapa byla kopcovitá a vítězem se nakonec stal Romain Bardet před Rodriguezem a Valverdem. V první desítce se umístil i Froome, či Quintana. Podívejme se, jak si vedl náš základní bezparametrový model. Ten v sestupném pořadí predikoval na vítězství P. Sagana, dále Greipela, Kittela, Frooma a Degengolba. Mezi 5 největších favoritů tedy zařadil pouze jediného vrchaře (Chrise Frooma), který měl reálnou naději na vítězství. O ostatních 4 závodnících bylo přitom odborníkům dopředu jasné, že zvítězit prakticky nemohou.

k	$\overline{p_s}$	$\hat{p}_s$	ϵ
0	0.1649	0.1754	10%
2	0.2034	0.2239	10.04%
4	0.2171	0.2364	11.67%
6	0.2271	0.2419	16.7%
8	0.2408	0.2576	19.92%
10	0.261	0.2618	21.7%
12	0.2646	0.262	20.63%
14	0.2646	0.2698	23.04%
16	0.271	0.2675	26.54%
20	0.2823	0.2755	33.25%
50	0.304	0.2896	56.7%
100	0.3165	0.3037	101.7%

Tabulka 6.12: Volba koeficientů pro parametr založený na výkonu a profilové náročnosti závodu

Nyní se podívejme na model s parametrem $w_{power}(k = 16)$. V sestupném pořadí favorizuje Quintanu, Nibaliho, Frooma, Majku, Dumoulina. Model tedy velmi dobře vybral hlavní favority závodu a nerozkládal pravděpodobnost na vítězství mezi závodníky, kteří nemají reálnou naději na úspěch. Při pohledu na některé další etapy se povedlo nalézt i množství závodů, při kterých predikce není ideální jako v tomto případě. Jedná se zejména o středně kopcovité etapy, kdy model ponechává šance rozdělené mezi vrchaře, sprintery i výbušné jezdce.

6.3.8 Konečná volba koeficientů

Ukázali jsme si, jak stanovit koeficienty v případě použití jediného parametru závodu. Nejjednodušší možností se zdá vzít navržené koeficienty pro každý parametr a vytvořit tak výsledný model. Tato úvaha je pravděpodobně správná v případě, že máme k dispozici dostatečný počet předcházejících výsledků. K tomu však prakticky nikdy nemůže dojít, v cyklistice je odlišných typů závodů mnoho a počet závodů pro tento způsob nemusí být dostatečný. Samozřejmě není možné přesně říct, jaký počet závodů je již dostatečný ale ideálně by se jistě měl blížit nekonečnu. Příliš vysoký

počet použitých parametrů, může vést k nepříznivým výsledkům.

Podívejme se zpět na jednotlivé parametry, největší efekt přinesl parametr profilové náročnosti trasy, odvozený od výkonu závodníků, dále parametr data závodu a rozlišení jízdy proti chronometru a hromadného závodu pomocí váhové funkce $w_{start}(0)$. Naopak parametr délky a úrovně se nezdá být příliš užitečný.

Při samostatném stanovení koeficientů jsme doporučili některé hodnoty, nyní je použijeme zároveň. Testovaný model bude opět poziční a zaměříme se na predikci prvních 10 závodníků v cíli etapy. V tabulce 6.13 vidíme výsledky v závislosti na koeficientech, které byly zvoleny pro jednotlivé váhové funkce. V prvním řádku jsou koeficienty, které byly postupně doporučeny při zahrnutí jediného parametru. Odchylka pro takový model je však již velmi vysoká $\epsilon = 100\%$. Se stejnou odchylkou lze na obdobné výsledky dosáhnout i s jediným parametrem $w_{power}(100)$. Stanovit ideální váhové parametry je poměrně složité, lze použít hrubou sílu a testovat různé možnosti, což je výpočetně extrémně náročné. Přesto takový postup povede k nejlepším výsledkům. My využijeme určitou znalost modelů a pokusíme se stanovit, co nelepší možné koeficienty, pro Tour de France. Budeme přitom požadovat, aby odchylka ϵ výrazně nepřesáhla 15

Exponenciální funkce pro datum závodů potlačuje starší výsledky, které však mohou být užitečné. V druhém řádku tabulky vidíme obdobné koeficienty, jen je nahrazena exponenciální funkce lomenou. Výsledky zůstávají stejné, ale odchylka se zmenšila na 74.5%. Dále jsme zvolili koeficient času a profilu tak, abychom se přiblížili 15%. Zároveň jsme vyzkoušeli vynechat jednotlivé parametry, abychom zjistili, jestli mají skutečný přínos pro výsledný model. Za vhodnou kombinaci tedy můžeme označit $w_j = w_{start}(0)w_{date}(100)w_{power}(4)$. Tento model nepřekoná, při zachování ϵ, žádný jednoparametrový, který jsme dříve vyzkoušeli.

w_{start}	w_{ex_date}	w_{date}	w_{power}	$\overline{p_s}$	$\hat{p}_s$	ϵ
0	0.008	-	16	0.3026	0.2896	100%
0	-	20	16	0.303	0.2893	74.5%
0	-	100	8	0.2865	0.2811	37.37%
0	-	100	4	0.2645	0.2818	15.5%
0	-	-	4	0.2572	0.262	16.5%
-	-	100	4	0.2332	0.2487	16.79%
0	-	300	4	0.26	0.2552	13.2%

Tabulka 6.13: Stanovení parametrů pro Tour de France

7 Implementace modelů v dalších sportovních odvětvích

Navržený model i jeho parametry jsou obecné a může být nasazen i na ostatních sportovních odvětvích. Pokud parametr ve vybraném sportu neexistuje, je váha nastavena na 1, a neovlivňuje tak žádným způsobem parametry ostatní. Modely i statistiky jsou programovány podle abstraktního návrhu, který byl již dříve popsán. Vytvářena je tak již jen konečná implementace.

7.1 Formule 1

Výsledky závodů F1 byly staženy ve formě databáze MSSQL, ubyla tak potřeba psát pomocný program pro stažení dat. Jediný zásah vyžadovala transformace dat do databáze H2, která je podobná spíše mysql. V případě, že není nutno přepisovat triggery, či procedury jsou databáze relativně podobné a migrace na jinou databázi nepředstavuje problém.

Formulové závody mají jediný parametr, který byl představen a je jim datum konání. Závody F1 jsou velmi specifické a není zde potřeba testovat základní typy modelů, zřejmě nejlépe opět bude vycházet model poziční. Nejprve se podívejme na bezparametrický model, který by tedy měl váhu $w_j = 1$. Otestujeme model na sezónách 2010-2015. Celková průměrná úspěšnost $\overline{p_s} = 1.402$, což nám moc nenaznačí, každý sport se totiž vyznačuje vlastní mírou obtížnosti predikce. Zajímavý údaj poskytuje především odchylka $\epsilon = 19.75\%$. Připomeňme, že u silniční cyklistiky $\epsilon = 74.65\%$. Důvodem zřejmě bude, že u závodů F1 máme dostatek dat, vzhledem k možným pozicím. Dále je F1 poměrně specifická tím, že startovní pole zůstává po celou sezónu stejně početné a zároveň se závodu účastní stejní závodníci.

k	$\overline{p_s}$	$\hat{p}_s$	$\overline{p_r}$	$\hat{p}_r$	ϵ
0	1.4023	1.4001	0.3168	0.3119	19.75%
0.001	1.5503	1.5523	0.3723	0.3582	7.96%
0.002	1.6422	1.6335	0.4145	0.393	7%
0.004	1.7417	1.7306	0.4646	0.423	6.7%
0.006	1.7999	1.7859	0.4937	0.4506	10.5%
0.008	1.8417	1.8192	0.5161	0.4628	12.2%
0.01	1.8737	1.84	0.5303	0.4703	16.33%
0.012	1.895	1.862	0.538	0.4798	20.7%
0.014	1.912	1.886	0.5498	0.488	24.7%
0.016	1.922	1.892	0.5588	0.4789	29.4%
0.018	1.933	1.891	0.5643	0.4916	33.6%
0.02	1.9404	1.8991	0.5693	0.5031	39.3%
0.1	1.9952	1.8578	0.634	0.538	131.6%

Tabulka 7.1: Stanovení časového koeficientu pro F1

Nyní navrhneme pro funkci váhy $w_{ex\_date}(k)$ hledaný koeficient. V tabulce 7.1 jsou zaneseny výsledky testů s mnoha různými koeficienty k. Se zvyšujícím se koeficientem k se pravidelně zvyšují i metriky $\overline{p_s}, \hat{p}_s, \overline{p_r}$ i $\hat{p}_r$. Zajímavá je především odchylka ϵ, která se od základního bezparametrového modelu ($\epsilon = 19.75\%$) dokonce výrazně sníží až na 6.7% při koeficientu $k = 0.004$. Forma závodníků, ale zejména aktuální výkonnost jejich vozů výrazně ovlivňuje potenciální výsledky. Navíc máme dostatek relevantních dat k tomu, aby model dobře fungoval. Proto zde vidíme, že zvýšený koeficient k dokáže snížit odchylku ϵ. Dále se stoupajícím koeficientem k roste úspěšnost modelu, ale zároveň i odchylka ϵ a ideální koeficient tedy bude záviset na požadavcích tvůrce modelu.

Redukce pravděpodobností je výpočetně náročná operace. V případě závodů F1 stačí méně simulací (1000), než u cyklistiky, jelikož počet různých pozic je výrazně nižší a zejména malý počet umístění zrychluje řazení v metodě Monte Carlo. V tabulce 7.1 vidíme výsledky základní redukční metody, jejíž koeficienty jsou počítány na základě posledních 40 závodů. Je nutné podotknout, že se po redukci žádná pravděpodobnost nevyskytuje v intervalu $(0, 0.001)$. Odchylka zůstává pro všechny

k	$\overline{p_s}$	$\hat{p}_s$	$\overline{p_r}$	$\hat{p}_r$	ϵ
0	1.3079	1.3033	0.2706	0.2706	7%
0.001	1.4414	1.4742	0.3153	0.3265	6.08%
0.002	1.5632	1.5625	0.3682	0.3563	11.7%
0.004	1.6246	1.6197	0.405	0.3829	9.8%
0.006	1.6596	1.64	0.4285	0.3994	14%
0.008	1.6758	1.6276	0.4454	0.4	14.3%
0.01	1.6657	1.6337	0.4356	0.3773	10.2%
0.012	1.6466	1.586	0.4311	0.3905	13%
0.014	1.6297	1.5703	0.426	0.382	10.7%
0.018	1.6107	1.5521	0.4167	0.3704	12.1%
0.1	1.3641	1.347	0.2995	0.2677	11.2%

Tabulka 7.2: Stanovení časového koeficientu pro F1 pomocí redukce pravděpodobností

časové koeficienty k velmi podobná, jelikož se pravděpodobnost automaticky redukuje. Pokud je k příliš vysoké, začne úspěšnost celého modelu klesat, jelikož se pravděpodobnosti příliš redukují.

Byla provedena i poziční redukce, koeficienty redukce se opět počítají na základě 40 posledních závodů a pozičním koeficientu $k_1 = 5$. Výsledky jsou uvedeny v tabulce 7.1. Oproti běžné redukci se snížila odchylka ϵ a zároveň se zlepšily i všechny sledované metriky úspěšnosti modelu.

Pro závody F1 nejlepších výsledků dosahuje běžná metoda predikce, objevují se při ní však nulové pravděpodobnosti. Pokud tomuto problému chceme předejít volíme redukční metodu založenou na pozicích, která v každém ohledu předčila základní redukci.

7.2 Běžecké lyžování

Data z běžeckého lyžování i biatlonu opět nejsou volně dostupné a bylo třeba napsat obdobně složitý program, na jejich stahování, jako v případě cyklistiky. Výsledky jsou však centrálně k dispozici na oficiálním serveru mezinárodní běžecké unie. De-

k	$\overline{p_s}$	$\hat{p}_s$	$\overline{p_r}$	$\hat{p}_r$	ϵ
0.004	1.6739	1.6323	0.4264	0.3823	7.35%
0.006	1.7062	1.6691	0.4466	0.3958	9.17%
0.008	1.7074	1.667	0.449	0.4026	9.48%
0.01	1.708	1.6584	0.4541	0.4166	9.12%
0.012	1.7096	1.6759	0.4522	0.4106	9.68%
0.014	1.6957	1.6276	0.4485	0.4076	6.33%
0.016	1.687	1.6319	0.4466	0.4122	7.44%
0.1	1.465	1.4282	0.3521	0.3051	9.53%

Tabulka 7.3: Stanovení časového koeficientu pomocí poziční redukce pravděpodobnosti pro F1

tailní výsledky jsou však ve formě souborů pdf a dokonce je nutné, aby je uživatel sám stáhnul. Po jejich stažení vloží jejich textovou kopii do uživatelského rozhraní, které bylo vytvořeno. PDF soubory mají poměrně složitý formát a navíc není jednotný, proto v případě chyby je uživatel informován a musí provést změnu, jinak data nemohou být úspěšně uloženy do databáze. Tato práce však prozatím zůstává jen velmi kvalifikovanému uživateli, který je obeznámen se strukturou PDF souboru a může jej správně modifikovat.

Pro běžecké lyžování jsou prozatím k dispozici jen závody nejvyšší kategorie, žen i mužů, které se konaly mezi 29.11.2014 a 1.7.2017. Kompletní jsou tedy 2 sezóny, sezóna 2016-2017 je zpracována jen částečně. S ohledem na nedostatek dat se ukazuje, že metoda redukce pravděpodobností nepřináší dobré výsledky. Testovat budeme závody od sezóny 2015/2016.

Stejně jako v cyklistice, se podíváme na základní typy modelů. Vynechán je pouze model založený na celkovém čase, který se naprosto zřejmě nehodí. V tabulce 7.4 vidíme výsledky všech 4 typů modelů. Zobrazené metriky jsou vztaženy jen k prvním 10 umístěním, které nás zajímají především. Relativní časový model se oproti cyklistice výrazně zlepšil a vykazuje již velmi zajímavé výsledky, drobné zlepšení pak přináší vylepšený model reflektující časový odstup vzhledem k délce závodu. Nejlepší výsledky však stále prokazuje model poziční, který u běžeckého lyžování výrazně poráží relativně poziční.

model	$\overline{p_s}$	$\hat{p}_s$	ϵ
relativní časový	0.4871	0.4758	42.7%
relativní / délka	0.503	0.5001	25.3%
relativní poziční	0.608	0.5602	37.96%
poziční	0.6887	0.6796	0.204

Tabulka 7.4: Výsledky základních typů modelů na běžeckém lyžování

Bezparametrový poziční model ($w_j = 1$) vykazuje úspěšnost $\overline{p_s} = 2.1226$ při odchylce $\epsilon = 53.2\%$. Pokud bereme v potaz jen prvních 10 umístění, pak $\overline{p_s} = 0.6916$ a $\epsilon = 17.3\%$. Ve srovnání s cyklistikou dosahují obě uvedené metriky lepších výsledků, což je dáno motivací závodníků závodit o horší umístění a pravděpodobně i větší rozdíly ve startovním poli. Při oddělení závodů podle způsobu startu (hromadný, individuální) bude mít model váhu j-tého závodu $w_j = w_{start}(0)$ získáme výsledky $\overline{p_s} = 2.223$, $\epsilon = 105.9\%$. Za výrazné zvýšení odchylky může nedostatek dat.

U běžeckého lyžování jsou k dispozici informace o celkovém převýšení, lze tedy využít váhovou funkci (5.13). Při testování se však ukazuje, že se výsledky příliš nezlepšují, ale výrazně se zvyšuje odchylka ϵ. Výsledky, které máme v databázi jsou ze světového poháru, kde je profilová náročnost často velmi podobná. Zároveň se lze domnívat, že profil trasy nehraje tak podstatnou roli jako v cyklistice.

Délka závodu je naopak v běžeckém lyžování velmi podstatná, mnozí závodníci se specializují na určité vzdálenosti. Nadefinovali jsme dvě funkce (5.11), (5.12). Ukážeme si výsledky první z nich, která byla jednoznačně úspěšnější. V tabulce 7.2 jsou uvedeny výsledky testu pro koeficient k funkce $w_{length}(k)$ s filtrem nastaveným na prvních 10 závodníků. Při zvyšování koeficientu k se metrika $\overline{p_s}$ pravidelně zvyšuje, až do $k = 100$. Medián $\hat{p}_s$ naopak dosahuje maximální hodnoty u $k = 20$, zároveň se pro všechny koeficienty zvyšuje odchylka ϵ. V případě použití redukce pravděpodobností se neustále zmenšuje $\overline{p_s}$ a koeficient pro ní tedy nemá žádný význam. Důvodem zřejmě bude nedostatečný počet dat.

Posledním zajímavým parametrem je datum závodu. Opět byly představeny dvě funkce, výsledky testů jsou zaneseny do tabulek 7.6, 7.7. Dle předpokladů opět roste úspěšnost modelu $\overline{p_s}$ v závislosti na kladení vyššího významu nedávným závodům. Zároveň se však zvyšuje i odchylka ϵ. Obě funkce vykazují velmi podobné výsledky

k	$\overline{p_s}$	$\hat{p}_s$	ϵ
0	0.6894	0.6917	20.2%
1	0.7348	0.7204	23.3%
2	0.7445	0.7189	27.4%
5	0.7892	0.7452	33.74%
10	0.8316	0.7676	71.96%
20	0.8653	0.8019	85.1%
50	0.8813	0.7978	88.32%
100	0.8814	0.7889	102%
200	0.8397	0.7535	123%

Tabulka 7.5: Vyhodnocení vzdálenostního koeficientu pro běžecké lyžování

k	$\overline{p_s}$	$\hat{p}_s$	ϵ
0.001	0.7074	0.7056	19.7%
0.002	0.7177	0.7045	18.2%
0.005	0.74	0.7031	25.1%
0.01	0.7589	0.7486	43%
0.02	0.7639	0.7639	55.7%
0.05	0.7762	0.7557	63.5%

Tabulka 7.6: Časový koeficient pro exponenciální funkci v běžeckém lyžování

a nelze jednoznačně určit, která je vhodnější.

Základní model $w_j = 1$ pro běžecké lyžování má odchylku 20.2%. Pomocí samostatných parametrů času, či délky závodu jsme byli schopni při obdobné odchylce získat lepší výsledky. Na závěr stanovíme váhovou funkci v závislosti na všech 3 parametrech zároveň tak, aby výrazně nepřesáhla 30%. Již z cyklistiky víme, že je třeba kombinovat parametry takové, které měly velmi nízké odchylky ϵ.

Vhodné se zdají být parametry pro délku závodu $(w_{length}(k))$ $k \in \{1, 2\}$. Pro časovou exponenciální funkci $(w_{ex\_date}(k))$ $k \in \{0.002, 0.005\}$. Lomená časová funkce $(w_{date}(k))$ $k \in \{100, 50\}$. Máme 2 délkové parametry a 4 časové, celkově 8 různých kombinací. Použijeme metodu hrubé síly a vyzkoušíme všechny možné kombinace. V tabulce 7.8 jsou zobrazeny výsledky všech 8 kombinací. Nejlépe vychází volba váhy

k	$\overline{p_s}$	$\hat{p_s}$	ϵ
200	0.7226	0.7124	20.4%
100	0.7303	0.702	23.2%
50	0.7421	0.706	28.9%
20	0.7518	0.7275	38.3%
10	0.7631	0.735	41.8%
5	0.7644	0.7267	51.5%
1	0.7664	0.7333	72%

Tabulka 7.7: Časový koeficient pro lomenou funkci v běžeckém lyžování

$w_{length}(k)$	w_{ex_date}	w_{date}	$\overline{p_s}$	$\hat{p_s}$	ϵ
1	0.002	-	0.76034	0.7213	26.2%
1	0.005	-	0.78034	0.7611	33%
1	-	50	0.7794	0.7524	34.77%
1	-	100	0.7723	0.7549	26.6%
2	0.002	-	0.7652	0.7348	38.4%
2	0.005	-	0.7885	0.779	42%
2	-	50	0.7886	0.7788	44.4%
2	-	100	0.781	0.7745	42%

Tabulka 7.8: Konečný návrh parametrů pro běžecké lyžování

$w_j = w_{length}(1)w_{date}(100)$. Pokud mírně překročíme nastavenou hranici 30%, potom vychází nejlépe $w_j = w_{length}(1)w_{ex\_date}(0.005)$.

8 Závěr

Základním předpokladem nasazení matematických modelů pro predikci sportovních výsledků je získání dostatečného množství dat pro vytvoření statistik. Primárním sportem, na který se práce zaměřovala, byla silniční cyklistika. Nemožnost získání výsledků v předpřipravené formě znamenala nutnost vytvořit program, který výsledky nejprve získal z webového serveru ve formě běžných HTML stránek. Následně je pomocí parseru a regulárních výrazů zpracoval a uložil do lokální databáze. Server, ze kterého jsou data získávána, změnil v průběhu psaní práce formát výstupu HTML stránek a program pro stahování dat musel být ze značné části přepsán. V rámci práce byly plně implementovány i statistiky pro biatlon, běžecké lyžování a formulí 1. Jedině v případě F1 se podařilo získat data ve formě databáze mssql, u ostatních sportů byl opět vytvořen program na stahování dat, který čerpal z oficiálních stránek organizací. Biatlonové statistiky byly již připraveny v rámci předešlé bakalářské práce, ale změna webu mezinárodní biatlonové federace opět znamenala nutnost přepsání původního programu. Vytváření programů, které získávají data pomocí parserování webových stránek, vede nutně při každé změně na straně serveru k zásahu do napsaného kódu. U vybraných sportů však neexistovala jiná přijatelná alternativa.

Ze získaných výsledků bylo nutné vytvořit statistiky, které by mohl využívat výsledný model. Byl tedy vytvořen velmi obecný objektový návrh statistik, využívající ve velké míře generických datových typů. Navržený způsob vytváří základní jádro statistik pro všechny individuální sporty založené na časové klasifikaci a případná specifika jednotlivých sportů jsou implementována pomocí dědičnosti a rozhraní. V programové implementaci model následně pracuje s vybranými statistikami a predikuje požadovaný závod po předložení startovní listiny. I zde je dbáno na vysokou míru abstrakce. Veškeré testování modelů i zobrazení výsledků je společné pro

všechna sportovní odvětví, která jsou schopna implementovat vytvořené rozhraní.

Matematické modely, které predikují výsledky závodů, jsou založeny na vytvoření nezávislých rozdělení pro každého závodníka na základě četnostních funkcí jejich předchozích výsledků. Z nezávislých rozdělení je následně vytvořena výsledná predikce. Výpočet je velmi náročný, a tak byla nasazena metoda Monte-Carlo. Jednotlivým výsledkům z předcházejících závodů jsou přikládány váhy tak, aby predikce získávala co nejlepší možné výsledky. Váha se získává na základě data konání závodů, ale i délky, způsobu startu nebo úrovně vybraného závodu. Zmíněné parametry jsou obecné pro všechna sportovní odvětví, která zapadají do diplomové práce. Speciální parametr byl vytvořen pro cyklistiku, kde je známo, že výškový profil závodu výrazně ovlivňuje konečné výsledky. Funkce, která stanovuje váhu v závislosti na výškovém profilu, vychází z fyzikálních zákonů, konkrétně výpočtu výkonu závodníků, a snaží se na jeho základě vypočíst, zda trasa více vyhovuje silovým závodníkům, nebo jezdcům specializujícím se na těžká a dlouhá stoupání. Výškový profil se získává z obrázků pomocí detekce hran, ale i pomocí dalších speciálně navržených funkcí.

Modely v některých případech predikují pravděpodobnosti, které při jejich nasčítání neodpovídají reálným výsledkům. Problematickým místem je především častá predikce nulových pravděpodobností, která plyne z nedostatku dat, špatně stanovených parametrů nebo malého počtu simulací metody Monte Carlo. Byla navržena metoda redukce pravděpodobností a následné přegenerování pravděpodobností pomocí metody Monte Carlo. Tím se problém vyřešil, ovšem za cenu rozmělnění predikce.

Stanovení optimálních parametrů pro modely je poměrně složité, zejména v případě, kdy použijeme více parametrů zároveň. Úspěšnost volby parametrů je testována experimentálně za použití navržených metod pro vyhodnocení modelů. U většiny sportovních odvětví je jednodušší predikovat přední umístění a zároveň je tato predikce i zajímavější s ohledem na možné nasazení modelů u sázkových kanceláří. Proto při stanovení parametrů, byla většinou použita jen úspěšnost predikce předních umístění. V cyklistice se velmi dobře ukázal parametr profilové náročnosti trasy a oddělení časovky od hromadných závodů. Zlepšení výsledků zaručí i parametr zohledňující aktuální výkonnost závodníků podle data konání závodů. Ukázána je zejména vhodná volba parametrů pro Tour de France. Stanovení vhodných para-

metrů je dále provedeno i pro běžecké lyžování a závody F1.

Důležitou součást práce je i grafické rozhraní. Stanovení výškového profilu trasy není možné plně automatizovat, a tak některé kroky musí vykonat uživatel, k čemuž mu slouží právě grafické rozhraní. V rámci práce bylo zpracováno více jak 500 výškových profilů. Dále lze zobrazit i výsledky predikce. Uživatel si vybere sportovní odvětví, závod, který chce predikovat, počet simulací pomocí metody Monte-Carlo. Parametry modelu a jeho typ si rovněž může libovolně zvolit. Zároveň si může libovolně poskládat startovní listinu závodu. Pro cyklistiku jsou dostupné i některé startovní listiny nadcházejících závodů. Závodníky je také možné třídit na základě jejich příslušnosti k národu, či týmu v případě cyklistiky. Po odsimulování závodu si uživatel může prohlédnout výsledky ve formě tabulky, kde jsou závodníci seřazeni podle pravděpodobností umístění. Po kliknutí na vybraného závodníka se graficky zobrazí jeho distribuční funkce v závislosti na umístění a také se vypíší pravděpodobnosti, že daný závodník bude v cíli dříve než konkrétní jeho soupeř. Grafické rozhraní dovoluje i testovat navržené modely a zároveň je srovnávat podle několika základních metrik. Testování může být výpočetně velmi náročné a je vhodné testovat pouze několik modelů zároveň. Jelikož je každý model testován v jiném vlákně, pro víceprocesorový počítač je vhodné testovat více jak jeden model. Problematické místo představuje především možný nedostatek operační paměti.

Diplomová práce je poměrně rozsáhlá, přesto existuje mnoho možností, jak ji dále rozvíjet. Nejdůležitější parametr pro predikci výsledků cyklistických závodů byl naprosto zřejmě založen na výškovém profilu trasy. Zpracování profilu z obrázku byla věnována celá kapitola, přesto se prozatím nelze úplně spolehnout na sklon vozovky ve vybraných úsecích. Hranová detekce i kombinovaná sloupcová detekce mají své problémy. Sloupcová často není tak dobře vyhlazená jako hranová. Hranová naopak má problémy s určením začátku a konce trasy, jelikož se nenachází na začátku a konci obrázku. U hranové detekce se navíc vždy nedaří vybrat skutečnou hranu profilu. Řešení může skýtat další kombinace metod, tentokrát kombinované sloupcové s hranovou detekcí. Kombinovaná metoda by nejprve přibližně určila výškový profil, velmi dobře totiž stanovuje začátek a konec profilu. Zároveň by přibližně určila body, kterými má procházet profil a tato vodítka by následně byla použita při detekci pomocí hran.

Zlepšení výškového profilu by rovněž umožnilo zavést další způsob určení vah v závislosti na tomto profilu. Nejprve bylo spočteno prosté převýšení na celé trase vzhledem k jeho délce, dále byl postup vylepšen a počítán čas závodníků podle jejich somatotypů a jejich předpokladů k určitým závodům. Při hromadných závodech však největší roli hraje profil na konci závodu, kde se obvykle o výsledcích rozhoduje. Nejjednodušším způsobem by bylo vzít funkce, které již byly vytvořeny, a upravit je tak, aby přikládaly větší váhu konci závodu. Složitější způsob by pak mohl vycházet ze snížení odporu vzduchu v závětří a na jeho základě, se pokusit určit co se stane, když v jednotlivých fázích závodu zaútočí osamocený závodník. Respektive spočítat o kolik větší výkonnost musí mít závodník, aby byl schopen v daném okamžiku vyhrát osamocen etapu. Pokud na začátku etapy je velké stoupání, ale následuje 100 km po rovině, peloton je obvykle na uprchlíky schopen stáhnout kolem deseti minut. V takovém případě se nejedná o kopec, kde by měl útočit vrchař.

Experimentální testování parametrů v případě, že je parametrů mnoho, je poměrně složité. V práci je nastíněna možnost testování pomocí hrubé síly. Bylo by vhodné vytvořit algoritmus, který by sám testoval základní parametry a případně se je snažil i korigovat. Samotný test je již nyní velice náročný, a proto se nabízí v dalším kroku takový test provést na clusteru.

9 Příloha A : Obsah přiloženého CD

Uživatelská příručka k vytvořenému grafickému rozhraní

Diplomová práce ve formátu PDF

Zdrojový kód aplikace (projekt pro vývojové prostředí NetBeans IDE 8.2)

Literatura

[1] MISHRA, R. K. a Simaranjeet KAUR. Mathematical Modeling Approach to Predict Athletic Time, Performance. Universal Journal of Applied Mathematics [online]. 2013, 2013(1(4), 242-246 [cit. 2016-07-03]. DOI: 10.13189/ujam.2013.010406. Dostupné z: http://www.hrpub.org/download/20131201/UJAM6-12601446.pdf

[2] Equipment. Union Cycliste Internationale [online]. [cit. 2016-07-19]. Dostupné z: http://www.uci.ch/inside-uci/rules-and-regulations/equipment-165067/

[3] COTE, Mark. Aerodynamics of Time Trial versus Road Configurations [online]. , 1-11 [cit. 2016-07-19]. Dostupné z: http://a2wt.com/research.pdf

[4] DE JONG, Jenny. On the optimal power distribution for cycling a time trial. Utrecht, 2015. Master's thesis. Utrecht University. Vedoucí práce Dr R. Fokkink.

[5] LUKES, R. A., S. B. CHIN, S. J. HAAKE a Nicholas A.T. BROWN. The understanding and development of cycling aerodynamics. Sports Engineering [online]. 2005, 8(2), 59-74 [cit. 2016-07-21]. DOI: 10.1007/BF02844004. ISSN 1369-7072. Dostupné z: http://link.springer.com/10.1007/BF02844004

[6] BARRY, Nathan, John SHERIDAN, David BURTON a Nicholas A.T. BROWN. The Effect of Spatial Position on the Aerodynamic Interactions between Cyclists. Procedia Engineering [online]. 2014, 72, 774-779 [cit. 2016-07-21]. DOI: 10.1016/j.proeng.2014.06.131. ISSN 18777058. Dostupné z: http://linkinghub.elsevier.com/retrieve/pii/S187770581400647X

[7] Just How Good Are These Guys? Cycling tips [online]. [cit. 2016-10-27]. Dostupné z: http://cyclingtips.com/2009/07/just-how-good-are-these-guys/

[8] ROKYTA, Mirko. Řešení obecné kubické rovnice. [online]. [cit. 2016-10-31]. Dostupné z: https://www.karlin.mff.cuni.cz/ rokyta/vyuka/general/tahaky/kubicka.html

[9] KAMINSKY, Alan. Ixent: Subversion: trunksrcorgjvnetixentmathequationsEquationManager.java Project Kenai [online]. [cit. 2016-10-31]. Dostupné z: https://java.net/projects/ixent/sources/svn/content/trunk/src/org/jvnet/ixent/math/equati

[10] MAHER, M.J. Modelling association football scores [online]. 1981 [cit. 2016-11-07]. Dostupné z: http://www.90minut.pl/misc/maher.pdf

[11] Modelling Association Football Score and Inefficiencies in the Football Betting Market [online]. DIXON, Mark J. a Stuart G. COLES. [cit. 2016-11-07]. Dostupné z: http://www.math.ku.dk/ rolf/teaching/thesis/DixonColes.pdf

[12] SLAVÍK, Václav. On-line systém pro modelování a predikci sportovních výsledků. Liberec, 2013. Bakalářská práce. Technická univerzita v Liberci. Vedoucí práce Petr Volf.

[13] Přehled výsledků provozování loterií a jiných podobných her za rok 2015. Ministerstvo financí České republiky [online]. [cit. 2016-11-15]. Dostupné z: http://www.mfcr.cz/cs/soukromy-sektor/loterie-a-sazkove-hry/vysledky-z-provozovani-loterii/2015/hodnoceni-vysledku-provozovani-loterii-25162

[14] MORONEY, M. (1951), Factsfrom figures, London, Pelican.

[15] Power estimates - Alpe d'Huez (1997-2013) [online]. [cit. 2017-02-05]. Dostupné z: http://rodman1r2.tumblr.com/post/57549681394/power-estimates-alpe-dhuez

[16] Cancellara's data for Olympic gold medal TT is truly awesome [online]. 2016 [cit. 2017-02-05]. Dostupné z: http://www.stickybottle.com/coaching/cancellaras-data-for-olympic-gold-medal-tt-is-truly-awesome/

[17] Useful Color Equations [online]. [cit. 2017-02-28]. Dostupné z: http://www.brucelindbloom.com/index.html?Equations.html

[18] Canny Edge Detector. Open CV Documentation [online]. [cit. 2017-03-05]. Dostupné z: http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.

[19] Model-View-Controller. Guides and Sample Code [online]. [cit. 2017-03-06]. Dostupné z: https://developer.apple.com/library/content/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html

[20] Mastering FXML [online]. [cit. 2017-03-06]. Dostupné z: http://docs.oracle.com/javafx/2/fxml_get_started/jfxpub-fxml_get_started.htm

[21] Factory Design Pattern in Java. JournalDev - Java, Java EE, Android, Web Development Tutorials [online]. [cit. 2017-05-06]. Dostupné z: http://www.journaldev.com/1392/factory-design-pattern-in-java

[22] Observer Design Pattern in Java [online]. [cit. 2017-05-06]. Dostupné z: https://sourcemaking.com/design_patterns/observer