Univerzita Palackého v Olomouci
Přírodovědecká fakulta
Katedra optiky

# BAKALÁŘSKÁ PRÁCE

## Využití Bayesovy statistiky v predikci výroby solární elektrárny

| | |
|---|---|
| Autor: | Vladimír Stuchlík |
| Studijní program: | Fyzika |
| Studijní obor: | Obecná fyzika a matematická fyzika |
| Forma studia: | Prezenční |
| Vedoucí práce: | RNDr. Tomáš Fürst, Ph.D. |
| Termín odevzdání práce | Květen 2022 |

Prohlašuji, že jsem předloženou diplomovou práci vypracoval samostatně pod vedením RNDr. Tomáše Fürsta, Ph.D. a že jsem použil zdroje, které cituji a uvádím v seznamu použité literatury.

V Olomouci …….  ...…….....

## Poděkování

Velmi rád bych poděkoval Jakubu Dostálovi za poskytnutí dat, která jsem v této práci použil a také za průběžné konzultace týkající se mých modelů strojového učení. Také bych rád poděkoval vedoucímu bakalářské práce Tomáši Fürstovi, se kterým jsem konzultoval obsah této bakalářské práce.

# Bibliografická identifikace

| | |
|---|---|
| Jméno a příjmení autora | Vladimír Stuchlík |
| Název práce | Využití Bayesovy statistiky v predikci výroby solární elektrárny |
| Typ práce | Bakalářská |
| Pracoviště | Katedra optiky |
| Vedoucí práce | RNDr. Tomáš Fürst, Ph.D. |
| Rok obhajoby práce | 2022 |
| Abstrakt | Cílem práce je na základě historických dat výroby solární elektrárny, dat počasí v daných historických dnech a numerických meteorologických modelů predikovat výrobu elektřiny pro další den a zjistit chybu predikce. |
| Klíčová slova | Lineární regrese, neuronové sítě, bayesovská inference |
| Počet stran | 69 |
| Počet příloh | 1 |
| Jazyk | český |

# Bibliographical identification

| | |
|---|---|
| Author's first name and surname | Vladimír Stuchlík |
| Title | Use of Bayesian statistics in the prediction of solar power plant production |
| Type of thesis | Bachelor's |
| Department | Department of Optics |
| Supervisor | RNDr. Tomáš Fürst, Ph.D. |
| The year of presentation | 2022 |
| Abstract | The aim of the Thesis is to predict the output of a photovoltaic (solar) power plant. The prediction is based on historical data, and weather forecast. The output one day ahead is predicted prediction error. |
| Keywords | Linear regression, neural networks, bayesian inference |
| Number of pages | 69 |
| Number of appendices | 1 |
| Language | czech |

# Obsah

# Úvod

V této práci se zabývám předpovědí hodinové výroby solární elektrárny (která je určena v kWh) jak na den dopředu ($D + 1$), tak na dvě hodiny dopředu ($H + 2$). Výrobu předpovídám pro každý hodinový časový interval v rozmezí 4–19. hodiny každého dne. V rámci $D + 1$ to znamená, že ze znalosti předpovědi počasí pro každý hodinový časový interval dalšího dne dovedu s určitou chybou předpovědět, kolik elektřiny daná solární elektrárna v daný časový interval vyrobí. Uvedu příklad: dnes je 11.11.2021, chci určit zítřejší celkovou výrobu elektřiny pro hodinový interval v rozmezí 10–11. hodiny, podívám se tedy, jaká jsou predikovaná data počasí pro interval 10–11. hodiny dalšího dne. Tato data můj model dovede zužitkovat a předpovědět danou výrobu s určitou chybou. V rámci $H + 2$ se bavíme o vnitrodenní predikci, to v praxi znamená, že víme, kolik elektřiny vyrobila solární elektrárna v poslední hodinový časový interval a zároveň známe předpověď počasí mezi 2–3. hodinou od okamžiku posledního hodinového časového intervalu. Uvedu příklad: dnes je 11.11.2021, 15:00 hodin a víme, kolik elektřiny vyrobila elektrárna mezi 14–15. hodinou. Známe předpověď počasí pro 16–17. hodinu dnešního dne a tedy dovedeme předpovědět výrobu elektřiny mezi 16–17. hodinou dnešního dne. V této práci se bohužel musíme spokojit s historickými hodnotami počasí, s predikovanými hodnotami počasí se zde nepracuje a počasí také nepředpovídám. Proto, abychom mohli dělat předpovědi, musíme samozřejmě znát historická data počasí a hodnoty vyrobené elektřiny k tomuto počasí náležící.

Úlohu jsem řešil prostřednictvím programovacího jazyka Python v prostředí Jupyter notebooku. Využil jsem Python knihovny *Pandas, NumPy, Matplotlib, Pymc3, Seaborn* a *Sklearn*.

Pro nafitování parametrů modelů používám knihovnu *Pymc3,* která parametry všech modelů počítá pomocí bayesovské inference, to znamená, že pro každý jeden koeficient můžeme získat i celé jeho pravděpodobnostní rozdělení, tudíž i nejistoty. Pravděpodobnostní rozdělení lze spočítat i pro předpovězenou výrobu elektřiny.

# Kapitola 1

# Teoretická část

V této kapitole se seznámíme s matematikou, která je použita v praktické části této práce. Jedná se o lineární regresi, dopředné vícevrstvé neuronové sítě a použití Bayesovy věty v určení koeficientů lineární regrese a vah neuronové sítě.

## 1.1 Použité metody strojového učení

V této podkapitole je uvedena základní struktura lineární regrese, neuronové sítě a jejich aktivační funkce. Při vypracování této kapitoly byly využity zdroje [1], [2], [4], [7], [8].

### 1.1.1 Lineární regrese

Lineární regrese je nejzákladnější statistická regresní metoda sloužící k nalezení závislostí mezi proměnnou $y$ a souborem nezávislých proměnných $\{x_1, x_2, x_3 \ldots . x_n, n \geq 1\}$. Cílem lineární regrese je buď otestovat, jak moc výstupní proměnná závisí na nezávislých proměnných, nebo předpovídat hodnoty $y$ pomocí nezávislých vstupních proměnných.

Lineární regresi lze vyjádřit vzorcem

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \varepsilon, \tag{1.1}$$

kde koeficienty $w_0, w_1, w_2, w_3 \ldots . w_n$, jsou regresní parametry, které se snažíme odhadnout.

Lineární regresi můžeme vyjádřit i obecněji ve tvaru

$$\mathbf{y} = \mathbf{X}\boldsymbol{w} + \boldsymbol{\varepsilon}, \tag{1.2}$$

kde **y** je n-rozměrný náhodný vektor se složkami $\mathbf{y} = (y_1, \ldots . . y_n)^{\mathrm{T}}$, který v regresních úlohách chceme predikovat na základě znalosti matice **X,** kde $\mathbf{X}_i = \left(x_{i1}, \ldots . . x_{ip}\right)$ je i-tý řádek této matice, který odpovídá hodnotě $y_i \in \mathbf{y}$, $\boldsymbol{\varepsilon}$ je vektor náhodných chyb. Vzorec (1.2) můžeme rozepsat do tvaru

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1p} \\ 1 & x_{21} & \ldots & x_{2p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \ldots & x_{np} \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix} + \begin{pmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}. \tag{1.3}$$

Vektor parametrů $\boldsymbol{\beta}$ může být vypočtena různými způsoby, jako třeba metodou nejmenších čtverců, která je vyučována v každém základním VŠ kurzu statistiky. Nebo, jak v této práci uvádím, bayesovskou inferencí. Při vypracování této podkapitoly byly využity zdroje [1], [10].

## 1.1.2 Umělé neuronové sítě

### 1.1.2.1 Percepron

Perceptron [1], [7] je základní jednotkou dopředné neuronové sítě. Jedná se o krabičku, do které putuje vektor vstupních informací $\mathbf{x} = [x_1, x_2, \dots, x_n]$ a vystupuje výstupní informace $y$. Výstup perceptronu můžeme matematicky zapsat následovně

$$y = f(\boldsymbol{w} \cdot \mathbf{x} + \theta) = f(\sum_{i=1}^{N} w_i x_i + \theta), \tag{1.4}$$

kde $\boldsymbol{w} = [w_1, w_2, \dots, w_n]$ je vektor vah, $\theta$ prahová hodnota neuronu (takzvaný bias) a $f$ přenosová funkce. Existence parametru $\theta$ není podmínkou pro fungování neuronové sítě. Při vypracování této podkapitoly byl využit zdroj [7].

### 1.1.2.2 Přenosové funkce

V této podkapitole jsou uvedeny nejzákladnější 3 nejznámější přenosové funkce neuronové sítě.

#### Lineární (identická) funkce

V případě, že tuto funkci aplikujeme na perceptron, přejde rovnice (1.4) do tvaru

$$y = \boldsymbol{w} \cdot \mathbf{x} + \theta = \sum_{i=1}^{N} w_i x_i + \theta. \tag{1.5}$$

Předešlý výraz je použitelný pro samostatnou lineární regresi či klasifikaci. Výhodou této funkce v rámci neuronových sítí je její malá výpočetní náročnost. Tato funkce se často používá jako aktivační funkce pro výstup neuronové sítě.

#### Sigmoidní funkce

Použitím sigmoidní funkce přejde výraz (1.4) do tvaru

$$y = \frac{1}{1+\mathrm{e}^{-(\boldsymbol{w}\cdot\mathbf{x}+\theta)}} = \frac{1}{1+\mathrm{e}^{-\left(\sum_{i=1}^{N} w_i x_i+\theta\right)}}. \tag{1.6}$$

Rovnice (1.6) bývá hojně využívána v tzv. logistické regresi a klasifikaci, v klasifikaci dosahuje lepších výsledků než lineární klasifikátor. Mezi výhody této funkce v dopředných neuronových sítích patří to, že libovolnému číslu dá hodnotu 0–1. Naopak nevýhodou je takzvaný problém mizejícího gradientu při použití algoritmů stochaického gradientního sestupu.

#### Hyperbolický tangens

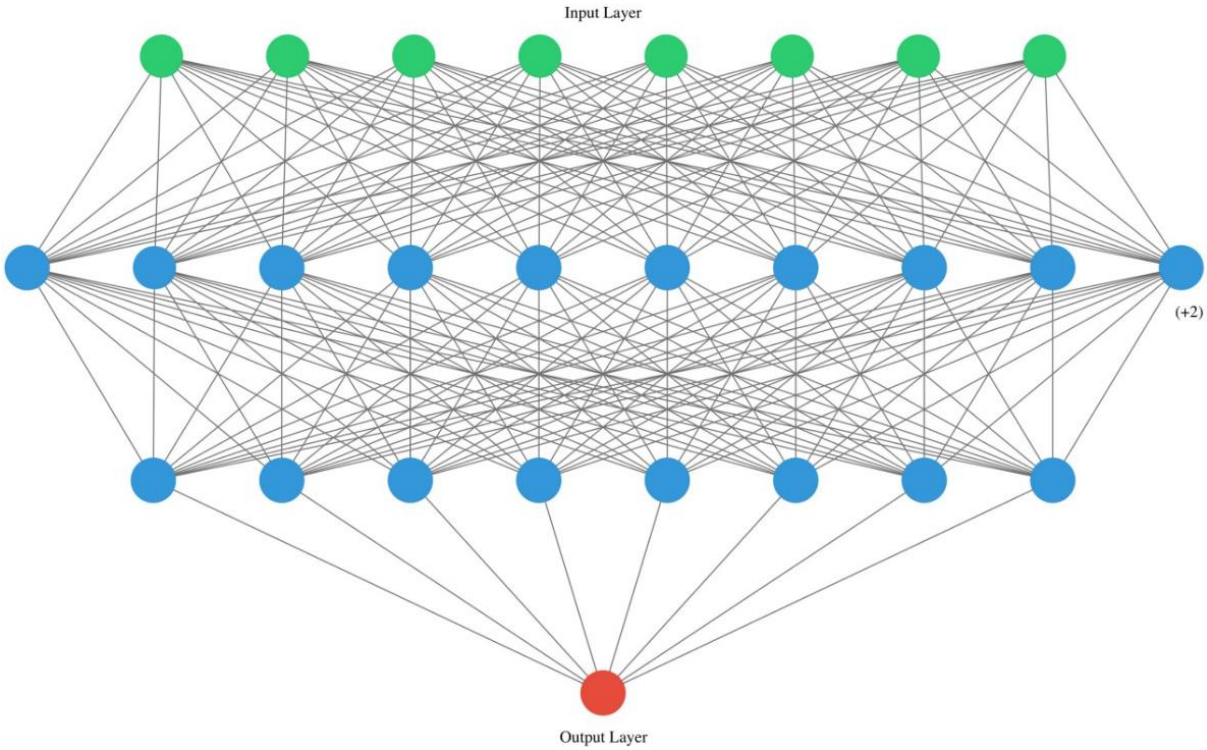
Použitím funkce hyperbolický tangens přejde výraz (1.4) do tvaru

$$y = \frac{1-\mathrm{e}^{2(\boldsymbol{w}\cdot\mathbf{x}+\theta)}}{1+\mathrm{e}^{2(\boldsymbol{w}\cdot\mathbf{x}+\theta)}} = \frac{1-\mathrm{e}^{2\cdot\left(\sum_{i=1}^{N} w_i x_i+\theta\right)}}{1+\mathrm{e}^{2\cdot\left(\sum_{i=1}^{N} w_i x_i+\theta\right)}}. \tag{1.7}$$

Mezi hlavní výhody této funkce v neuronových sítích patří to, že výstup této funkce vždy leží v rozsahu -1–1. Pokud jde o nevýhody, trpí problémem mizejícího gradientu a je náročnější ve výpočetní složitosti během trénování než sigmoidní funkce. Při vypracování této podkapitoly byl využit zdroj [8].

### 1.1.2.2 Dopředná vícevrstvá neuronová síť

Dopředná vícevrstvá neuronová síť je matematická struktura, jejíž cílem je provést regresi nebo klasifikaci na základě vstupních dat. Neuronová síť se skládá z perceptronů obsažených ve vrstvách, všechny neurony ve vrstvě $\boldsymbol{l}_i$ jsou váhami spojeny se všemi neurony ve vrstvě $\boldsymbol{l}_{i+1}$ a přenášejí do nich informace. Poslední vrstva neuronové sítě je váhami spojena s výstupy sítě. Pro každý jednotlivý perceptron v jakékoli vrstvě platí matematická pravidla popsaná v podkapitole Perceptron.




Obrázek 1.1: Struktura dvojvrstvé neuronové sítě, zdroj [2]

Nyní si matematicky popíšeme přenos informací v neuronové síti na obrázku 1.1, protože tato neuronová síť má velmi podobnou strukturu jako síť použitá v praktické části této práce. Mějme vektor vstupních dat $\mathbf{x}$, naše síť má dvě skryté vrstvy $\boldsymbol{l_1}$ a $\boldsymbol{l}_2$, biasy v první a druhé vrstvě jsou vektory a značíme je $\boldsymbol{\theta}_1$, $\boldsymbol{\theta}_2$. Bias přináležící výstupnímu neuronu je skalár a značíme ho $\theta_3$. Mezi vstupy a první skrytou vrstvou je matice vah $\mathbf{W_1}$, mezi skrytými vrstvami matice $\mathbf{W}_2$ a mezi druhou skrytou vrstvou a výstupním neuronem matice $\mathbf{W}_3$. Aktivační funkci pro první skrytou vrstvu budeme značit $f_1$, pro druhou $f_2$ a pro výstupní neuron $f_3$. Výstupní neuron (výstup sítě, který hledáme) označíme $y$. Jednotlivé rovnice pro mezivrstvy a výstupní vrstvu zapíšeme jako

$$\boldsymbol{l_1} = f_1(\mathbf{W}_1 \cdot \mathbf{x}^{\mathrm{T}} + \boldsymbol{\theta}_1), \tag{1.8}$$

$$\boldsymbol{l_2} = f_1(\mathbf{W}_2 \cdot \boldsymbol{l}_1^{\mathrm{T}} + \boldsymbol{\theta}_2), \tag{1.9}$$

$$y = f_3(\mathbf{W}_3 \cdot \boldsymbol{l}_2^{\mathrm{T}} + \theta_3). \tag{1.10}$$

Pokud bychom chtěli tři výše uvedené vztahy napsat do jednoho řádku, použijeme rovnici

$$y = f_3\{\mathbf{W}_3 \cdot [f_2(\mathbf{W}_2 \cdot f_1(\mathbf{W}_1 \cdot \mathbf{x}^{\mathrm{T}} + \boldsymbol{\theta}_1) + \boldsymbol{\theta}_2) + \theta_3]\}. \tag{1.11}$$

Definovali jsme si základní struktury použitých metod strojového učení, které budou použity v praktické části této práce, nyní se pojďme bavit o matematice, která slouží k nalezení parametrů výše zmíněných metod. Při vypracování této podkapitoly byly využity zdroje [1], [7].

## 1.2 Pravděpodobnostní rozdělení

V rámci této kapitoly se seznámíme s tím, co je spojité pravděpodobnostní rozdělení a zmíníme ta, která jsou použita v praktické části této práce. Při vypracování této kapitoly byly využity zdroje [6], [9].

### 1.2.1 Pravděpodobnostní rozdělení obecně

Jestliže mluvíme o pravděpodobnostním rozdělení, mluvíme o matematickém popisu náhodného jevu. Spojitá náhodná veličina má spojitou distribuční funkci $F(x)$. Tato funkce udává pravděpodobnost, že náhodná veličina nabyde hodnoty menší nebo rovné $x$. Další velmi důležitou veličinou je tzv. hustota pravděpodobnosti $f(x)$. Jedná se o funkci, jejíž integrál od $x_1$ do $x_2$ udává pravděpodobnost, že náhodná veličina leží mezi těmito čísly. Celkový obsah pod křivkou $f(x)$ je roven jedné. Pro hustotu pravděpodobnosti náhodné veličiny tedy platí následující vztahy

$$\int_{\Omega} f(x)\mathrm{d}x = 1, \tag{1.12}$$

$$\int_{x_1}^{x_2} f(x)\mathrm{d}x = P[x_1 \leq X \leq x_2], \tag{1.13}$$

kde $\Omega$ je definiční obor veličiny $X$. Pro hodnoty $x$ mimo definiční obor $\Omega$ je hustota pravděpodobnosti nulová.

Mezi hustotou pravděpodobnosti a distribuční funkcí platí následující vztah (v případě, že derivace distribuční funkce v daném bodě $x$ existuje)

$$f(x) = \frac{\mathrm{d}F(x)}{\mathrm{d}x}. \tag{1.14}$$

Při vypracování této podkapitoly byl využit zdroj [9].

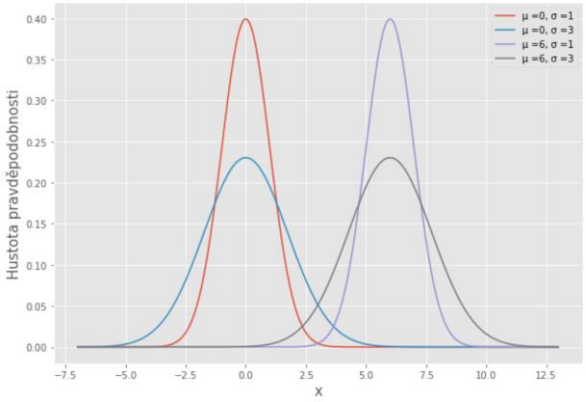
### 1.2.2 Vybraná spojitá pravděpodobnostní rozdělení

V této kapitole jsou rozebrány pouze rozdělení, která jsou použita v praktické části bakalářské práce.

Normální rozdělení

Normální rozdělení patří mezi nejdůležitější rozdělení pravděpodobnosti spojité náhodné veličiny. Toto rozdělení charakterizují dvě veličiny: střední hodnota $\mu$ a rozptyl $\sigma^2$. Hustota pravděpodobnosti tohoto rozdělení je dána vztahem

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{\frac{(x-\mu)^2}{2\sigma^2}}. \tag{1.15}$$

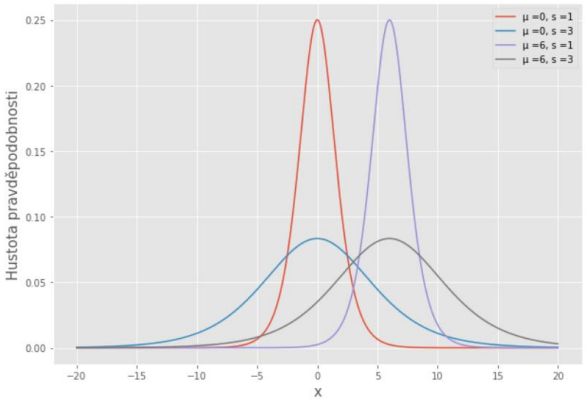



Obrázek 1.2: Grafické zobrazení hustot pravděpodobnosti normálního rozdělení při různých parametrech tohoto rozdělení

Logistické rozdělení

Distribuční funkce logistického rozdělení se používá při logistické regresi a klasifikaci a jako aktivační funkce u dopředných neuronových sítí. Toto rozdělení charakterizují dvě konstanty: střední hodnota $\mu$ a hodnota $s$. Hustota pravděpodobnosti tohoto rozdělení je dána vztahem

$$f(x) = \frac{\mathrm{e}^{(x-\mu)/s}}{s\left(1+\mathrm{e}^{(x-\mu)/s}\right)^2}. \tag{1.16}$$



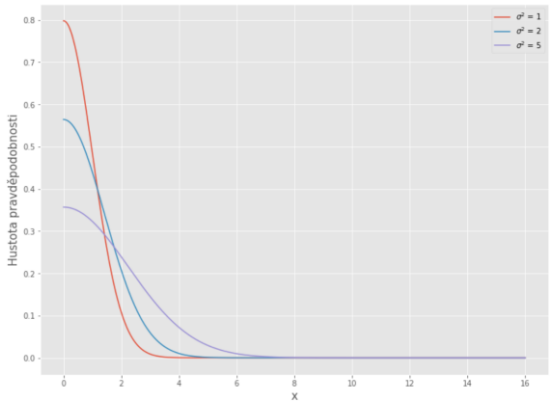

Obrázek 1.3: Grafické zobrazení hustot pravděpodobnosti logistického rozdělení při různých parametrech tohoto rozdělení

Polonormální rozdělení

Je zvláštním případem normálního rozdělení, kde $\mu = 0$ a $x \geq 0$. Běžně se používá jako apriorní pravděpodobnost pro parametry v aplikacích bayesovské inference. Hustota pravděpodobnosti tohoto rozdělení je dána předpisem

$$f(x) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} e^{\frac{x^2}{2\sigma^2}}. \tag{1.17}$$

Čím vyšší je parametr $\sigma$, tím vyšší hustota pravděpodobnosti je přikládána větším hodnotám, viz. Obrázek 1.4.




Obrázek 1.4: Grafické zobrazení hustot pravděpodobnosti polonormálního rozdělení při různých parametrech tohoto rozdělení

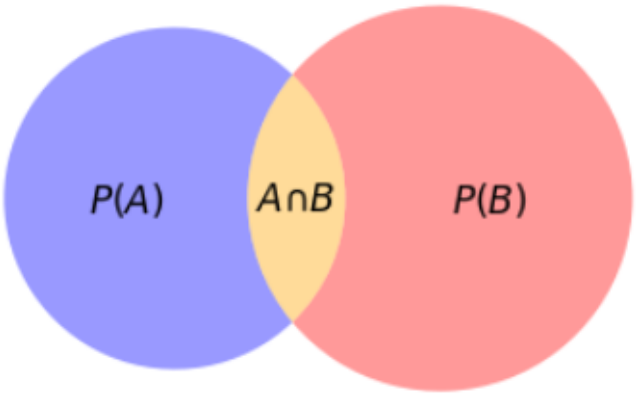
Pro vypracování této podkapitoly byl využit zdroj [6].

## 1.3 Bayesovská inference

V této kapitole si odvodíme Bayesovu větu jak pro diskrétní, tak spojité náhodné veličiny. Popíšeme si jednotlivé členy v této rovnici a ukážeme základní vztahy pro odhad pomocí maxima posteriorního rozdělení. V této kapitole byly využity zdroje [1], [3], [11].

### 1.3.1 Bayesův vzorec

Mějme dva jevy, $A$ a $B$, které mohou nastat každý zvlášť, ale i současně. Jejich pravděpodobnosti si lze zobrazit pomocí tzv. Vennových diagramů.




Obrázek 1.5: Vennovy diagramy

$P(A)$ je pravděpodobnost jevu $A, P(B)$ je pravděpodobnost jevu $B$ a $P(A \cap B)$ je pravděpodobnosti, že oba jevy nastanou současně. Pravděpodobnost, že jevy $A$ a $B$ nastanou současně lze vyjádřit následujícími dvěma způsoby, jde o tzv. pravidlo součinu

$$P(A \cap B) = P(A|B)P(B), \tag{1.18}$$

$$P(B \cap A) = P(B|A)P(A), \tag{1.19}$$

kde $P(A|B)$ je pravděpodobnost, že nastane jev $A$, když nastal jev $B$ a kde $P(B|A)$ je pravděpodobnost, že nastane jev $B$, když nastal jev $A$. Obě pravděpodobnosti $P(A \cap B)$ a $P(B \cap A)$ jsou stejné, graficky je lze znázornit jako žlutou plochu v našem Vennově diagramu (viz. Obrázek 1.5), platí tedy rovnosti

$$P(A \cap B) = P(B \cap A), \tag{1.20}$$

$$P(A|B)P(B) = P(B|A)P(A). \tag{1.21}$$

Nyní už jen upravíme vzorec (1.21) a získáme Bayesovu větu

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \tag{1.22}$$

Jmenovatele Bayesovy věty lze vyjádřit jako

$$P(B) = \sum_{i=1}^{N} P(B|A_i)P(A_i). \tag{1.23}$$

Dosazením tohoto vztahu do vzorce (1.22) získáme Bayesovu větu ve tvaru

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_{i=1}^{N} P(B|A_i)P(A_i)}. \tag{1.24}$$

Bayesova věta pro náhodné veličiny lze odvodit analogickým způsobem, nechť jsou $x$ a $y$ náhodné veličiny, $p(x)$ a $p(y)$ jsou pravděpodobnosti rozdělení veličin $x$ a $y$, potom pro diskrétní náhodné veličiny dostaneme Bayesovu větu ve tvaru

$$p(x|y) = \frac{p(y|x)p(x)}{\sum_{i=1}^{N} p(y|x_i)p(x_i)}. \tag{1.25}$$

A pro spojité veličiny ve tvaru

$$f(x|y) = \frac{f(y|x)f(x)}{\int_{-\infty}^{\infty} f(y|x_i)f(x_i)}. \tag{1.26}$$

V této podkapitole byly využit zdroj [3].

### 1.3.2 Aplikace Bayesova vzorce na datech

Jde o to nalézt pravděpodobnostní rozdělení parametru $\theta$ na základě dat $D$, tedy **posteriorní rozdělení** $f(\theta|D)$, které je zjištěno pomocí Bayesovy věty, která bere v úvahu:

1. **Apriorní rozdělení**, $p(\theta)$ (ve spojitém případě $f(\theta)$), která představuje naše předpoklady o parametru $\theta$.
2. **Věrohodnostní funkci**, $p(D|\theta)$ (ve spojitém případě $f(D|\theta)$ ), která vyjadřuje věrohodnost, že při daných hodnotách $\theta$ budou v modelu generována data $D$.
3. **Evidenci**, $p(D)$ (ve spojitém případě $f(D)$ ), což je celková pravděpodobnost dat $D$, která je odvozena sečtením přes všechny hodnoty $\theta$. V diskrétním případě vypočítáme evidenci jako

$$p(D) = \sum_{\theta} p(D|\theta)p(\theta). \tag{1.27}$$

Ve spojitém případě

$$f(D) = \int_{-\infty}^{\infty} f(D|\theta)f(\theta)\,\mathrm{d}\theta. \tag{1.28}$$

Posteriorní rozdělení je tedy v diskrétním případě spočtena ze vzorce

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{\sum_\theta p(D|\theta)p(\theta)}. \quad (1.29)$$

Ve spojitém případě

$$f(\theta|D) = \frac{f(D|\theta)f(\theta)}{\int_{-\infty}^{\infty} f(D|\theta)f(\theta)\mathrm{d}\theta}. \quad (1.30)$$

Pro predikci posteriorního rozdělení jednoho datového bodu $y$ použijeme následující vzorec

$$f(y) = \int_{-\infty}^{\infty} f(y|\theta)f(\theta)\mathrm{d}\theta, \quad (1.31)$$

kde $f(\theta)$ je na naše nejaktuálněji známé rozdělení parametru $\theta$, může se jednat o posteriorní hodnotu parametru $\theta$, nebo jeho apriorní hodnotu.

Evidence se v některých případech vynechává (např. MAP odhad), protože nezávisí na parametru $\theta$. Rovnost je tedy nahrazena úměrností, ve spojitém případě platí následující vztah

$$f(\theta|D) \propto f(D|\theta)f(\theta). \quad (1.32)$$

V této podkapitole byl využit zdroj [1].

### 1.3.3 Odhad pomocí maxima posteriorního rozdělení (MAP)

Jedná se o bodový odhad parametru $\theta$ tak, aby bylo nalezeno maximum posteriorního rozdělení, tedy

$$\hat{\theta}_{\mathrm{MAP}} = \mathrm{argmax}_\theta\, f(\theta|D). \quad (1.33)$$

Protože jmenovatel Bayesovy věty nezávisí na parametru $\theta$, můžeme napsat

$$\hat{\theta}_{\mathrm{MAP}} = \mathrm{argmax}_\theta \left(f(D|\theta)f(\theta)\right). \quad (1.34)$$

Velmi často se hledá logaritmus posteriorního rozdělení (kvůli zjednodušení výpočtu)

$$\hat{\theta}_{\mathrm{MAP}} = \arg\max_\theta \log\left(f(D|\theta)f(\theta)\right). \quad (1.35)$$

V této podkapitole byl využit zdroj [11].

## 1.4 Použití bayesovských metod

V této kapitole si na příkladech ukážeme praktické využití bayesovské inference. Čerpáno ze zdroje [1].

Příklad č. 1

Mějme vstupní data **x** a výstupní data **y** (budeme značit písmenem $D$). Nechť má každý výstupní datový bod $y_i$ normální rozdělení pravděpodobnosti. Cílem je na základě vstupních dat **x** a výstupních dat **y** nalézt posteriorní rozdělení parametrů $a$ a $b$ přímky, kterou definujeme jako $\mathbf{y} = a\mathbf{x} + b$.

Nejdříve definujeme apriorní rozdělení koeficientů $a, b$. Předpokládáme, že mají normální rozdělení pravděpodobnosti s hyperparametry $a_0 = 0,\ b_0 = 0$. Hodnoty těchto hyperparametrů si sami volíme. Mezi dalšími hyperparametry, jejichž hodnoty si sami volíme, jsou $\sigma_a$, $\sigma_b$ a $\sigma$. Jedná se o naše předpoklady o směrodatných odchylkách parametrů $a, b$ a věrohodnostní funkce. Apriorní rozdělení parametrů $a$ a $b$ se v případě předpokladu, že jsou normálně rozdělené, definují jako

$$f(a|\, a_0, \sigma_a) = \frac{1}{\sigma_a\sqrt{2\pi}} \mathrm{e}^{-\frac{(a-a_0)^2}{2{\sigma_a}^2}} = \frac{1}{\sigma_a\sqrt{2\pi}} \mathrm{e}^{-\frac{a^2}{2{\sigma_a}^2}},$$

$$f(b|b_0, \sigma_b) = \frac{1}{\sigma_b\sqrt{2\pi}} \mathrm{e}^{-\frac{(b-b_0)^2}{2{\sigma_b}^2}} = \frac{1}{\sigma_b\sqrt{2\pi}} \mathrm{e}^{-\frac{b^2}{2{\sigma_b}^2}}.$$

Následně definujeme věrohodnostní funkci

$$f(D|a, b, \sigma) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{\left(y_i-(ax_i+b)\right)^2}{2\sigma^2}}.$$

Nyní s využitím rovnice (1.32) získáme

$$f(a, b, \sigma|D) \propto \frac{1}{\sigma_a\sqrt{2\pi}} \mathrm{e}^{-\frac{a^2}{2{\sigma_a}^2}} \frac{1}{\sigma_b\sqrt{2\pi}} \mathrm{e}^{-\frac{b^2}{2{\sigma_b}^2}} \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{\left(y_i-(ax_i+b)\right)^2}{2{\sigma_a}^2}}.$$

Teď zlogaritmujeme předchozí rovnici (přirozeným logaritmem) a získáme

$$f(a, b, \sigma|D) \propto \log\left(\frac{1}{\sigma_a\sqrt{2\pi}}\right) - \frac{a^2}{2\sigma_a^2} + \log\left(\frac{1}{\sigma_b\sqrt{2\pi}}\right) - \frac{b^2}{2\sigma_b^2} + \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - ax_i - b)^2.$$

Následně derivujeme předchozí funkci podle koeficientů $a$ a $b$. Tyto derivace budou rovny nule, jde tedy o to najít extrémy posteriorní funkce vzhledem k daným parametrům. Dostaneme rovnice

$$\frac{df(a,b,\sigma|D)}{da} = \frac{d}{da}\left[-\frac{a^2}{2\sigma_a^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - ax_i - b)^2\right] = -\frac{a}{\sigma_a^2} + \frac{1}{\sigma^2}\sum_{i=1}^{N} x_i(y_i - ax_i - b) = 0, \qquad (1.36)$$

$$\frac{df(a,b,\sigma|D)}{db} = \frac{d}{db}\left[-\frac{b^2}{2\sigma_b^2} - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - ax_i - b)^2\right] = -\frac{b}{\sigma_b^2} + \frac{1}{\sigma^2}\sum_{i=1}^{N}(y_i - ax_i - b) = 0. \qquad (1.37)$$

Z rovnice (1.36) zjistíme koeficient $a$ a z rovnice (1.37) zjistíme koeficient $b$. Aritmetický průměr pozorovaných vstupních dat je $\bar{x}$ a průměr pozorovaných dat budeme značit $\bar{y}$. $N$ je celkový počet dat, které k bayesovské inferenci používáme. Dostaneme

$$a = \frac{\sum_{i=1}^{N} x_i y_i - N b \bar{x}}{\frac{\sigma^2}{\sigma_a^2} + \sum_{i=1}^{N} x_i^2},$$

$$b = \frac{N}{\frac{\sigma^2}{\sigma_a^2} + N}(\bar{y} - a\bar{x}).$$

Poměr $\frac{\sigma^2}{\sigma_a^2}$ dále budeme značit $E$, jedná se o náš odhad. Celkovým vyřešením soustavy rovnic (1.36) a (1.37) dostaneme

$$a = \frac{(E - N)\sum_{i=1}^{N} x_i y_i - N^2 \bar{x}\bar{y}}{(E + \sum_{i=1}^{N} x_i^2)(E + N) - N^2 \bar{x}^2},$$

$$b = \frac{N}{E + N}(\bar{y} - a\bar{x}).$$

Příklad č. 2

V tomto jednoduchém příkladu lineární regrese si přiblížíme, jak funguje bayesovská inference, řešíme tedy příklad č.1, jen ne jako MAP odhad. Věrohodnostní funkce je ve tvaru

$$f(D|a, b, \sigma) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{(y_i - a x_i - b)^2}{2\sigma^2}} = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^N \mathrm{e}^{-\frac{\sum_{i=1}^{N}(y_i - a x_i - b)^2}{2\sigma^2}},$$

kde $\sigma$ je číslo, které si v *Pymc3* můžeme definovat jako jedno dané číslo nebo celé pravděpodobnostní rozdělení.

Teď definujme apriorní rozdělení parametrů $a$, $b$ a $\sigma$, náš předpoklad je, že parametry $a$, $b$ budou mít normální rozdělení a $\sigma$ polonormální rozdělení

$$f(a|a_0, \sigma_a) = \frac{1}{\sigma_a\sqrt{2\pi}} \mathrm{e}^{-\frac{(a - a_0)^2}{2{\sigma_a}^2}},$$

$$f(b|b_0, \sigma_b) = \frac{1}{\sigma_b\sqrt{2\pi}} \mathrm{e}^{-\frac{(b - b_0)^2}{2{\sigma_b}^2}},$$

$$f(\sigma|\sigma_0) = \frac{\sqrt{2\pi}}{\sigma_0\sqrt{\pi}} \mathrm{e}^{-\frac{\sigma^2}{2\sigma_0^2}},$$

kde $\sigma_a$ a $a_0$ a $\sigma_b$, $b_0$ a $\sigma_0$ jsou hyperparametry apriorního rozdělení, v *Pymc3* je definujeme jako reálná čísla.

Posteriorní rozdělení vyjádříme podle vzorce

$$f(a,b,\sigma|D) = \frac{f(a|a_0,\sigma_a)f(b|b_0,\sigma_b)f(\sigma|\sigma_0)f(D|a,b,\sigma))}{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(a|a_0,\sigma_a)f(b|b_0,\sigma_b)f(\sigma|\sigma_0)f(D|a,b,\sigma))\,\mathrm{d}a\mathrm{d}b\mathrm{d}\sigma}.$$

V *Pymc3* můžeme definovat $\sigma_a$, $a_0$, $b_0, \sigma_b$ $\sigma_0$, takto: $\sigma_a = \sigma_b = \sigma_0 = 1$ a $a_0 = b_0 = 0$ – naše předpoklady. Po dosazení apriorních rozdělení a věrohodnostní funkce do Bayesovy věty získáme

$$f(a,b,\sigma|D) = K \cdot \mathrm{e}^{-\frac{1}{2}(a^2+b^2+\sigma^2+\sum_{i=1}^{N}(y_i-ax_i-b)^2)},$$

kde $K$ je normalizační člen, který lze vyjádřit vztahem

$$K = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\infty} \mathrm{e}^{-\frac{1}{2}(a^2+b^2+\sigma^2+\sum_{i=1}^{N}(y_i-ax_i-b)^2)}\,\mathrm{d}a\mathrm{d}b\mathrm{d}\sigma.$$

Dostali jsme celkovou posteriorní hustotu pravděpodobnosti. Hustotu pravděpodobnosti pro koeficient $a$ získáme

$$f(a|D) = K \cdot \int_{-\infty}^{\infty}\int_{0}^{\infty} \mathrm{e}^{-\frac{1}{2}(a^2+b^2+\sigma^2+\sum_{i=1}^{N}(y_i-ax_i-b)^2)}\,\mathrm{d}b\mathrm{d}\sigma.$$

A pro koeficienty $\sigma, b$ obdobně

$$f(b|D) = K \cdot \int_{-\infty}^{\infty}\int_{0}^{\infty} \mathrm{e}^{-\frac{1}{2}(a^2+b^2+\sigma^2+\sum_{i=1}^{N}(y_i-ax_i-b)^2)}\,\mathrm{d}a\mathrm{d}\sigma,$$

$$f(\sigma|D) = K \cdot \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{e}^{-\frac{1}{2}(a^2+b^2+\sigma^2+\sum_{i=1}^{N}(y_i-ax_i-b)^2)}\,\mathrm{d}a\mathrm{d}b.$$

Příklad č. 3

V rámci tohoto příkladu si nastíníme, jak se pomocí bayesovské inference určí koeficienty dopředné vícevrstvé neuronové sítě, které používáme v praktické části bakalářské práce. Naše neuronové sítě mají různý počet vstupů $N$. Mají dvě skryté vrstvy neuronů, kde v první vrstvě je 10 neuronů a ve druhé 2 neurony. Tuto strukturu jsem zvolil, protože ji používám v praktické části této práce. S biasy nepracujeme.

Matici vah mezi vstupy a první skrytou vrstvou definujeme jako

$$\mathbf{W}_1 = \begin{pmatrix} w_{1,1}^1 & w_{1,2}^1 & \dots & w_{1,n}^1 \\ w_{2,1}^1 & w_{2,2}^1 & \dots & w_{2,n}^1 \\ \vdots & \vdots & & \vdots \\ w_{10,1}^1 & w_{10,2}^1 & \dots & w_{10,n}^1 \end{pmatrix}.$$

Matici vah mezi první a druhou vrstvou neuronů jako

$$\mathbf{W}_2 = \begin{pmatrix} w_{1,1}^2 & w_{1,2}^2 & \dots & w_{1,10}^2 \\ w_{2,1}^2 & w_{2,2}^2 & \dots & w_{2,10}^2 \end{pmatrix}.$$

A mezi druhou vrstvou a výstupem neuronové sítě jako

$$\mathbf{W}_3 = (w_1^3 \quad w_2^3).$$

Celkový výstup neuronové sítě lze zapsat následovně

$$y_i = f_3\left\{\mathbf{W}_3 \cdot \left[f_2\left(\mathbf{W_2} \cdot f_1\left(\mathbf{W}_1 \cdot \mathbf{x}_i^{\mathrm{T}}\right)\right)\right]\right\},$$

kde $f_1,\ f_2\ a\ f_3$ jsou aktivační funkce a $\mathbf{x}_i^{\mathrm{T}}$ vstupní vektor dat.

Věrohodnostní funkce má logistické rozdělení

$$f(D|\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3) = \prod_{i=1}^{N} \frac{\mathrm{e}^{-\left(y_i - f_3\left\{\mathbf{W}_3 \cdot \left[f_2\left(\mathbf{W}_2 \cdot f_1\left(\mathbf{W}_1 \cdot \mathbf{x}_i^{\mathrm{T}}\right)\right)\right]\right\}\right)/s}}{s\left(1 + \mathrm{e}^{-\left(y_i - f_3\left\{\mathbf{W}_3 \cdot \left[f_2\left(\mathbf{W}_2 \cdot f_1\left(\mathbf{W}_1 \cdot \mathbf{x}_i^{\mathrm{T}}\right)\right)\right]\right\}\right)/s}\right)^2}.$$

Pokud o parametru $s$ nemáme žádné informace, nemůžeme mu přisoudit žádnou hodnotu, proto mu přiřadíme polonormální rozdělení s hyperparametrem $s_0$.

Nyní si definujme matice $\mathbf{W}_0^1$, $\mathbf{W}_0^2$ a $\mathbf{W}_0^3$. Matice $\mathbf{W}_0^1$ má stejné rozměry jako matice $\mathbf{W}_1$. To stejné platí pro $\mathbf{W}_0^2$ a $\mathbf{W}_2$ a taktéž pro $\mathbf{W}_0^3$ a $\mathbf{W}_3$. Prvky těchto nových matic odráží naše předpoklady o středních hodnotách jednotlivých vah neuronové sítě a jsou obsaženy v normálních apriorních rozděleních zvlášť pro $\mathbf{W}_1$, $\mathbf{W}_2$ a $\mathbf{W}_3$**.** Hyperparametry $\sigma_1$ , $\sigma_2$ a $\sigma_3$ jsou naše předpoklady o směrodatných odchylkách všech prvků matic $\mathbf{W}_1$, $\mathbf{W}_2$ a $\mathbf{W}_3$.

Jednotlivá apriorní rozdělení našeho modelu pro jednotlivé váhy a koeficient $s$ jsou

$$f(\mathbf{W}_1|\mathbf{W}_{01}, \sigma_1) = \frac{1}{\sigma_1^2\sqrt{2\pi}}\mathrm{e}^{-\frac{\sum_{i=1}^{10}\sum_{j=1}^{n}\left(w_{i,j}^1 - w_{0\,i,j}^1\right)^2}{2\sigma_1^2}},$$

$$f(\mathbf{W}_2|\mathbf{W}_{02}, \sigma_2) = \frac{1}{\sigma_2^2\sqrt{2\pi}}\mathrm{e}^{-\frac{\sum_{m=1}^{2}\sum_{n=1}^{10}\left(w_{m,n}^2 - w_{0\,m,n}^2\right)^2}{2\sigma_2^2}},$$

$$f(\mathbf{W}_3|\mathbf{W}_{03}, \sigma_3) = \frac{1}{\sigma_3^2\sqrt{2\pi}}\mathrm{e}^{-\frac{\sum_{i=1}^{2}\left(w_i^3 - w_{0\,i}^3\right)^2}{2\sigma_3^2}},$$

$$f(s|s_0) = \frac{\sqrt{2\pi}}{\sigma_0\sqrt{\pi}}\mathrm{e}^{-\frac{s^2}{2s_0^2}}.$$

Posteriorní funkci ze znalosti věrohodnostní funkce a apriorních rozdělení napíšeme následovně

$$f(\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, s|D) = \frac{f(D|\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3) f(\mathbf{W}_1|\mathbf{W}_{01}, \sigma_1) f(\mathbf{W}_2|\mathbf{W}_{02}, \sigma_2) f(\mathbf{W}_3|\mathbf{W}_{03}, \sigma_3) f(s|s_0)}{\int_{-\infty}^{\infty} \dots \int_{s=0}^{\infty} f(D|\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3) f(\mathbf{W}_1|\mathbf{W}_{01}, \sigma_1) f(\mathbf{W}_2|\mathbf{W}_{02}, \sigma_2) f(\mathbf{W}_3|\mathbf{W}_{03}, \sigma_3) f(s|s_0) \mathrm{d}\mathbf{W}_1 \mathrm{d}\mathbf{W}_2 \mathrm{d}\mathbf{W}_3 \mathrm{d}s},$$

Pokud bychom chtěli získat posteriorní hodnoty jedné dané váhy (například $w_{11}$), použijeme vztah

$$f(w_{11}|D) = \int_{-\infty}^{\infty} f(\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, s|D)\, \mathrm{d}\mathbf{W}_{1/w_{1,1}^1} \mathrm{d}\mathbf{W}_2 \mathrm{d}\mathbf{W}_3 \mathrm{d}s,$$

který říká, že k tomu, abychom dostali posteriorní rozdělení váhy $w_{1,1}^1$, musíme celkovou posteriorní funkci integrovat přes všechny váhy, kromě váhy $w_{1,1}^1$.

Protože je neuronová síť nelineární matematická struktura, jednotlivé střední hodnoty ani posteriorní rozdělení vah sítě nám o této síti příliš neřeknou. Naskýtá se otázka, proč tedy nezjišťujeme pro každou jednotlivou váhu pouze jedno konkrétní číslo, kterému tato váha odpovídá. Odpověď je skryta ve vzorci (1.31), jehož přizpůsobením pro tento problém dostaneme následující vztah, který právě ze znalosti posteriorních rozdělení jednotlivých vah a věrohodnostní funkce zjistí posteriorní rozdělení výstupních dat

$$f(y_i) = \int_{-\infty}^{\infty} f(y_i|\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, s) f(\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, s) \mathrm{d}\mathbf{W}_1 \mathrm{dW}_\mathbf{2} \mathrm{d}\mathbf{W}_3 \mathrm{d}s.$$

# Kapitola 2
# Praktická část

V této kapitole si ukážeme, s jakými daty pracujeme, a vhodně je vizualizujeme. Poté data připravíme tak, abychom s nimi byli schopni dělat prediktivní modely strojového učení, a to lineární regresi a dopředné vícevrstvé neuronové sítě. Nejdříve začneme s nejméně přesnými lineárními modely pro předpověď výroby elektřiny na den dopředu $(D + 1)$ pro každou hodinu dalšího dne. Tyto lineární modely postupně vylepšujeme, dokud nenajdeme ten nejpřesnější možný model. Pokračujeme s předpověďmi výroby elektřiny na dvě hodiny dopředu $(H + 2)$ v rámci jednoho dne. Následně vytvoříme několik neuronových sítí, které srovnáme s lineárními modely. Každá neuronová síť má stejný počet skrytých vrstev a počty neuronů ve vrstvách, liší se jen v počtu vstupů. Nakonec vytvoříme dva kombinované modely, jeden pro $D + 1$ a druhý pro $H + 2$ tak, že vezmeme v úvahu nejlepší modely neuronových sítí a nejlepší lineární modely.

## 2.1 Data

### 2.1.1 Pracovní data

Tabulka 2.1 ukazuje výčet dat, se kterými budeme pracovat.

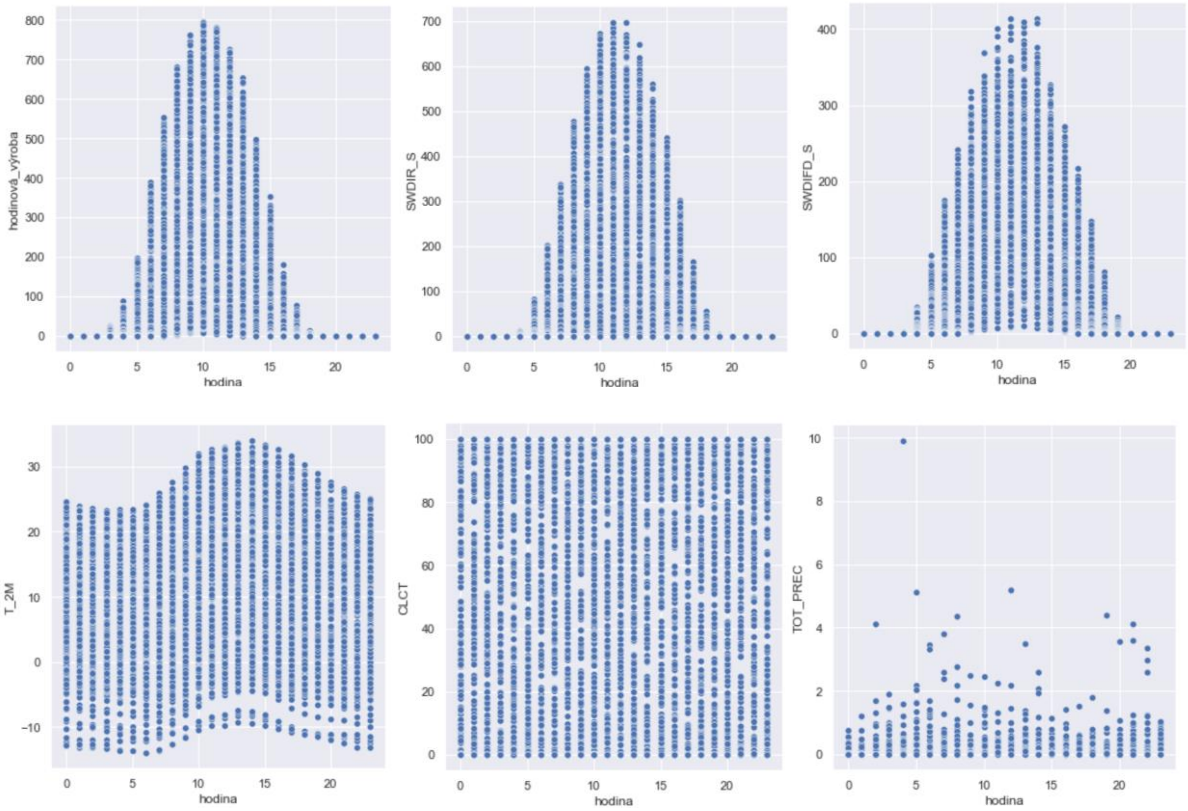
Tabulka 2.1: Tabulka se všemi pracovními daty

| | datum | hodinová_výroba | TOT_PREC | T_2M | SWDIFD_S | SWDIR_S | CLCT |
|---|---|---|---|---|---|---|---|
| **0** | 2018-01-01 00:00:00+00:00 | 0.0 | 0.0 | 4.425928 | 0.0 | 0.0 | 27.904297 |
| **1** | 2018-01-01 01:00:00+00:00 | 0.0 | 0.0 | 3.439615 | 0.0 | 0.0 | 69.236328 |
| **2** | 2018-01-01 02:00:00+00:00 | 0.0 | 0.0 | 2.770166 | 0.0 | 0.0 | 44.728516 |
| **3** | 2018-01-01 03:00:00+00:00 | 0.0 | 0.0 | 2.323419 | 0.0 | 0.0 | 100.000000 |
| **4** | 2018-01-01 04:00:00+00:00 | 0.0 | 0.0 | 2.825815 | 0.0 | 0.0 | 100.000000 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **14588** | 2019-08-31 20:00:00+00:00 | 0.0 | 0.0 | 23.060938 | 0.0 | 0.0 | 94.085938 |
| **14589** | 2019-08-31 21:00:00+00:00 | 0.0 | 0.0 | 22.113184 | 0.0 | 0.0 | 74.666016 |
| **14590** | 2019-08-31 22:00:00+00:00 | 0.0 | 0.0 | 21.540918 | 0.0 | 0.0 | 19.156250 |
| **14591** | 2019-08-31 23:00:00+00:00 | 0.0 | 0.0 | 21.478662 | 0.0 | 0.0 | 1.218750 |
| **14592** | 2019-09-01 00:00:00+00:00 | 0.0 | 0.0 | 21.575342 | 0.0 | 0.0 | 8.093750 |

14593 rows × 7 columns

Data ukazují hodinové výroby solární elektrárny v Tlumačově v daných časových hodinových intervalech a k nim příslušné hodnoty počasí v Tlumačově od 1.1.2018 do 1.9.2019. Každý jednotlivý řádek v tabulce *info* (Tabulka 2.1) udává data v daný rok, měsíc a hodinový interval. První sloupec (*datum*) je zřejmý, druhý sloupec (*hodinová_výroba*) udává výroby elektřiny v kWh, třetí sloupec (*TOT_PREC*) udává celkový úhrn srážek v $\mathrm{kg} \cdot \mathrm{m}^{-2}$, čtvrtý sloupec (*T_2M*) je teplota vzduchu ve dvou metrech nad zemským povrchem. Teplota je udávána

ve stupních Celsia. Pátý sloupec (*SWDIFD_S*) zobrazuje údaje o difuzním světle, jehož jednotkou je $\mathrm{W} \cdot \mathrm{m}^{-2}$, šestý sloupec (*SWDIR_S*) obsahuje informace o přímém světle, jehož jednotkou je taktéž $\mathrm{W} \cdot \mathrm{m}^{-2}$ a v sedmém sloupci (*CLCT*) jsou zaznamenány údaje o oblačnosti (uvedena v procentech).




Obrázek 2.1–2.6: Vizualizace výstupních a vstupních dat vzhledem k jednotlivým hodinám přes celý rok 2018

Uvedené obrázky ukazují grafické vyjádření použitých dat z tabulky *info* pro rok 2018 v jednotlivých hodinách. Z grafů lze zjistit, že veličiny *hodinová_výroba*, *SWDIR_S* a *SWDIFD_S* mají velmi podobný periodický charakter během dne. Předpokládáme tedy, že *SWDIR_S* a *SWDIFD_S* jsou nejdůležitějšími veličinami pro predikci veličiny *hodinová_výroba*, což i logicky odpovídá, protože obě zmíněné veličiny udávají informaci o tom, jak moc v daných hodinách svítilo. I laikovi je zřejmé, že osvit je pro výrobu solární elektrárny velmi důležitý. Graf pro *T_2M* ukazuje naši každodenní zkušenost s venkovním teploměrem, kdy je ráno teplota nízká, během dopoledne roste a k večeru klesá. Veličiny *CLCT* a *TOT_PREC* nemají žádný periodický charakter, ale přesto jsou důležité pro přesnější předpověď veličiny *hodinová_výroba*.

### 2.1.2 Příprava dat pro modely strojového učení

V této kapitole popisujeme, jak jsou zpracována data, která použijeme pro natrénování parametrů modelů lineární regrese a umělých neuronových sítí.

Nejdříve jsou importovány použité knihovny, konkrétně *Pandas*, *Numpy*, *Matplotlib* a *Seaborn*. *Pandas* slouží pro práci s tabulkami, *Numpy* je základní matematická knihovna, *Seaborn* umožňuje snadno vytvářet pokročilejší grafy. Pomocí knihovny *Matplotlib* vytváříme některé grafy. Poté si importujeme data z Excelu pomocí funkce knihovny *Pandas*– `read_excel()`. Tabulku pojmenujeme *info*. Následně si vytvoříme dva sloupce s názvy *hodina* a *měsíc*.

Výrobu elektřiny predikujeme jen mezi 4–19. hodinou z důvodu, že v ostatních hodinách je výroba buď nulová, nebo zanedbatelná. V našich modelech pro predikci elektřiny $H + 2$ je třeba znát i predikci výroby elektřiny v hodinovém intervalu h pro predikci v hodině $h + 2$, proto z tabulky *info* vybereme pouze řádky, které se týkají jen hodinových intervalů 4–19.

Standardizujeme data, se kterými budeme pracovat. Výrobu elektrárny, *SWDIR_S* a *SWDIFD_S* vydělíme jejich 90% percentily (to znamená, že je podělíme hodnotou, která je větší než 90 % čísel v těchto polích), veličinu *CLCT* dělíme maximální hodnotou ve sloupci *CLCT* číslem 100. Veličinu *T_2M* jsme standardizovali pomocí směrodatné odchylky, dostali jsme průměrnou hodnotu ve standardizovaném poli rovnu nule a rozptyl čísel v tomto poli standardizované veličiny *T_2M* je roven jedné.

Z důvodu, že predikujeme výrobu elektřiny jak na den dopředu, tak na dvě hodiny dopředu, si do tabulky přidáme několik nových sloupců, abychom v každém jednotlivém sloupci měli data příslušící hodině h v daném dni a data příslušící hodině $h + 2$.

Velmi důležitou veličinou pro předpověď výroby elektřiny solární elektrárny je přímý sluneční svit *SWDIR_S*. Naskýtá se otázka, zdali v hodinách, ve kterých více svítí, mají vstupní data počasí na vyrobené elektřině jinou závislost než v hodinách, ve kterých svítí méně. To, jestli svítí méně nebo více zjistíme tak, že si vezmeme data *SWDIR_S* z tabulky *info*, která připadají nějakému hodinovému intervalu (třeba 10–11. hodině) a vypočítáme medián těchto dat. Je-li daná hodnota *SWDIR_S* v poli menší než mediánová hodnota tohoto pole, svítí méně, je-li větší, svítí více. Tento postup zopakujeme pro všechny hodinové intervaly. Hodnoty mediánů budou v nově vytvořeném sloupci *hodina_median* a údaje o tom, zdali svítí méně či více budou v novém sloupci *svit*. Svítí-li málo (méně než daná mediánová hodina *SWDIR_S*), bude v daném řádku číslo 0, v opačném případě se doplní číslo 1.

V jednom našem modelu pracujeme i s výrobou elektřiny ve dni d, abychom byli schopni o něco lépe předpovídat výrobu elektřiny ve dni $d + 1$, proto v tabulce *info* vytvoříme nový sloupec s názvem *hodinová_výroba_den_zpět*, ve kterém jsou umístěny hodnoty vyrobené elektřiny ve dni d.

Nakonec si vytvoříme dvě tabulky s názvy *train* a *test*. V tabulce *train* jsou umístěna všechna data z tabulky *info*, která odpovídají roku 2018 a v tabulce test jsou data, která

odpovídají osmi měsícům roku 2019 (taktéž převzata z tabulky *info)*. Tabulka *train* slouží k natrénování parametrů jednotlivých modelů. Data z tabulky *test* použijeme k otestování přesnosti modelů.

Nakonec si v tabulkách *train* a *test* vytvoříme nový sloupec s názvem *hodina_jinak*. V tomto sloupci jsou hodiny indexovány od nuly.

## 2.2 Použité chybové metriky

Chybové metriky slouží k číselnému vyjádření přesnosti modelů. V této práci hodnotíme přesnost použitých modelů pomocí třech metrik a to průměrné absolutní chyby (MAE z anglického *mean absolute error*), střední čtvercové chyby (MSE z anglického *mean square error*) a mediánu absolutní odchylky (MAO z češtiny kvůli odlišení od MAE). V této kapitole byl využit zdroj [12].

**Průměrná absolutní chyba MAE** je, jak již název napovídá, aritmetický průměr absolutních chyb $|e_i| = |y_i - \bar{y}_i|$, kde $y_i$ je předpověď a $\bar{y}_i$ skutečná hodnota. MAE se vypočítá následovně

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |e_i|}{n} = \frac{\sum_{i=1}^{n} |y_i - \bar{y}_i|}{n},$$

kde $n$ je počet predikovaných hodnot.

**Střední čtvercová chyba MSE** počítáme pomocí vztahu

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}{n}.$$

Tato funkce neslouží jen k evaluaci modelů strojového učení, ale i k odhadu jejich parametrů pomocí gradientního sestupu, který cyklicky mění velikost parametrů modelů tak, aby se tato chyba minimalizovala.

**Medián absolutní odchylky MAO** vypočte střední chybu predikcí testovaných dat a zjistíme jej podle vztahu

$$\mathrm{MAO} = \mathrm{median}|y_i - \bar{y}_i|.$$

## 2.3 Lineární modely

Než začneme se strojovým učením, je nutné si uvědomit, že trénovací množina slouží k nafitování parametrů modelu a testovací množina slouží k ověření přesnosti modelu na reálných (testovacích) datech. Kdybychom model testovali na trénovacích datech, nezjistili bychom, zda nedošlo k overfittingu. Chyba modelu otestovaném na trénovacích datech by mohla být velmi malá, kdežto ve skutečném využití by pro tento model nebylo místo.

Ideou mého modelování je začít s nejprimitivnějším modelem s malým počtem parametrů, tento model postupně budovat, přidávat koeficienty, dokud se nenajde ten nejpřesnější možný model s pokud možno co nejnižším počtem parametrů při zachování vysoké relativní přesnosti. V této kapitole byly využity zdroje [1], [5], [6].

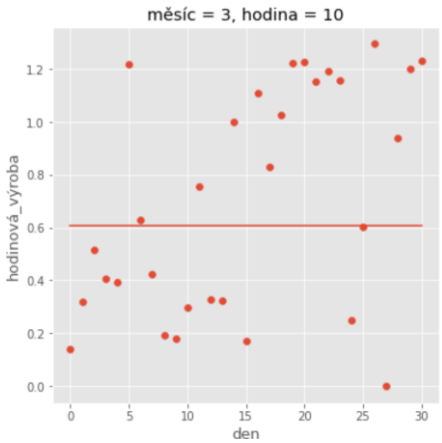
### 2.3.1 *Constant_D1_linear*

Nejdříve jsme vytvořili tzv. konstantní model, který nazveme *Constant_D1_linear*, ve kterém vůbec nepracujeme s počasím, zde jde o následující: pro měsíc m, den d a hodinu h chceme zjistit předpověď výroby elektřiny $v^{\mathrm{d}}_{\mathrm{m,h}}$. Náš první model můžeme vyjádřit jako

$$v^{\mathrm{d}}_{\mathrm{m,h}} = b(\mathrm{m}, \mathrm{h}),$$

kde koeficienty $b$ zjišťujeme za pomoci bayesovské inference. Značení $b(\mathrm{m}, \mathrm{h})$ znamená, že danému měsíci m a hodině h odpovídá jedna daná hodnota výroby elektřiny. Toto značení budeme využívat i v rámci popisu dalších modelů.

Predikce by se pro vybraný vzorek dat dala vizualizovat níže uvedeným obrázkem. Body jsou hodnoty hodinové výroby v testovací množině pro 3. měsíc, tedy duben (měsíce jsou ve sloupci indexovány od nuly) a 10–11. hodinu, přímka znázorňuje předpověď výroby pro tento hodinový interval v každém dni daného měsíce. Data vyobrazena v grafu jsou standardizována.




Obrázek 2.7: Znázornění předpovědi výroby elektřiny dle modelu *Constant_D1_linear* pro každý den a 10–11. hodinový interval v měsíci březnu. Data jsou standardizována.

Na následujícím obrázku uvidíme kód k prvnímu predikčnímu modelu

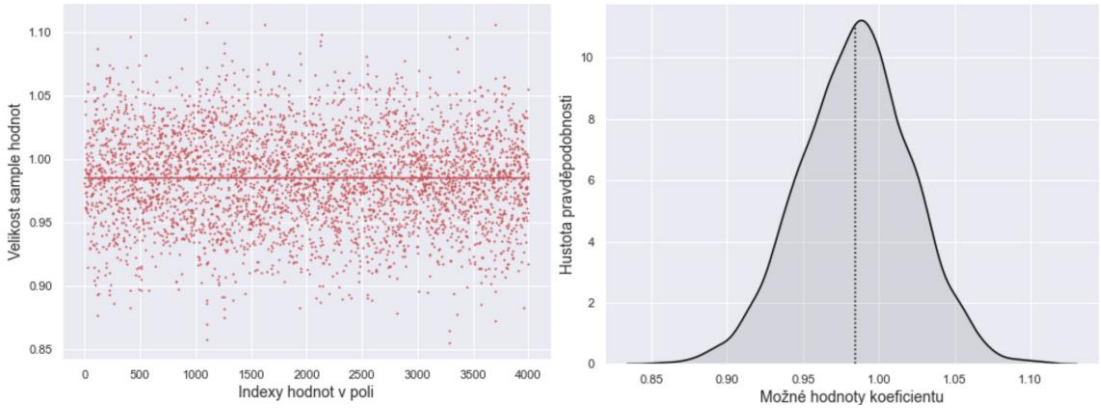
```
with pm.Model() as Constant_D1_linear:

    mesice = pm.Data('mesice', train['měsíc'].values-1)
    hodiny = pm.Data('hodiny', train['hodina'].values-4)

    w0 = pm.Normal('w0', mu=train['hodinová_výroba'].mean(), sd=1,shape = [12,15]) # Definice parametru w0,který je maticí

    w0 = (w0[mesice,hodiny]) # Z matice udělám vektor


    sd = pm.HalfNormal('sd', sd=5,shape=1)                          # definice hyperparametru likelihoodu
    likelihood = pm.Logistic('likelihood',                            # definice likelihoodu
                             mu = w0,
                             s  = sd,
                             observed=train['hodinová_výroba'].values)     #  výstupy modelu
```

```
with Constant_D1_linear:
    trace_1 = pm.sample()     # definice výpočetního algoritmu
```

Zdrojový kód 1: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Constant_D1_linear* v *Pymc3*

V prvních dvou odstavených řádcích napíšeme, s jakými vstupními daty budeme pracovat. Ve třetím odstaveném řádku definujeme apriorní rozdělení matice $\mathbf{W}_0$, náš předpoklad je, že matice $\mathbf{W}_0$ (tedy jednotlivé predikce pro dané časové intervaly) má normální rozdělení pravděpodobnosti se středem $\mu$ v průměrné hodnotě standardizované hodinové výroby a se směrodatnou odchylkou rovnou jedné. Ve čtvrtém odstaveném řádku přetvoříme matici $\mathbf{W}_0$ na vektor $\boldsymbol{w}_0$, který se „přizpůsobí“ trénovacím datům v tabulce *train*. V pátém řádku definujeme apriorní rozdělení parametru $\sigma$, tento parametr udává hodnotu směrodatné odchylky věrohodnostní funkce. Apriornímu rozdělení tohoto parametru přiřazujeme polonormální rozdělení pravděpodobnosti. Poté definujeme věrohodnostní funkci, které dáme logistické rozdělení pravděpodobnosti. Předpokládáme tedy, že každý jednotlivý datový bod (výroba elektřiny) má logistické pravděpodobnostní rozdělení. Věrohodnostní funkci je dán vektor $\boldsymbol{w_0}$ jako argument pro $\mu$ a *sd* jako argument pro $s$. Nakonec definujeme typ výpočetního algoritmu, v tomto případě využijeme funkci knihovny *Pymc3* – `pm.sample()`, která získá i nejistoty koeficientů modelu, přesněji posteriorní hustotu pravděpodobnosti, a tím i celkovou nejistotu předpovědi pro danou hodinu, den a měsíc. Sample algoritmy k vybrání náhodných hodnot z posteriorního rozdělení využívají metody Monte Carlo.

Nyní už máme zjištěné posteriorní hodnoty každého parametru modelu, co to tedy znamená? Numerické metody nedovedou zjistit analytický zápis výsledného posteriorního rozdělení, ale získají nám jen výběr některých jeho hodnot. Obrázek 2.8 graficky znázorňuje jednotlivé hodnoty v tomto poli, kde úsečka ukazuje průměr tohoto pole. Obrázek 2.9 graficky znázorňuje posteriorní „funkci“ daného koeficientu, kde přerušovaná úsečka zobrazuje průměrnou hodnotu tohoto pole.




Obrázek 2.8-2.9: Grafické vyobrazení standardizovaných sample hodnot predikce v poli a vyobrazení hodnot v tomto poli jako posteriorní funkce. Jedná se o standardizovanou predikci výroby elektřiny pro měsíc červen mezi 9–10. hodinou.

Nyní už jen otestujeme model na testovacích datech. První odstavený řádek určuje, s jakými daty pracujeme. Druhým odstaveným řádkem definujeme, že na základě těchto dat a již zjištěné posteriorní funkce koeficientů chceme zjistit posteriorní rozdělení predikované výroby elektrické energie pro každý hodinový interval v testovací množině.

```
with Constant_D1_linear:
    pm.set_data({'mesice': test['měsíc'].values-1, 'hodiny': test['hodina'].values-4})
    post_pred_1 = pm.sample_posterior_predictive(trace_1)
```

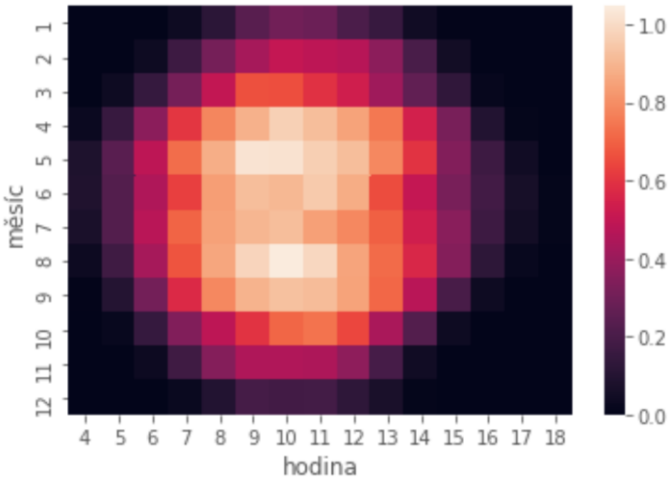
Zdrojový kód 2: Definice dat použitých pro otestování modelu *Constant_D1_linear* a definice výpočetního algoritmu parametrů tohoto modelu v *Pymc3*

Poté si vytvoříme pole, do kterého uložíme průměrné hodnoty předpovědí pro celou testovací množinu. Předpovězeným hodnotám, které jsou menší než 0, dáme hodnotu 0. Toto pole použijeme k ověření přesnosti modelu.

```
predikce_Constant_D1_linear                              = post_pred_1['likelihood'].mean(axis=0) * vyroba_delic
predikce_Constant_D1_linear[predikce_Constant_D1_linear<0]   = 0
```

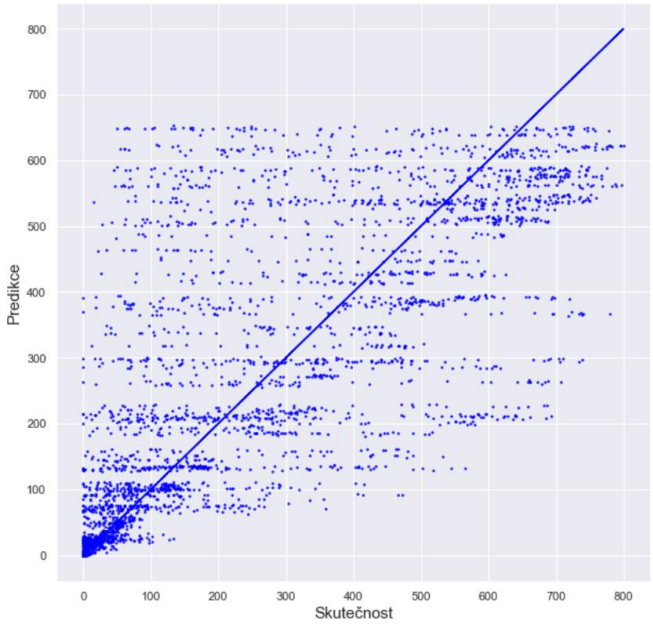
Zdrojový kód 3: Uložení absolutních hodnot předpovězené výroby elektřiny do pole predikce_Constant_D1_linear

Obrázek 2.10 graficky vyobrazuje velikost koeficientů matice $\mathbf{W}_0$. V této matici jsou obsaženy predikované standardizované hodnoty výroby elektřiny pro jednotlivé měsíce a hodiny. Z obrázku lze vyčíst velmi malé hodnoty výroby elektřiny v ranních a pozdních hodinách a vysoké hodnoty výroby mezi 9–12. hodinou. Na obrázku lze také vidět, že v zimních měsících je predikovaná výroba elektřiny velmi malá.




Obrázek 2.10: Grafické znázornění standardizovaných předpovědí hodinové výroby elektřiny pro daný měsíc a hodinu. Jde o hodnoty matice $\mathbf{W}_0$.

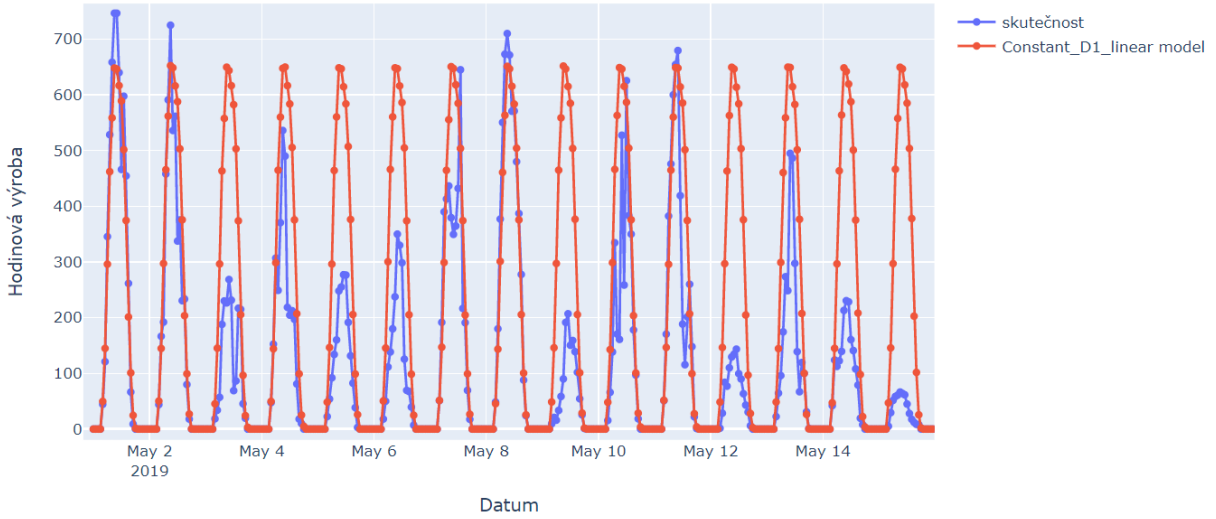
Nyní si tento model vyhodnotíme. Obrázek 2.11 znázorňuje grafické vyhodnocení našeho prvního modelu. Kdyby náš model nabýval 100% přesnosti, všechny jednotlivé body (předpovědi) by ležely na přímce. Čím je bod dál od přímky, tím má daná předpověď nižší přesnost. Je-li bod pod přímkou, číselná hodnota předpovědi je nižší než skutečná. Je-li bod nad přímkou, číselná hodnota předpovědi je vyšší. V rámci tohoto modelu můžeme vidět, že model je enormně nepřesný.



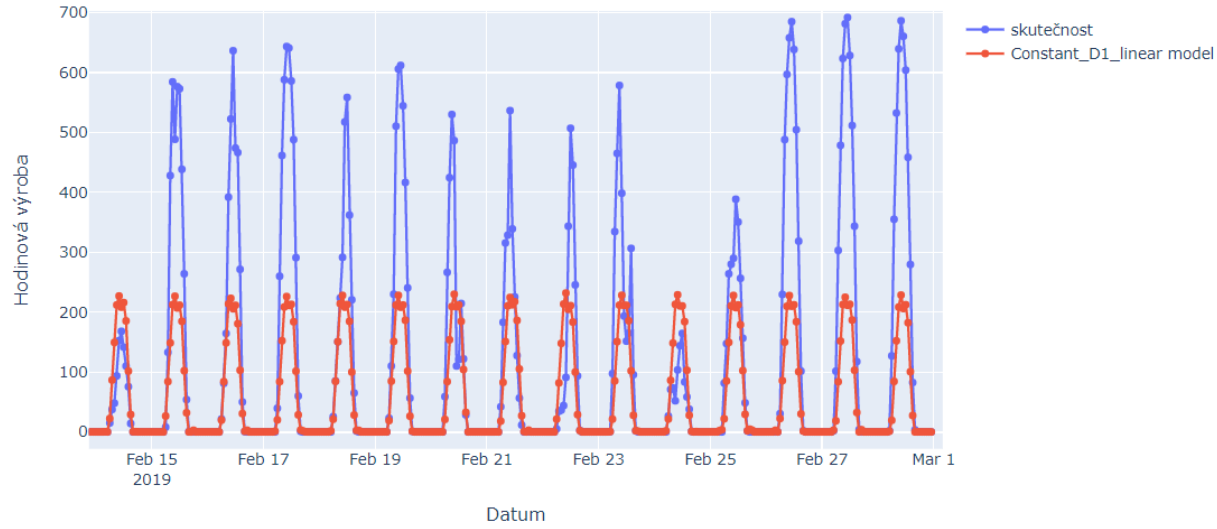

Obrázek 2.11: Grafické vyhodnocení modelu *Constant_D1_linear*

U každého jednotlivého modelu porovnáme jeho grafické vyhodnocení se skutečnou výrobou a jiným modelem. Pro porovnávání vybereme časové úseky první poloviny měsíce května (protože v tomto období si můžeme všimnout rozmanitých tvarů solárních křivek)

a druhé poloviny měsíce února (protože na tomto modelu lze rozumně ilustrovat rozdílnost modelů). Naši predikci srovnáváme se skutečnou výrobou elektřiny, která je součástí testovacích dat. Náš model zatím s žádným modelem neporovnáváme, protože se jedná o první zkoumaný model. Z obrázku je jasné, že tento model příliš neodpovídá skutečnosti, předpovězené solární křivky mají pro daný měsíc pořád stejný tvar.



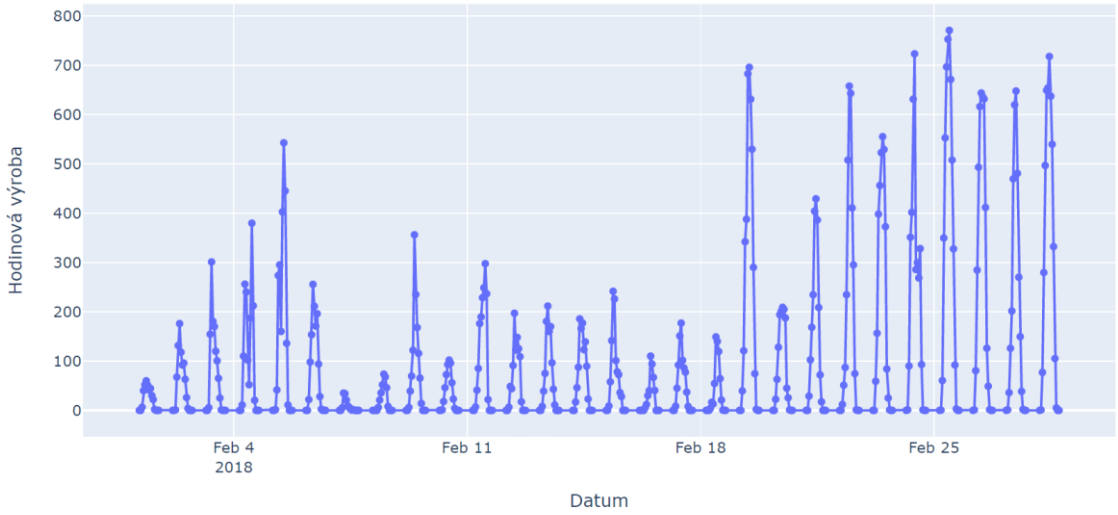

Obrázek 2.12: Srovnání predikcí modelu *Constant_D1_linear* se skutečnou výrobou elektřiny v první polovině měsíce května




Obrázek 2.13: Srovnání predikcí modelu *Constant_D1_linear* se skutečnou výrobou elektřiny v druhé polovině měsíce února

Obrázek 2.13 ukazuje, že predikovaná výroba elektřiny se velmi liší od té skutečné. Důvodem jsou data z trénovací množiny. Tato data znázorňuje Obrázek 2.14. Z obrázku lze

vyčíst, že výroba elektřiny ve většině dnů nabývala malých hodnot, proto jsme dostali tak malé hodnoty predikce.




Obrázek 2.14: Výroba elektřiny solární elektrárny v únoru roku 2018

Číselné vyhodnocení je následující

Tabulka 2.2: Číselné vyhodnocení modelu *Constant_D1_linear*

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Constant_D1_linear* | Testovací množina | 95,57 | 23015 | 49,41 |
| | 1. polovina května | 170,9 | 56722 | 111,22 |
| | 2. polovina února | 113,66 | 32920 | 47,05 |

Tyto hodnoty jsou pro nás zatím nicneříkající a budeme je porovnávat s vyhodnocením dalších modelů. Cílem je co nejvíce překonat tento model.

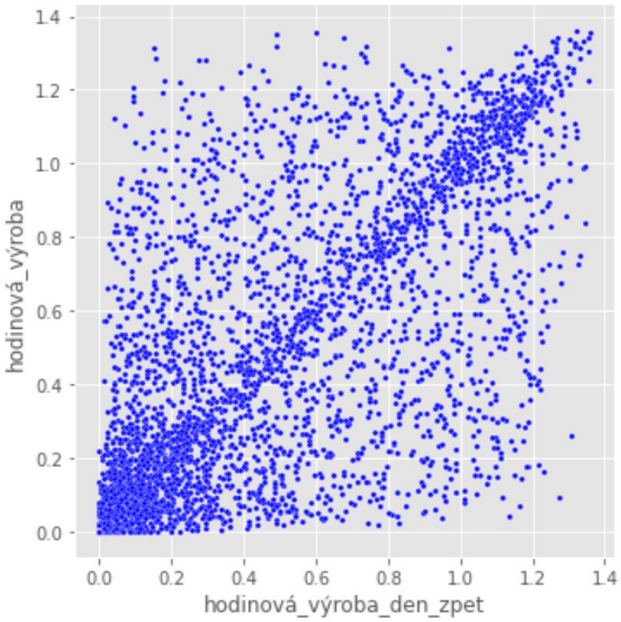
### 2.3.2 *Linear_D1_day_back*

Tento model predikuje výrobu elektřiny pro měsíc m, den d a hodinu h. Jedná se o vylepšenou verzi předchozího modelu. Liší se jen v jednom koeficientu $a$, který násobí výrobu elektřiny v měsíci m dni $\mathrm{d}-1$ a hodině h. Model lze matematicky vyjádřit následovně:

$$v_{\mathrm{m,h}}^{\mathrm{d}} = a v_{\mathrm{m,h}}^{\mathrm{d}-1} + b(\mathrm{m},\mathrm{h}),$$

kde $v_{\mathrm{m,h}}^{\mathrm{d}-1}$ je výroba elektřiny v měsíci m, hodině h a dni $\mathrm{d}-1$.

Než začneme s tvořením modelů strojového učení, je důležité znát nějaké závislosti vstupních veličin na těch výstupních, proto si vyobrazíme graf závislostí výroby elektřiny v hodině h a dni d na výrobě elektřiny v hodině h a dni $\mathrm{d}-1$. Z grafu můžeme vyčíst lineární závislosti s velkým šumem.



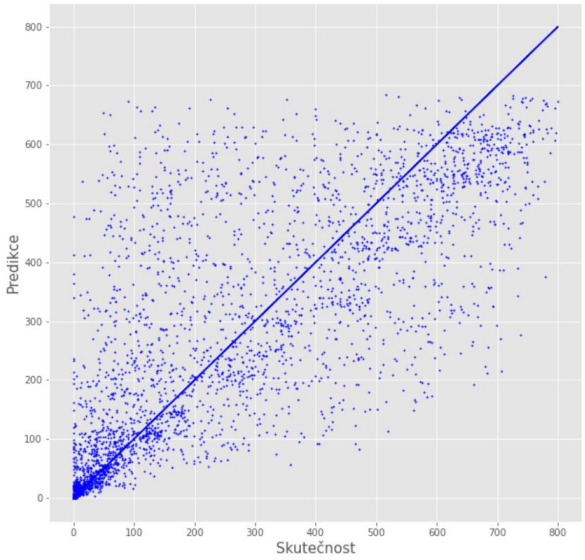

Obrázek 2.15: Závislost standardizované výroby elektřiny ve dni d a hodině h na výrobě elektřiny ve dni d − 1 a hodině h

Zde, i v rámci dalších modelů, ukážeme pouze část kódu, která je pro každý model jiná, zbylá část je pro všechny modely stejná.

```
with pm.Model() as Linear_D1_day_back:

    mesice        = pm.Data('mesice', train['měsíc'].values-1)
    hodiny        = pm.Data('hodiny', train['hodina'].values-4)
    data_den_zpet = pm.Data('data_den_zpet', train['hodinová_výroba_den_zpet'].values)

    w0 = pm.Normal('w0', mu=train['hodinová_výroba'].mean(), sd=1,shape = [12,15]) # Definice parametru w0,který je maticí
    a  = pm.Normal('a', mu = 0, sd = 1, shape = 1)                                 # Nový koeficient a


    predikce = (w0[mesice,hodiny]) + a*data_den_zpet


    sd = pm.HalfNormal('sd', sd=5,shape=1)                          # definice hyperparametru likelihoodu
    likelihood = pm.Logistic('likelihood',                            # definice likelihoodu
                          mu = predikce,
                          s  = sd,
                          observed=train['hodinová_výroba'].values)     #  výstupy modelu
```

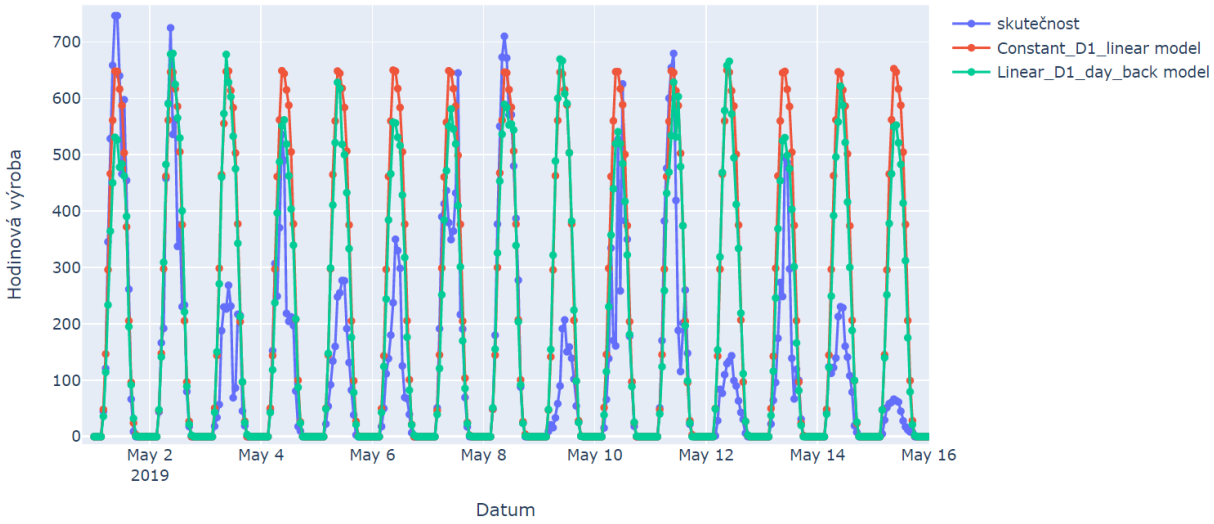
Zdrojový kód 4: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Linear_D1_day_back* v *Pymc3*

Na níže uvedeném obrázku je vyobrazeno grafické vyhodnocení modelu. Z obrázku je jasné, že tento vylepšený model nabývá vyšší přesnosti než model minulý.

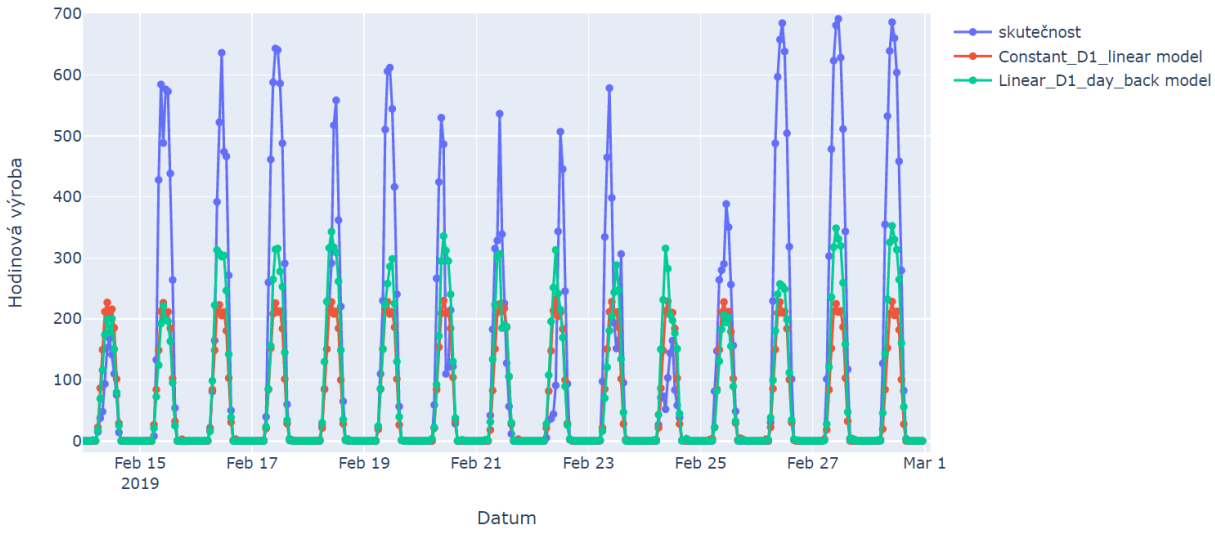



Obrázek 2.16: Grafické vyhodnocení modelu *Linear_D1_day_back*

Na obrázcích vidíme, že tento nový model je přesnější než předchozí model.




Obrázek 2.17: Srovnání predikce modelu *Linear_D1_day_back* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_day_back* v první polovině měsíce května




Obrázek 2.18: Srovnání predikce modelu *Linear_D1_day_back* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_day_back* v druhé polovině měsíce února

Tabulka 2.3: Srovnání chybových funkcí modelů *Constant_D1_linear* a *Linear_D1_day_back* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Constant_D1_linear* | Testovací množina | 95,57 | 23015 | 49,41 |
| | 1. polovina května | 170,9 | 56722 | 111,22 |
| | 2. polovina února | 113,66 | 32920 | 47,05 |
| *Linear_D1_day_back* | Testovací množina | 89,13 | 19914 | 46,74 |
| | 1. polovina května | 151,40 | 44619 | 111,91 |
| | 2. polovina února | 102,51 | 25176 | 49,34 |

### 2.3.3 *Linear_base_D1*

Další model už je mnohem zajímavější, protože budeme pracovat s počasím. Jde o následující: pro měsíc m, den d a hodinu h chceme opět zjistit předpověď výroby elektřiny $v^{\mathrm{d}}_{\mathrm{m,h}}$. Pro predikci ještě dále musíme znát již zmíněná data o počasí. Tento model můžeme vyjádřit jako

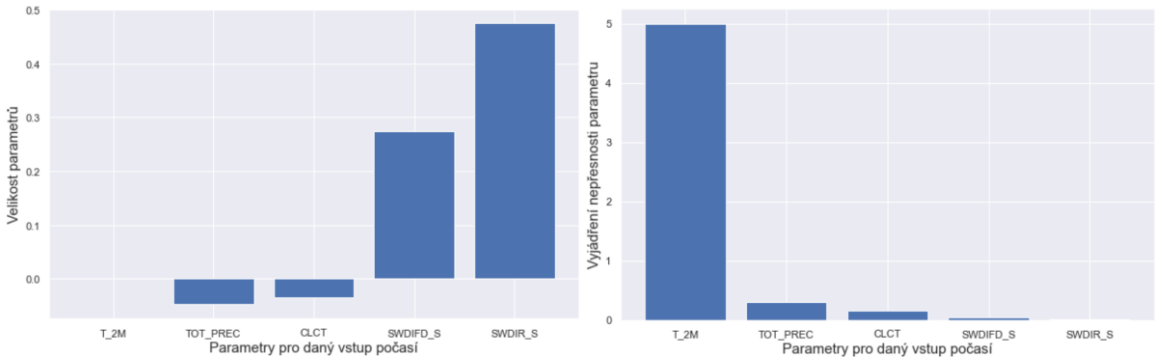
$$v^{\mathrm{d}}_{\mathrm{m,h}} = \boldsymbol{a} \cdot \mathbf{x}^{\mathrm{d}}_{\mathrm{m,h}} + b(\mathrm{m,h}),$$

kde $\mathbf{x}^{\mathrm{d}}_{\mathrm{m,h}}$ je vektor vstupních dat počasí, která náleží měsíci m, dni d a hodině h (v našem modelu se jedná o *CLCT*, *T_2M*, *SWDIFD_S*, *TOT_PREC*, *SWDIR_S*). $\boldsymbol{a}$ je vektor parametrů, který tato vstupní data násobí.

```
with pm.Model() as Linear_base_D1:

    mesice      = pm.Data('mesice', train['měsíc'].values-1)
    hodiny      = pm.Data('hodiny', train['hodina'].values-4)
    data_vstup  = pm.Data("data_vstup", train[vstupy_1].values)


    w0 = pm.Normal('w0', mu=train['hodinová_výroba'].mean(), sd=1,shape = [12,15])
    w  = pm.Normal('w', mu=0.25, sd=0.5,shape = 2)                              # shape je roven počtu údajů o počasí pro

    w0 = w0[mesice,hodiny]     # udělá vektor

                                                                                #  přizpůsobený datu v tabulce
                        # skalární součin
    rovnice = w0 + (data_vstup *w).sum(axis=1)


    sd = pm.HalfNormal('sd', sd=5,shape=1)
    likelihood = pm.Logistic('likelihood',
                             mu = rovnice,
                             s  = sd,
                             observed=train['hodinová_výroba'].values)
```

Zdrojový kód 5: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Linear_base_D1* v *Pymc3*

V rámci bayesovské regrese lze hodnotit i přesnost koeficientů. V této práci jsme zvolili přesnost koeficientů jako podíl směrodatné odchylky daného koeficientu s jeho průměrnou hodnotou. Čím je podíl těchto dvou čísel vyšší, tím je parametr méně přesný. V tomto modelu je nejméně přesný koeficient násobící data ze sloupce *T_2M* (teplota). Nejpřesnější je parametr, který násobí data ve sloupci *SWDIR_S* (přímý sluneční svit).

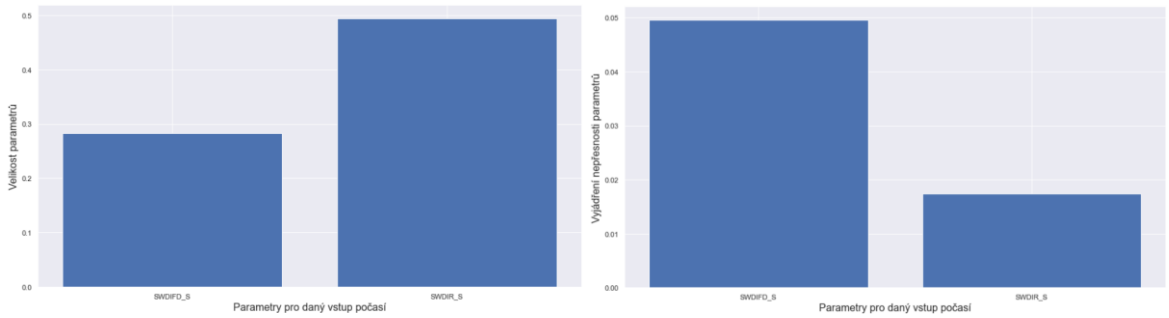



Obrázek 2.19-2.20: Grafické znázornění velikostí parametrů modelu a jejich nejistot ve formě podílu směrodatné odchylky s průměrem, zde pracujeme se všemi pěti parametry, které náleží vstupním hodnotám počasí.

Vzhledem k tomu, že jsou všechna vstupní data standardizována, v drtivé většině případů nabývají hodnot mezi 0–1 (vstupy *T_2M* jsou standardizovány tak, že jeho hodnoty leží ve většině případů mezi -1–1), můžeme důležitost parametrů hodnotit podle jejich velikosti. Velikostí parametru myslíme průměr z pole jeho sample hodnot. Když srovnáme předchozí dva grafy, tak zjistíme, že čím je daný koeficient přesněji nafitován, tím je důležitější pro naše predikce.

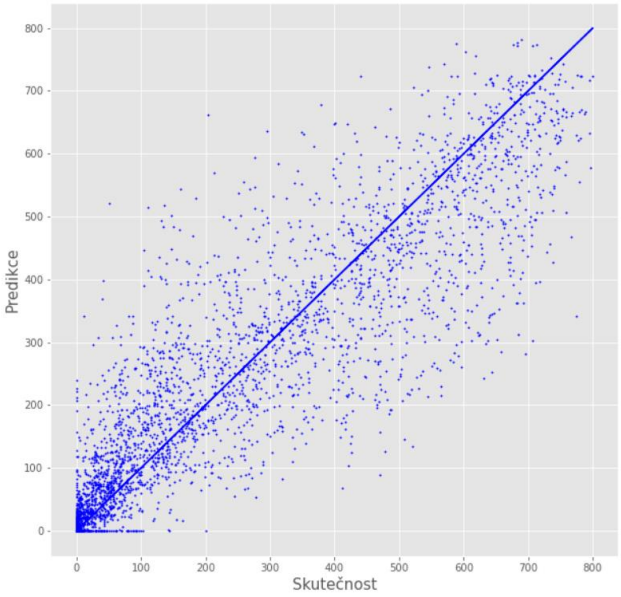
Z důvodu velké závislosti modelu na datech *SWDIR_S* a *SWDIFD_S* si zkusme vytvořit model jen se dvěma zmíněnými vstupy. Ukáže se, že tento nový model je stejný jako předchozí, data o teplotě (*T_2M*), oblačnosti (*CLCT*) a souhrnném úhrnu srážek (*TOT_PREC*) jsou

nepotřebná, proto pro vyhodnocení použijeme tento méně parametrický model, a to nejen teď, ale i v dalších částech této práce.




Obrázek 2.21–2.22: Grafické znázornění velikostí parametrů modelu a jejich nejistot ve formě podílu směrodatné odchylky s průměrem, zde pracujeme jen se dvěma nejdůležitějšími parametry, které násobí vstupní hodnoty *SWDIR_S* a *SWDIFD_S*.

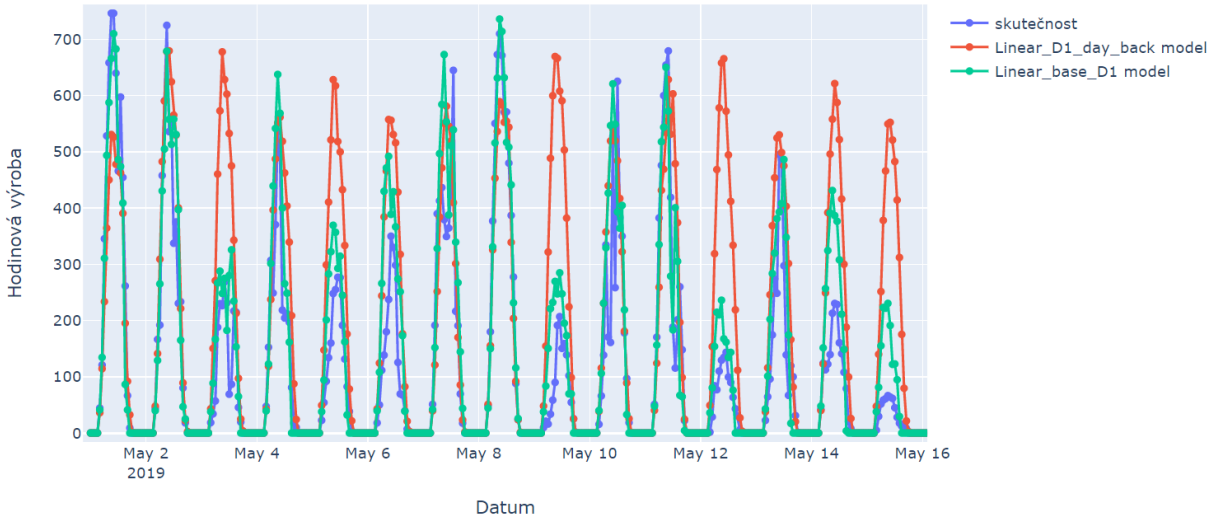
Zde v celkovém grafickém vyhodnocení modelu vidíme větší přiléhavost k přímce než v případě předchozího modelu, tento model můžeme pokládat za přesnější. Pro jistotu si tento fakt dále ověříme i prostřednictví jiných metrik.



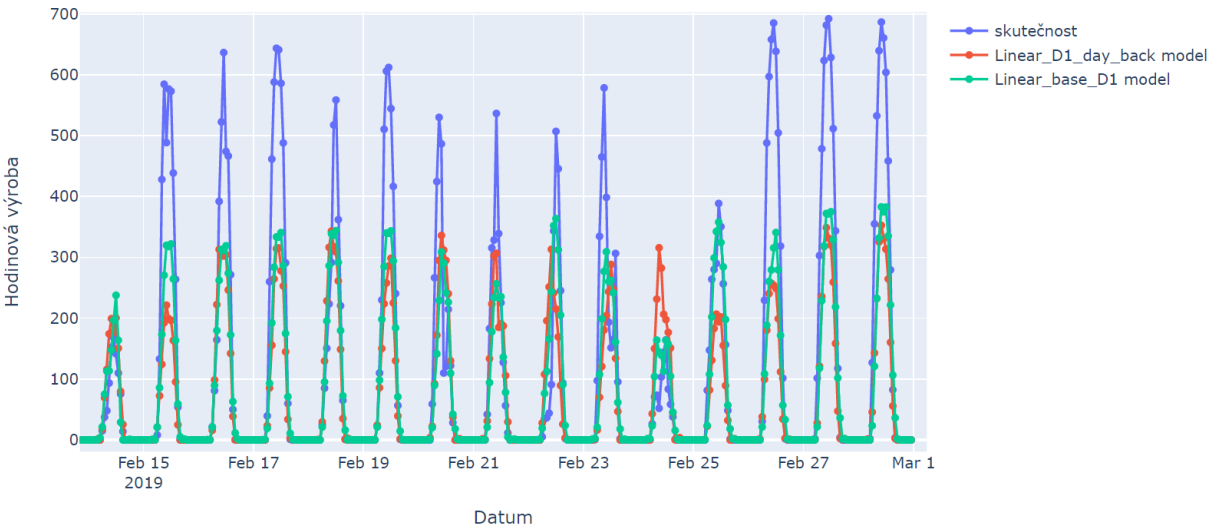

Obrázek 2.23: Grafické vyhodnocení modelu *Linear_base_D1*

V rámci dalších modelů nebudeme tohle grafické vyhodnocení přesnosti modelů příliš znázorňovat, protože je grafické vyhodnocení dalších modelů téměř totožné s tímto obrázkem.

Níže uvedené grafy vykreslují výrazné zlepšení přesnosti modelu pro oba zkoumané časové intervaly. Počasí je tedy pro naše denní modelování velmi důležité a budeme s ním dále pracovat. Otázkou je, jak ho nejlépe využít k co nejpřesnějšímu modelu.




Obrázek 2.24: Srovnání predikce modelu *Linear_base_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_D1_day_back* v první polovině měsíce května




Obrázek 2.25: Srovnání predikce modelu *Linear_base_D1* se skutečnou výrobou elektřiny a predikcí modelu L*inear_D1_day_back* v druhé polovině měsíce února

Tabulka 2.4 ukazuje, že tento model byl ve všech zkoumaných aspektech výrazně lepší.

Tabulka 2.4: Srovnání chybových funkcí modelů *Constant_D1_linear* a *Linear_D1_day_back* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Linear_D1_day_back* | Testovací množina | 89,13 | 19914 | 46,74 |
| | 1. polovina května | 151,40 | 44619 | 111,91 |
| | 2. polovina února | 102,51 | 25176 | 49,34 |
| *Linear_base_D1* | Testovací množina | 65,85 | 10154 | 36,23 |
| | 1. polovina května | 74,4 | 10844 | 47,84 |
| | 2. polovina února | 82,04 | 17531 | 29,57 |

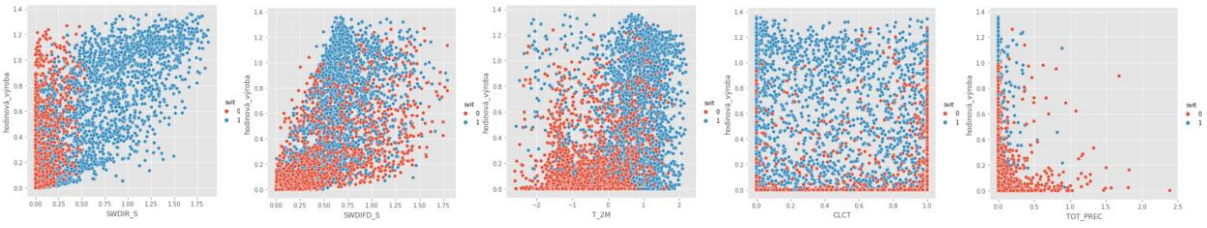
### 2.3.4 *Linear_shine_D1*

Další model pro měsíc m, den d a hodinu h spočítá předpověď výroby elektřiny $v_{\mathrm{m,h}}^{\mathrm{d}}$. Pro predikci ještě dále musíme znát i hodnotu s ve sloupci *svit*, která nabývá pouze hodnot 0 či 1, a také již zmíněná data o počasí. Začneme s modelem, který bere v úvahu všech 5 vstupů o počasí. Tento model můžeme vyjádřit jako

$$v_{\mathrm{m,h}}^{\mathrm{d}} = \boldsymbol{a}(\mathrm{s}) \cdot \mathbf{x}_{\mathrm{m,h}}^{\mathrm{d}} + b(\mathrm{h}, \mathrm{s}),$$

kde $\mathbf{x}_{\mathrm{m,h}}^{\mathrm{d}}$ je vstupní vektor počasí, který je jiný pro den d, měsíc m a hodinu h, $\boldsymbol{a}(\mathrm{s})$ je vektor koeficientů, který je jiný pro různou hodnotu s ve sloupci *svit*.

Důležitost dělení dat podle sloupce *svit* ukazují Obrázky 2.26–2.30. Lze si povšimnout, že takto rozdělená vstupní data mají zcela jiný průběh. V případě prvních tří grafů zleva vidíme lineárnější závislost pro hodnoty veličiny *svit*, které jsou rovny 1. Pokud se bavíme o veličině *CLCT*, už nevidíme žádnou lineární závislost, ale rozprostření hodnot, kdy hodnota veličiny *svit* je rovna 1, po celém grafu, u hodnot, kde je svit roven 0 nabývá většina hodnot *CLCT* hodnoty 0 či 100. Graf pro *TOT_PREC* ukazuje, že nejvyšších hodnot výroba dosahuje, když je hodnota *TOT_PREC* nulová (tedy když neprší) a zároveň tyto hodnoty samozřejmě odpovídají pro datové body, kde je hodnota svit rovna jedné.

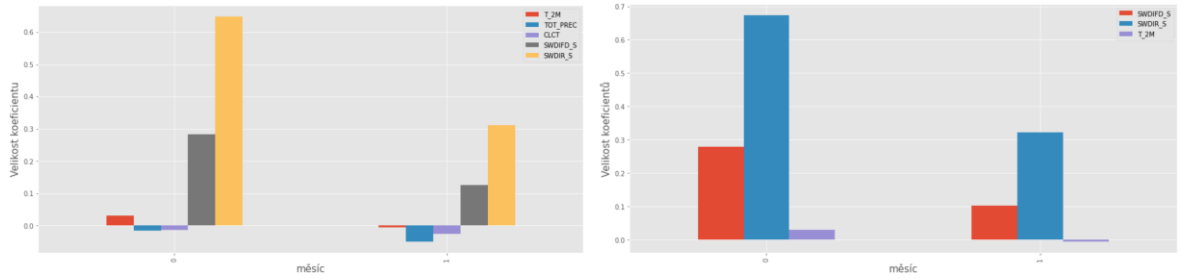


Obrázek 2.26–2.30: Závislost vstupních dat počasí z tabulky *train* na výstupních datech s ohledem na dělení podle binárních hodnot ve sloupci *svit*. Data jsou standardizována.

```
with pm.Model() as Linear_shine_D1:

    svity       = pm.Data('svity', train['svit'].values)
    hodiny      = pm.Data('hodiny', train['hodina'].values-4)
    data_vstup  = pm.Data("data_vstup", train[vstupy_2].values)

    w0 = pm.Normal('w0', mu=train['hodinová_výroba'].mean(), sd=1,shape = [15,2])
    w  = pm.Normal('w', mu=0.25, sd=0.5,shape = [3,2])

    w0 = w0[hodiny,svity]
    w  = w[:,svity].transpose()                    # Vybíráme, zda-li vektor dat skalárně vynásobit 1. nebo 2.sloupec v matici
                                                     # řádkem matice w na základě hodnoty svitu, tedy 0 nebo 1
    rovnice = w0 + (data_vstup *w).sum(axis=1)


    sd = pm.HalfNormal('sd', sd=5,shape=1)
    likelihood = pm.Logistic('likelihood',
                             mu = rovnice,
                             s  = sd,
                             observed=train['hodinová_výroba'].values)
```

Zdrojový kód 6: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Linear_shine_D1* v *Pymc3*

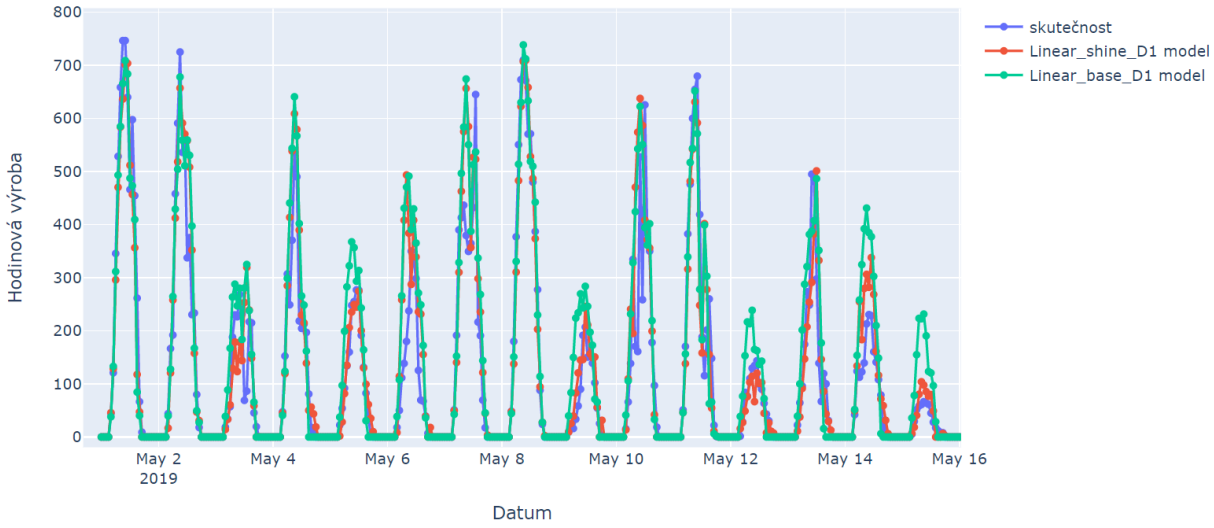
Jak jsme zmínili výše, nejdříve jsme vytvořili model pro všech 5 hodnot počasí k dané hodině. Poté jsme vytvořili model jen s daty *SWDIR_S* a *SWDIFD_S*, který měl nižší přesnost. Pokračovali jsme tedy tím, že jsme do modelu k těmto dvěma nejdůležitějším vstupům přidali vždy jeden nový vstup. Zjistili jsme, že vstup pro teplotu není nadbytečný, protože model s těmito třemi vstupy (*SWDIR_S*, *SWDIFD_S* a *T_2M*) nabýval stejné přesnosti jako model se všemi pěti vstupy. Pracujeme tedy se třemi vstupy, a to nejen v rámci tohoto modelu ale i těch složitějších.



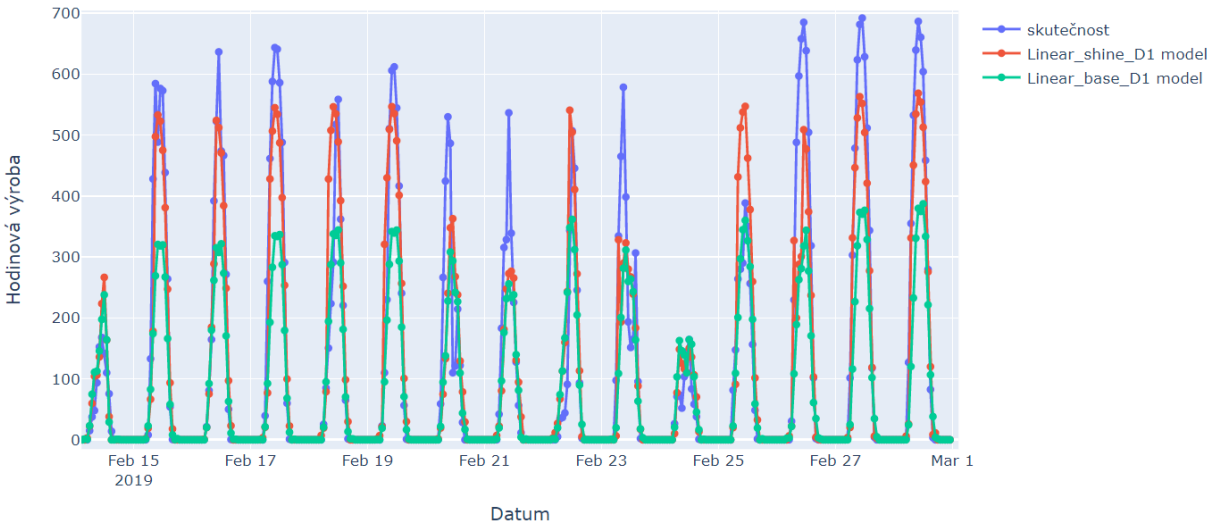

Obrázek 2.31–2.32: Grafické znázornění velikostí hodnot všech parametrů modelu, kde model jako vstup bere v úvahu všech pět vstupů počasí a znázornění modelu stejné struktury, kde bere v úvahu jen tři důležité vstupy.

Obrázek 2.31 vyobrazuje velikosti parametrů modelu. Tyto parametry násobí i nepotřebné vstupy. Obrázek 2.32 znázorňuje velikosti jen potřebných parametrů.

Na obrázcích vidíme, že nový, lehce složitější, model je přesnější než ten předchozí. V měsíci květnu se předpovědi zlepšily hlavně ve dnech, kdy se elektřiny moc nevyrobilo, únorové předpovědi se razantně zlepšily obzvlášť v hodinách, kdy se vyrobilo hodně elektřiny.




Obrázek 2.33: Srovnání predikce modelu *Linear_shine_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_base_D1* v první polovině měsíce května




Obrázek 2.34: Srovnání predikce modelu *Linear_shine_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_base_D1* v druhé polovině měsíce února

Tabulka 2.5: Srovnání chybových funkcí modelů *Linear_shine_D1* a *Linear_base_D1* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Linear_base_D1* | Testovací množina | 65,85 | 10154 | 36,23 |
| | 1. polovina května | 74,40 | 10844 | 47,84 |
| | 2. polovina února | 82,04 | 17531 | 29,57 |
| *Linear_shine_D1* | Testovací množina | 58,76 | 8823 | 31,69 |
| | 1. polovina května | 57,34 | 8099 | 33,23 |
| | 2. polovina února | 58,23 | 9234 | 28,34 |

Překonali jsme předchozí model, kde jsme koeficienty nedělili podle hodnoty ve sloupci *svit*. Ukázalo se, že má smysl zkusit složitější model.

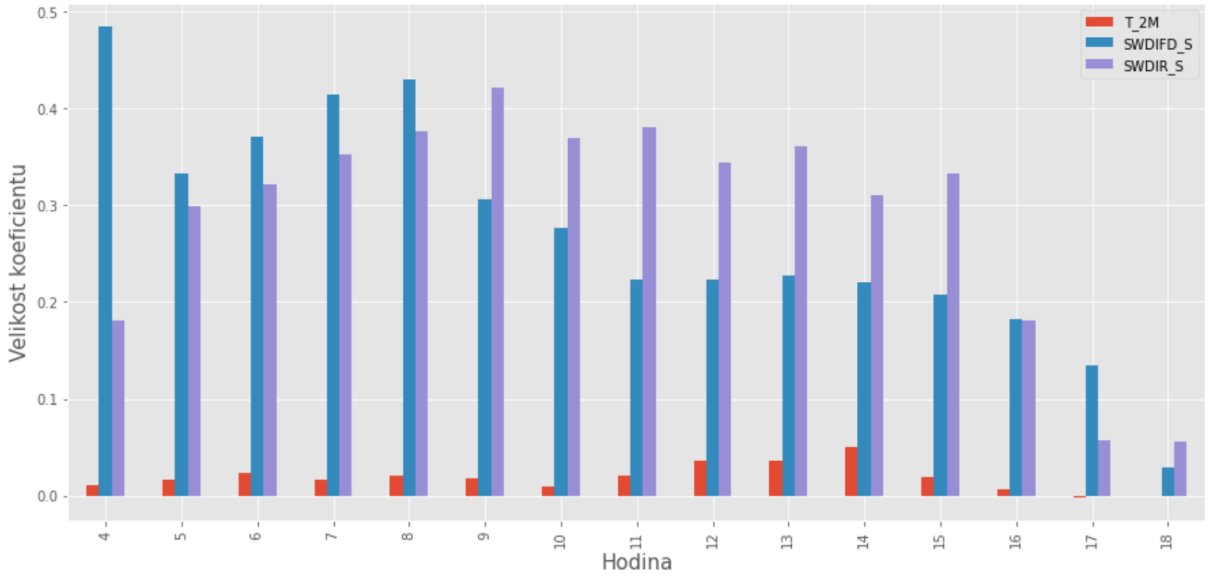
### 2.3.5 *Linear_hours_D1*

Nyní vytvoříme model podobný předchozímu modelu, vyjádříme ho rovnicí

$$v^{\mathrm{d}}_{\mathrm{m,h}} = \boldsymbol{a}(\mathrm{h}) \cdot \mathbf{x}^{\mathrm{d}}_{\mathrm{m,h}} + b(\mathrm{h}, \mathrm{s}).$$

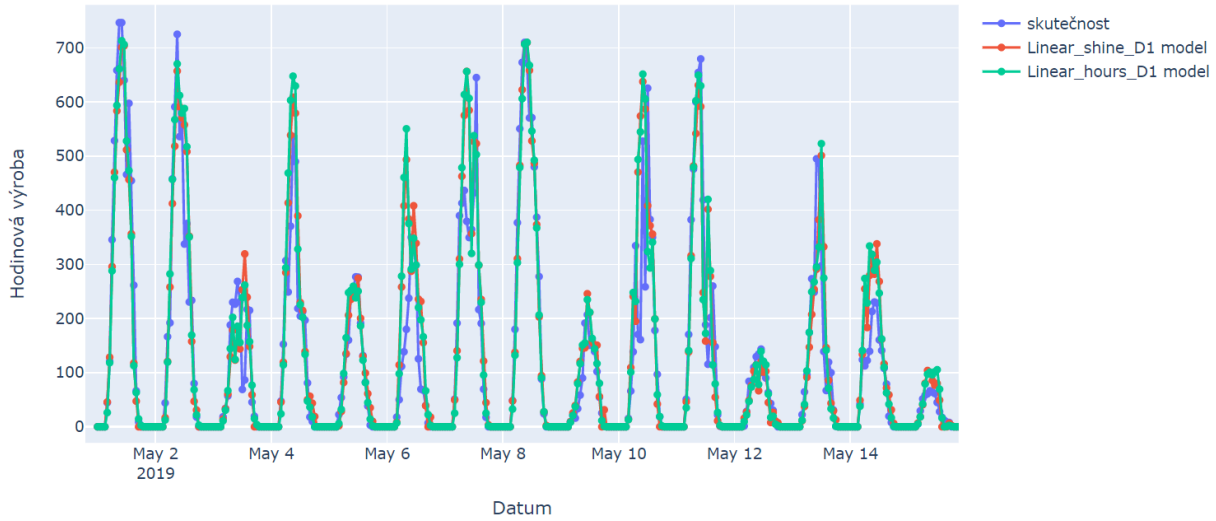
Zde je vektor koeficientů $\boldsymbol{a}(\mathrm{h})$ jiný pro různé hodinové intervaly. Kód je téměř identický jako v předchozím případě, proto ho zde neuvedu.

Obrázek 2.35 vyobrazuje velikost koeficientů lineárního modelu *Linear_hours_D1*. Lze si povšimnout, že v rámci tohoto modelu je pro brzké ranní hodiny výroba elektrické energie nejvíce závislá na difuzním slunečním světle *SWDIFD_S* . V pozdějších hodinách je výroba elektřiny nejvíce závislá na přímém slunečním světle *SWDIR_S*. Výroba pro každou hodinu závisí na teplotě jen zanedbatelně.

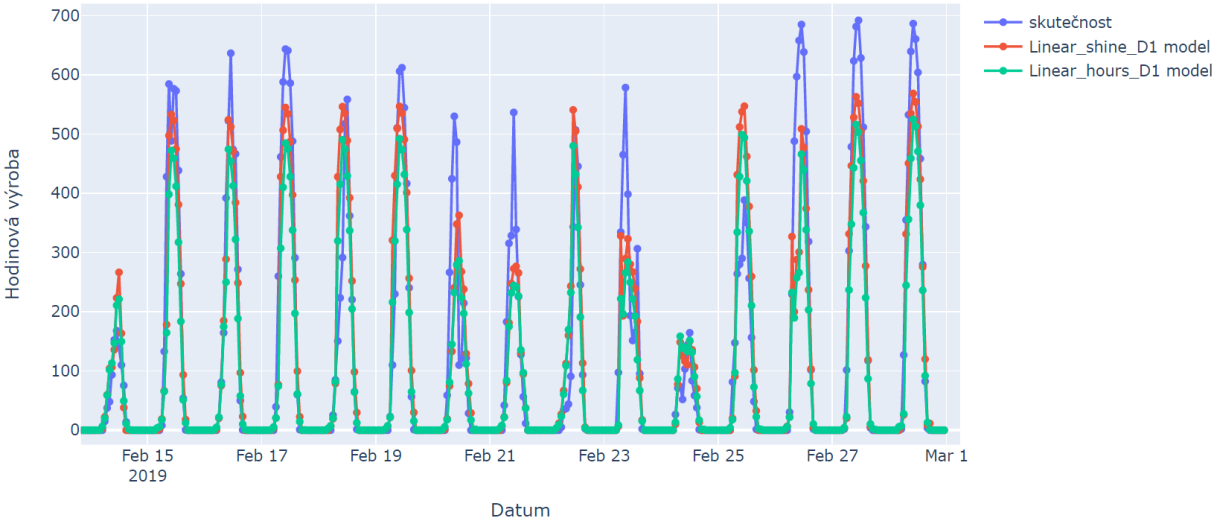



Obrázek 2.35: Grafické zobrazení parametrů modelu *Linear_hours_D1*, ve kterém se berou v úvahu jen potřebná data o počasí, tj. *SWDIR_S*, *SWDIFD_S* a *T_2M*.

Srovnávané modely nabývají velmi podobných hodnot chybových metrik. Tento nový model je jen těsně méně přesný než model *Linear_shine_D1*. Nižší nepřesnosti si lze všimnout na grafickém vyobrazení solárních křivek v druhé polovině měsíce února.




Obrázek 2.36: Srovnání predikce modelu *Linear_hours_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_shine_D1* v první polovině měsíce května




Obrázek 2.37: Srovnání predikce modelu *Linear_hours_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_shine_D1* v druhé polovině měsíce února

Tabulka 2.6: Srovnání chybových funkcí modelů *Linear_hours_D1* a *Linear_shine_D1* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Linear_shine_D1* | Testovací množina | 58,76 | 8823 | 31,69 |
| | 1. polovina května | 57,34 | 8099 | 33,23 |
| | 2. polovina února | 58,23 | 9234 | 28,34 |
| *Linear_hours_D1* | Testovací množina | 59,43 | 9334 | 30,68 |
| | 1. polovina května | 58,11 | 8997 | 27,84 |
| | 2. polovina února | 65,55 | 10636 | 25,35 |

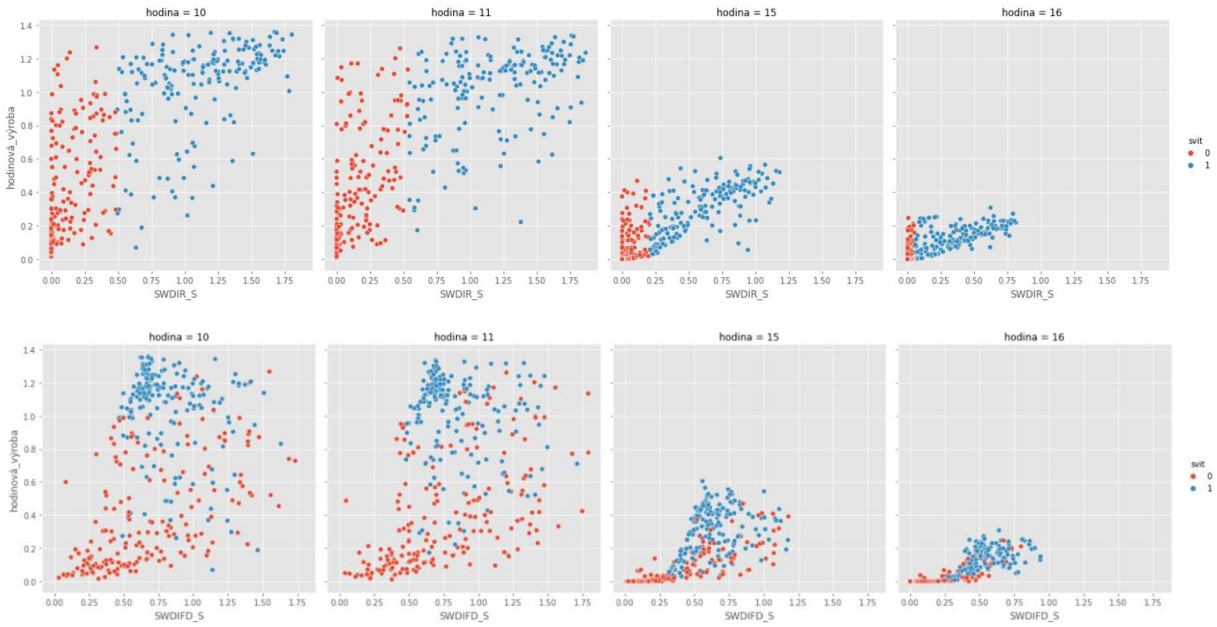
Tento model má nižší přesnost než ten předchozí, ale není výrazně nepřesný. Zkusíme si vytvořit ještě složitější model.

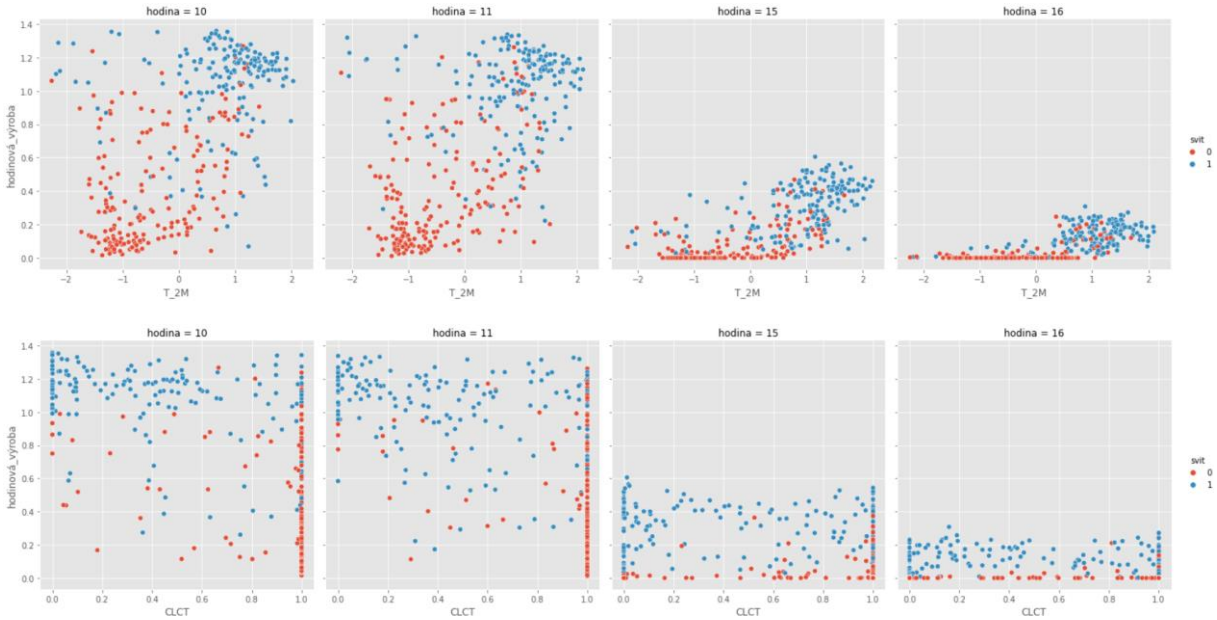
### 2.3.6 *Linear_complex_D1*

Nakonec si ukážeme výsledný model lineární regrese, kde jsou vektory koeficientů $\boldsymbol{a}(\mathrm{h}, \mathrm{s})$ odlišné pro různé hodinové intervaly h a *svit* s, pro tento model platí rovnice

$$v^{\mathrm{d}}_{\mathrm{m,h}} = \boldsymbol{a}(\mathrm{h}, \mathrm{s}) \cdot \mathbf{x}^{\mathrm{d}}_{\mathrm{m,h}} + b(\mathrm{h}, \mathrm{s}).$$

Níže uvedené obrázky znázorňují závislosti hodinové výroby na jednotlivých vstupních datech (kromě *TOT_PREC*, protože tento vstup nevykazuje žádnou závislost) v různých hodinových intervalech, konkrétně mezi 10–12. a 15–17. hodinou. Můžeme si všimnout, že dělení dat podle hodnot ve sloupci *svit* oddělí tyto závislosti do dvou zcela rozdílných skupin. Pro *svit* rovný 1 má hodinová výroba velmi dobrou lineární závislost na jednotlivých vstupech, pro *svit* roven 0 už moc ne. Největší zajímavostí, kterou lze z obrázků vyčíst, je, že pro hodinové intervaly mezi 10–11. a 11–12. hodinou, má výroba elektřiny na teplotě negativní závislost pro hodnoty veličiny *svit* rovny 1. Čím je tedy vyšší teplota v hodinách, kdy nabývá veličina *SWDIR_S* velkých hodnot, tím méně elektřiny elektrárna vyrobí. Pro hodinové intervaly mezi 15–16. a 16–17 už je závislost pozitivně lineární a to zvláště pro *svit* roven 1.





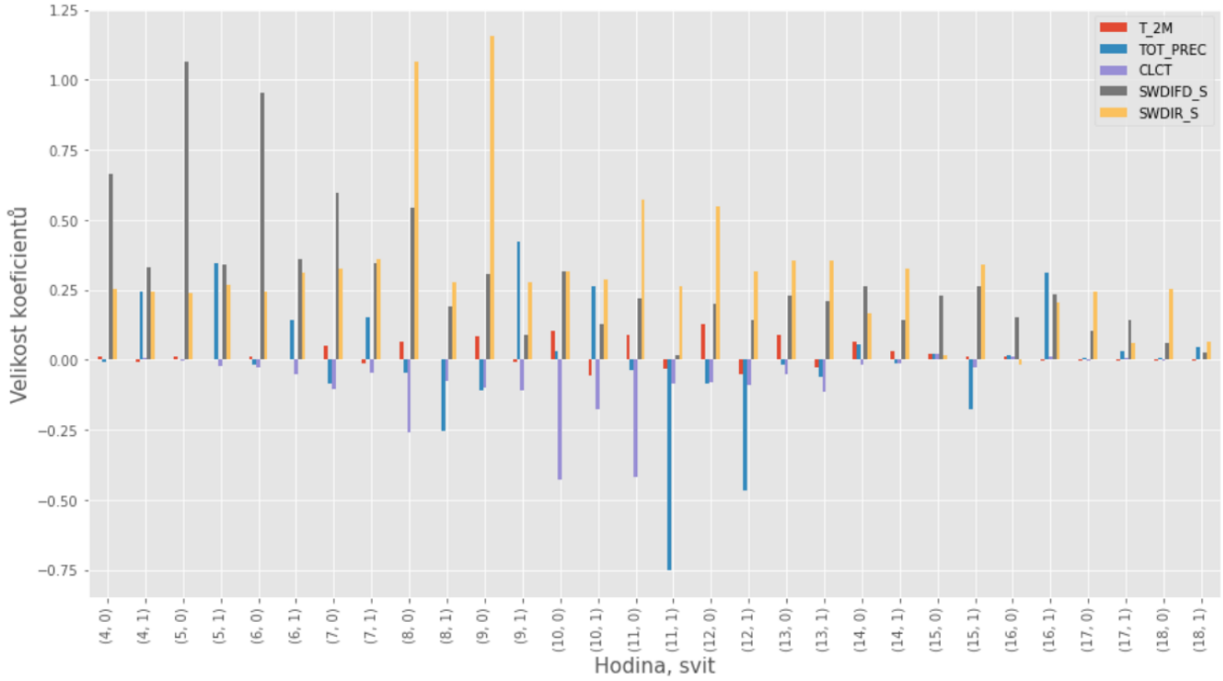

Obrázek 2.38–2.53: Závislosti standardizované hodinové výroby elektřiny v hodinových intervalech mezi 10–11., 11–12., 15–16. a 16–17. hodinou na standardizovaných vstupních datech počasí *SWDIR_S*, *SWDIFD_S*, *T2_M* a *CLCT* s ohledem na dělení dat podle binárních hodnot ve sloupci *svit*

```
with pm.Model() as Linear_complex_D1:

    data_vstup   = pm.Data("data_vstup",  train[vstupy_2].values)
    hodiny       = pm.Data("hodiny", train['hodina'].values-4)
    svity        = pm.Data("svity",   train['svit'].values)

    w0 = pm.Normal('w0', mu = train['hodinová_výroba'].mean(), sd=1, shape = [15,2])
    w  = pm.Normal('w',  mu = 0.25, sd=0.5, shape = [3,15,2])

    w0 = w0[hodiny,svity]
    w  = w[:,hodiny,svity].transpose()


    rovnice = w0 + (data_vstup *w).sum(axis=1)


    sd = pm.HalfNormal('sd', sd = 5,shape = 1)
    likelihood = pm.Logistic('likelihood',
                             mu = rovnice,
                              s = sd,
                             observed= train['hodinová_výroba'].values)
```

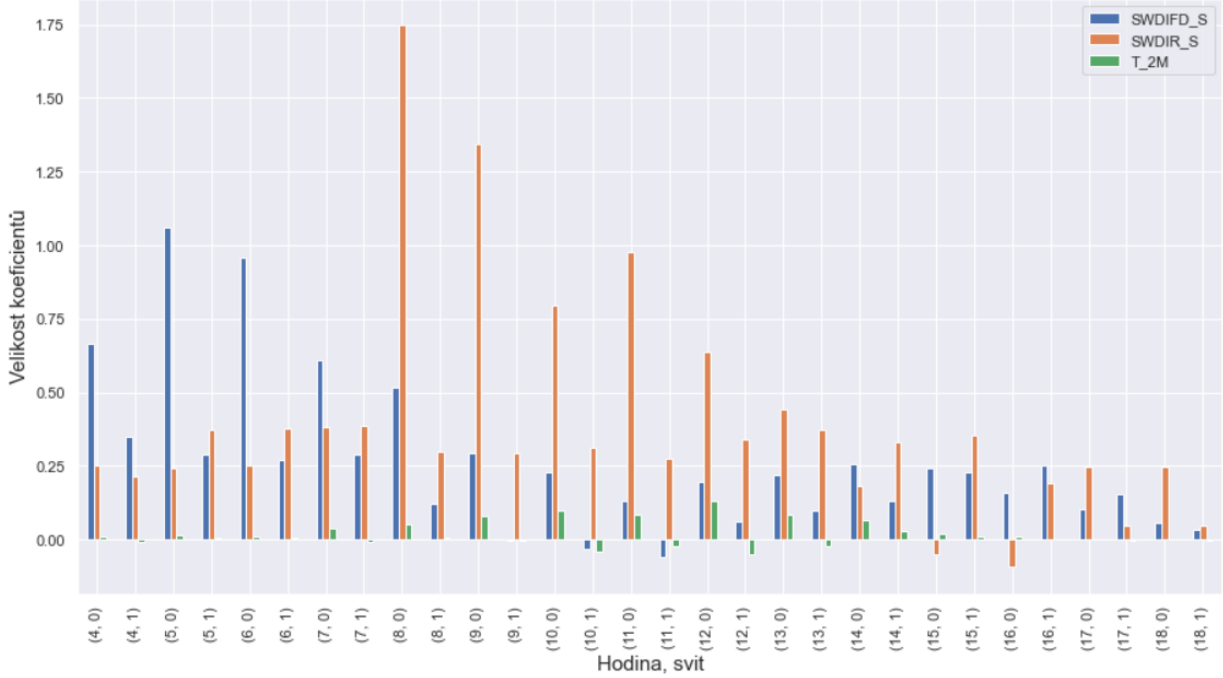
Zdrojový kód 7: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Linear_complex_D1* v *Pymc3*

Následující grafická zobrazení jsme opět vytvořili jak pro model, který bere v úvahu všechny vstupy o počasí, tak pro model, který bere v úvahu jen 3 vstupy. Můžeme všimnout, že pro brzké ranní hodiny jsou koeficienty pro *SWDIFD_S* vyšší než pro *SWDIR_S*. Pro odpolední hodiny je to naopak. Obrázek 2.54 znázorňuje, že vstup pro celkový úhrn srážek je zcela nepodstatný vzhledem k tomu, že koeficienty pro tento údaj jsou někdy příliš vysoké a někdy příliš nízké. Závislost mezi *TOT_PREC* a hodinovou výrobou je nedostatečná, ani zde není ukázána. Na Obrázku 2.55 si můžeme všimnout, že mezi 10–13. hodinou pro svit roven 1 jsou koeficienty pro teplotu záporné, tedy čím vyšší je teplota, tím nižší výroba. Tento fakt se

shoduje s fyzikálním pohledem, který říká, že při velké hodnotě přímého osvitu (což v těchto hodinách pro *svit* s hodnotou 1 je) má teplota na výrobu negativní vliv. Bohužel došlo k mírnému overfittingu, protože ve 4 případech z 60 byly hodnoty koeficientů pro osvit menší než nula.

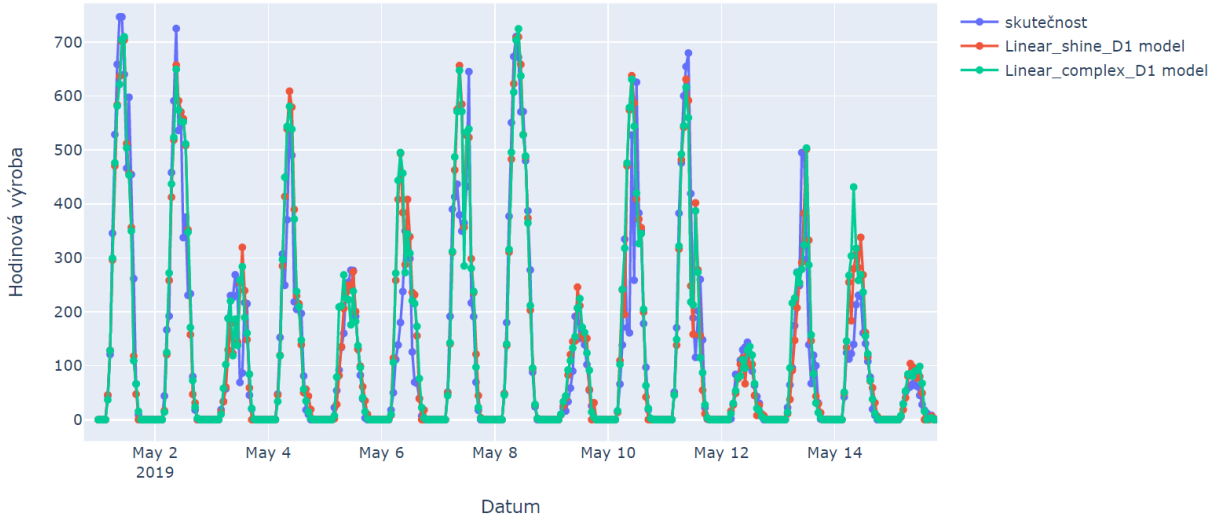



Obrázek 2.54: Grafické zobrazení parametrů modelu *Linear_complex_D1*, ve kterém se berou v úvahu i nepotřebná data o počasí, tj. *TOT_PREC* a *CLCT*.

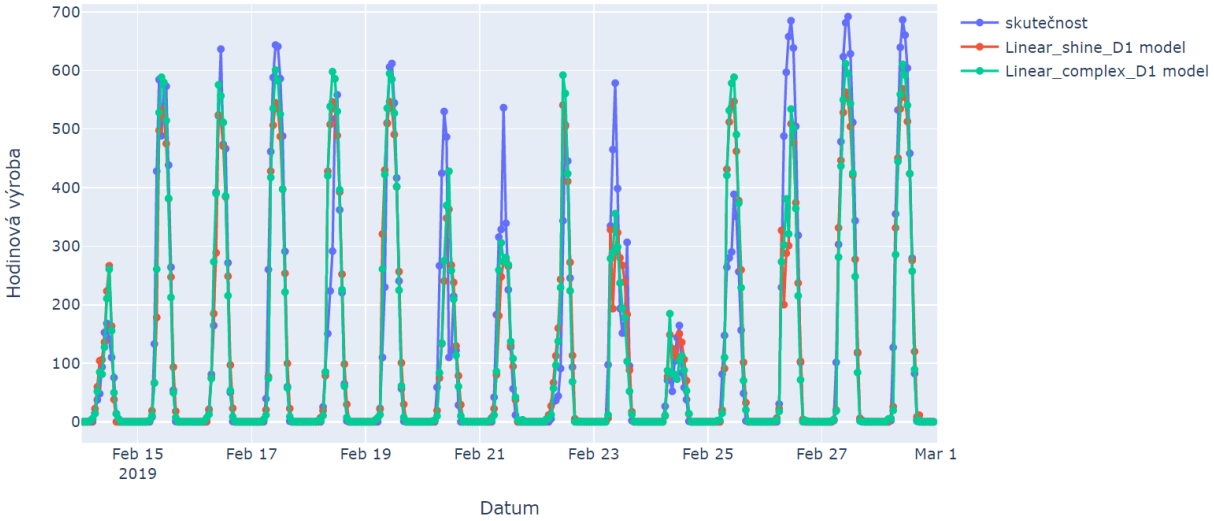



Obrázek 2.55: Grafické zobrazení parametrů modelu *Linear_complex_D1*, ve kterém se berou v úvahu jen potřebná data o počasí, tj. *SWDIR_S*, *SWDIFD_S* a *T_2M*.

Pokud si vyobrazíme historické hodnoty výroby, zjistíme, že v měsících, kdy hodně svítí, je nový model z pohledu přesnosti pouhým okem těžko porovnatelný s modelem *Linear_shine_D1*. Pokud se podíváme na zimní měsíce, uvidíme, že je nový model o něco přesnější.




Obrázek 2.56: Srovnání predikce modelu *Linear_complex_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_shine_D1* v první polovině měsíce května



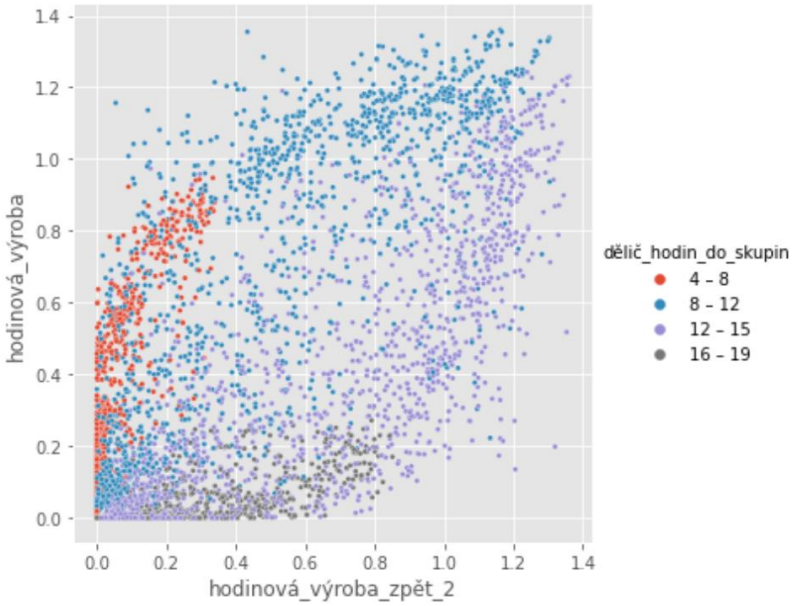

Obrázek 2.57: Srovnání predikce modelu *Linear_complex_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_shine_D1* v druhé polovině měsíce února

Tabulka 2.7: Srovnání chybových funkcí modelů *Linear_complex_D1* a *Linear_shine_D1* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Linear_shine_D1* | Testovací množina | 58,76 | 8823 | 31,69 |
| | 1. polovina května | 57,34 | 8099 | 33,23 |
| | 2. polovina února | 58,23 | 9234 | 28,34 |
| *Linear_complex_D1* | Testovací množina | 55,42 | 8278 | 27,75 |
| | 1. polovina května | 58,35 | 8612 | 32,69 |
| | 2. polovina února | 52,55 | 8069 | 22,52 |

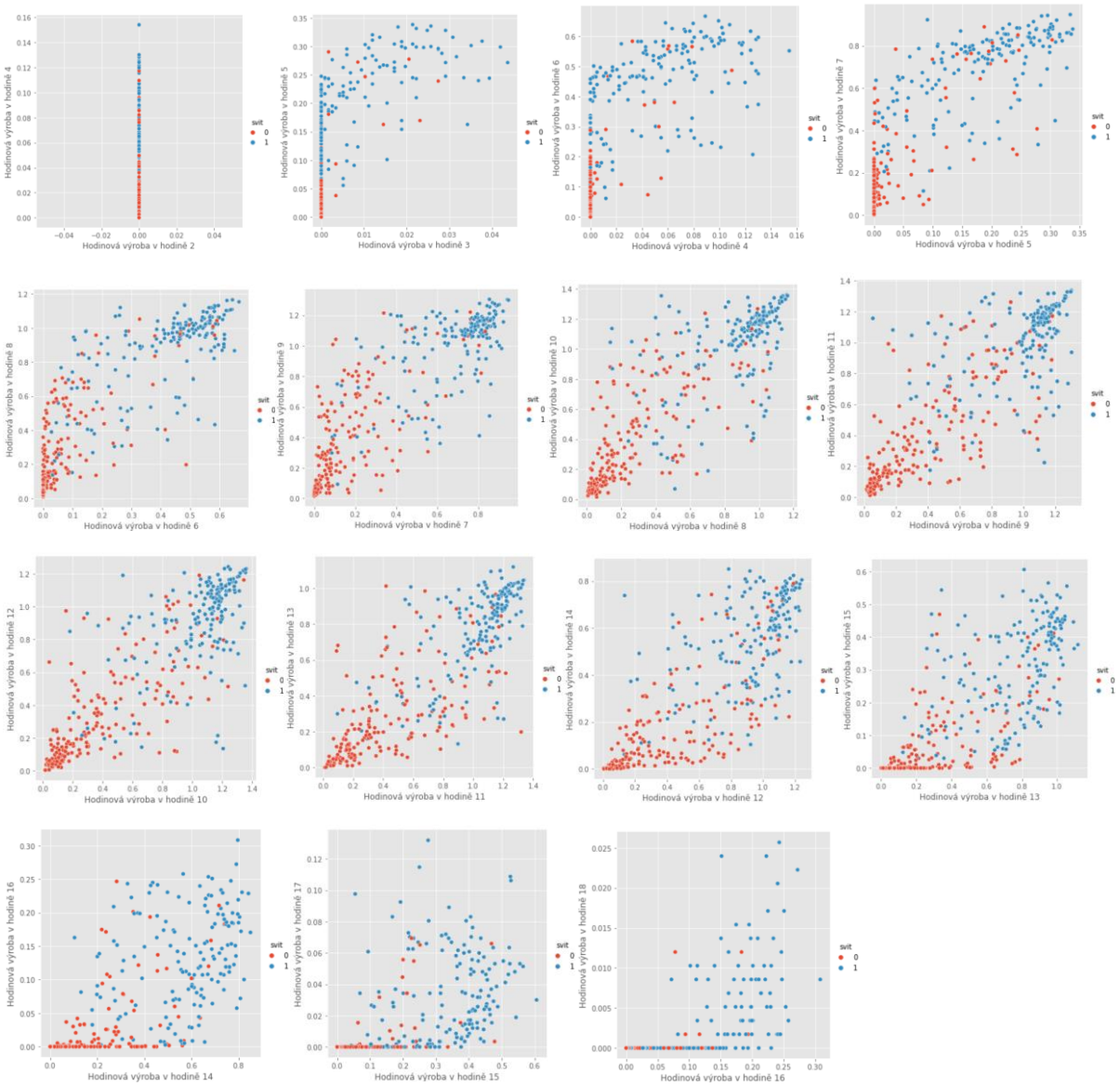
### 2.3.7 *Constant_H2_linear*

Tento model predikuje výrobu elektřiny solární elektrárny v hodině h ze znalosti výroby elektřiny v hodině $h-2$ v daném dni d a měsíci m. Závislost mezi výrobou elektřiny v hodině h na hodině $h-2$ je zobrazena na následujícím grafu, na kterém si můžeme všimnout lineární závislosti mezi oběma veličinami. Data v grafu jsou rozdělena do čtyř skupin podle hodinových intervalů. Lze si povšimnout periodicity této závislosti.




Obrázek 2.58: Závislost hodinové výroby elektřiny v hodině h na výrobě elektřiny v hodině $h-2$ s ohledem na dělení dat na čtyři hodinové intervaly. Data jsou standardizována a jsou čerpána z tabulky *train*.

Níže uvedené grafy znázorňují závislosti výroby v hodině h na výrobě v hodině $h-2$ postupně pro všechny hodinové intervaly, lze si povšimnout clustrové povahy dat pro hodnoty veličiny *svit* a velmi dobré lineární závislosti. Hodnoty ve sloupci *svit* se vztahují k hodině h.




Obrázek 2.59–2.73: Závislosti standardizovaných hodnot výroby elektřiny v hodinách h na výrobách elektřiny v hodině $\mathrm{h}-2$ s ohledem na dělení dat podle binárních hodnot ve sloupci *svit*

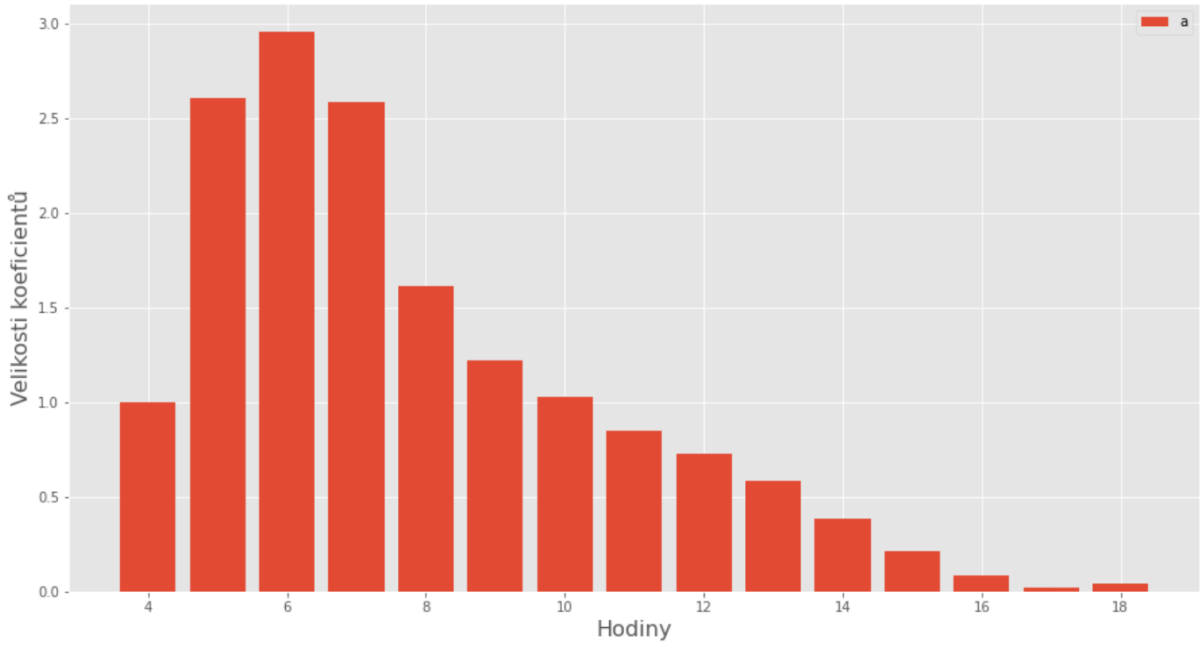
Náš model vznikne přidáním členu $a(\mathrm{h}) \cdot v^{\mathrm{d}}_{\mathrm{m,h}-2}$ do modelu *Constant_D1_linear*, kde $a(\mathrm{h})$ je parametr, který odpovídá danému hodinovému intervalu, pro který chceme předpovídat výrobu elektřiny a $v^{\mathrm{d}}_{\mathrm{m,h}-2}$ je výroba elektřiny v měsíci m, dni d a hodině $\mathrm{h}-2$. Model definujeme následovně

$$v^{\mathrm{d}}_{\mathrm{m,h}} = a(\mathrm{h})v^{\mathrm{d}}_{\mathrm{m,h}-2} + b(\mathrm{m},\mathrm{h}).$$

```
with pm.Model() as model_Constant_H2_linear:

    data_vstup  = pm.Data("data_vstup",  train['hodinová_výroba_zpět_2'].values)
    mesice      = pm.Data('mesice', train['měsíc'].values-1)
    hodiny      = pm.Data("hodiny", train['hodina'].values-4)

    w0 = pm.Normal('w0', mu=train['hodinová_výroba_zpět_2'].mean(), sd=1, shape=[12,15])
    a  = pm.Normal('a', mu=1, sd = 1, shape = 15)

    w0 = w0[mesice ,hodiny ]
    a  = a[hodiny ]*data_vstup

    rovnice = w0 + a

    sd = pm.HalfNormal('sd', sd=5, shape = 1)
    likelihood = pm.Logistic('likelihood',
                          mu= rovnice,
                          s=sd,
                          observed=train['hodinová_výroba'].values)
```

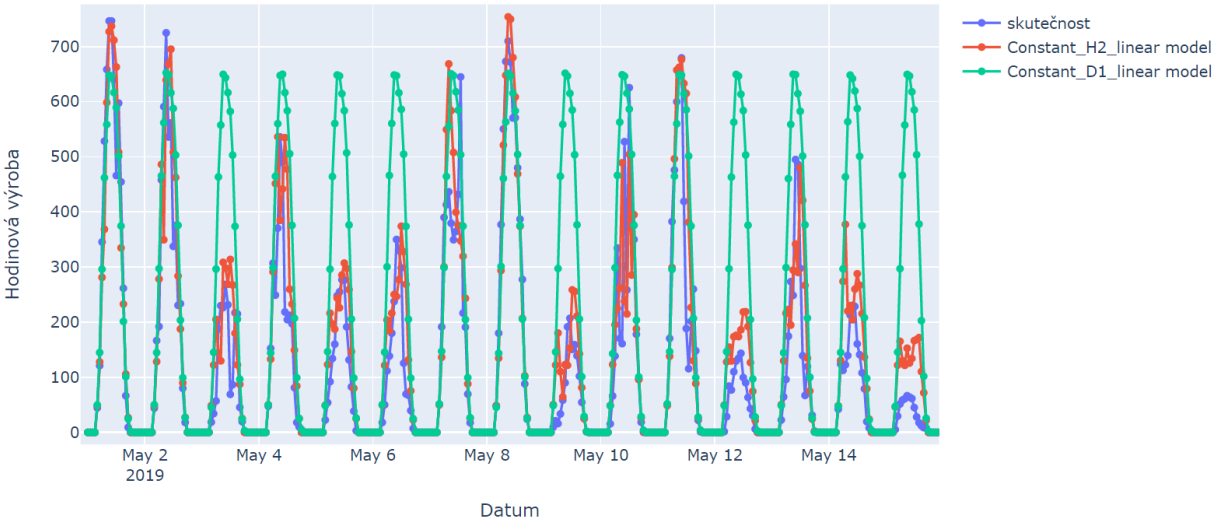
Zdrojový kód 8: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Constant_H2_linear* v *Pymc3*

Níže uvedený obrázek ukazuje velikosti koeficientů $a(\mathrm{h})$ pro zjišťovanou hodnotu predikce pro danou hodinu. Pokud je velikost koeficientu $a(\mathrm{h}) > 1$, pak je předpověď pro danou hodinu (pro hodinu h) vyšší, než pro hodinu $\mathrm{h} - 2$, pokud je velikost koeficientu $a(\mathrm{h}) < 1$, pak je předpověď pro danou hodinu nižší než pro hodinu $\mathrm{h} - 2$.

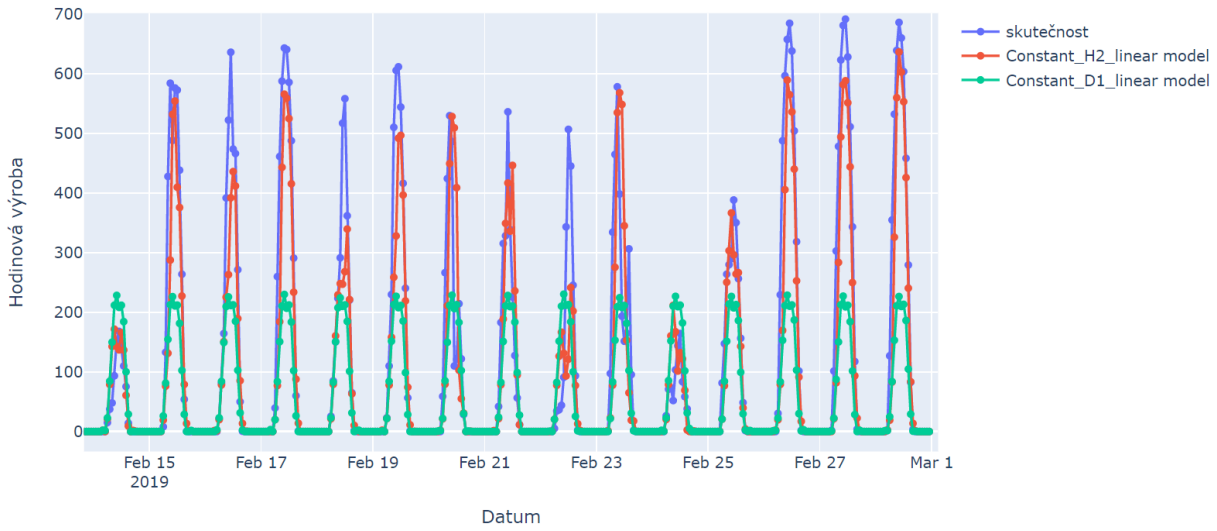



Obrázek 2.74: Vyobrazení velikostí koeficientů modelu *Constant_H2_linear*

Na obrázcích vidíme srovnání přesnosti dvou modelů, které se liší jen ve využití koeficientu $a(\mathrm{h})$ ve prospěch tohoto nového modelu. Výroba elektřiny v hodině $\mathrm{h} - 2$ má na výrobu elektřiny v hodině h významný vliv.




Obrázek 2.75: Srovnání predikce modelu *Constant_H2_linear* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_linear* v první polovině měsíce května




Obrázek 2.76: Srovnání predikce modelu *Constant_H2_linear* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_linear* v druhé polovině měsíce února

Tabulka 2.8: Srovnání chybových funkcí modelů *Constant_H2_linar* a *Constant_D1_linar* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Constat_D1_linear* | Testovací množina | 95,57 | 23015 | 49,41 |
| | 1. polovina května | 170,9 | 56722 | 111,22 |
| | 2. polovina února | 113,66 | 32920 | 47,05 |
| *Constant_H2_linear* | Testovací množina | 61,07 | 9906 | 31,00 |
| | 1. polovina května | 73,87 | 10831 | 53,27 |
| | 2. polovina února | 62,08 | 11356 | 21,96 |

### 2.3.8 *Linear_Short_term_H2*

Model, který jsme pro předpověď sestavili, je velmi podobný nejlepšímu modelu pro D + 1, jen jsme k němu přidali koeficienty násobící poslední hodinovou výrobu elektrárny, jde o autoregresi. Model lze přiblížit následující rovnicí

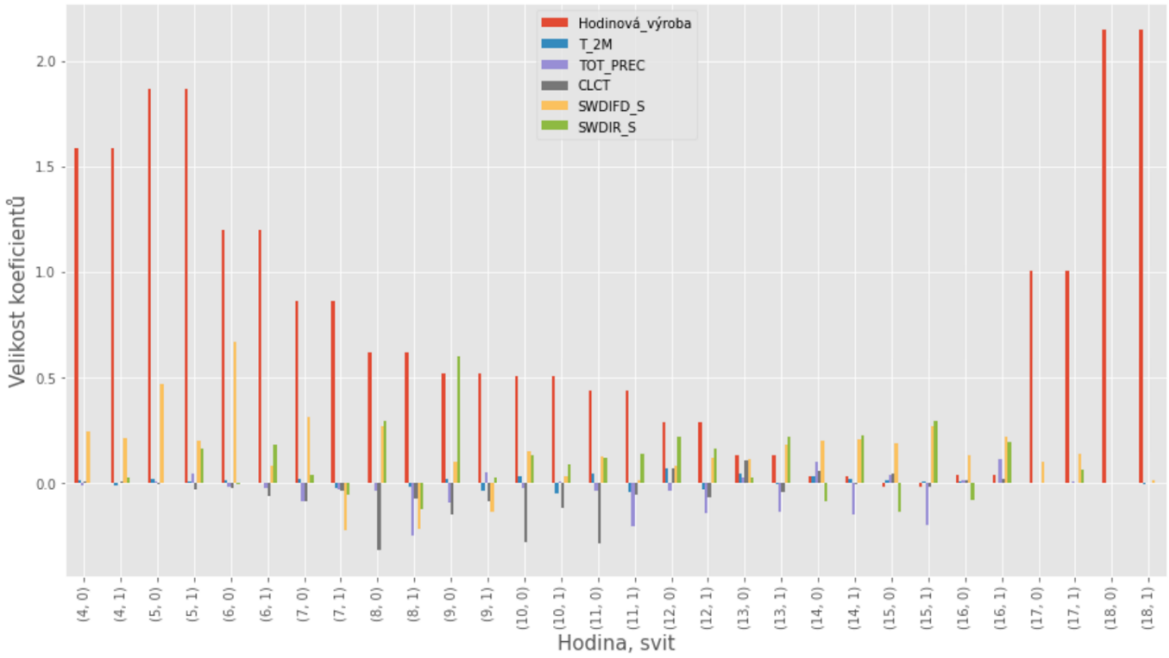
$$v^{\mathrm{d}}_{\mathrm{m,h}} = a(\mathrm{h})v^{\mathrm{d}}_{\mathrm{m,h-2}} + b(\mathrm{h,s}) \cdot \mathbf{x}^{\mathrm{d}}_{\mathrm{m,h}} + c(\mathrm{h,s}).$$

Koeficienty $a(\mathrm{h})$ jsou odlišné pro různé hodinové intervaly.

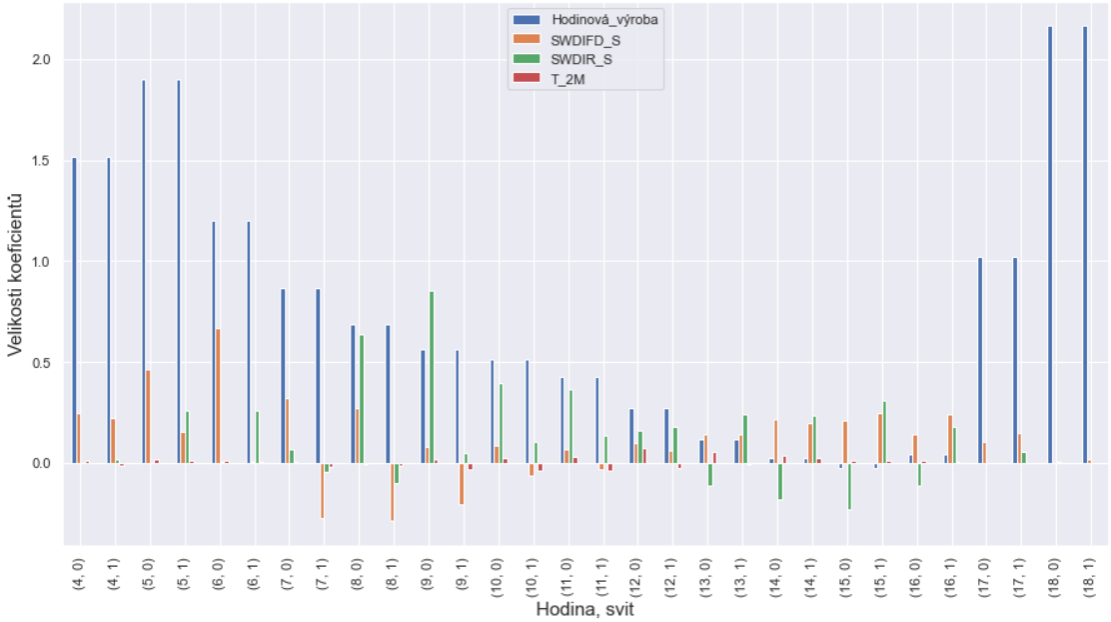
```
with pm.Model() as Linear_Short_term_H2:

    vstup_vyroba = pm.Data("vstup_vyroba",  train['hodinová_výroba_zpět_2'].values)
    vstup_pocasi = pm.Data("vstup_pocasi",  train[vstupy].values)
    hodiny       = pm.Data("hodiny", train['hodina'].values-4)
    hodiny_zpet  = pm.Data("hodiny_zpet", train['hodina_zpět_2'].values.astype(int)-4)
    svity        = pm.Data('svity', train['svit'].values)

    w0 = pm.Normal('w0', mu=train['hodinová_výroba_zpět_2'].mean(), sd=1, shape=[15,2])
    w  = pm.Normal('w', mu=0, sd=0.25,shape = [5,15,2])
    a  = pm.Normal('a', mu=1, sd = 1, shape = 15)

    w0 = w0[hodiny,svity ]
    a  = a[hodiny_zpet]*vstup_vyroba
    w  = w[:, hodiny,svity].transpose()

    rovnice =  w0 + a + (vstup_pocasi * w).sum(axis=1)

    sd = pm.HalfNormal('sd', sd=5, shape = 1)
    likelihood = pm.Logistic('likelihood',
                             mu= rovnice,
                             s=sd,
                             observed=train['hodinová_výroba'].values)
```

Zdrojový kód 9: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Linear_Short_term_H2* v *Pymc3*

Oba obrázky 2.77–2.78 ukazují, že parametry $a(\mathrm{h})$ (které násobí hodinovou výrobu v hodině $\mathrm{h} - 2$) mají na predikci výroby elektrárny nejzásadnější vliv, hlavně v brzkých ranních hodinách. Pro 17. a 18. hodinu nabývaly koeficienty $a(\mathrm{h})$ velkých hodnot kvůli ne moc rozumným závislostem mezi výrobou v hodině $\mathrm{h} - 2$ a výrobou v hodině h.

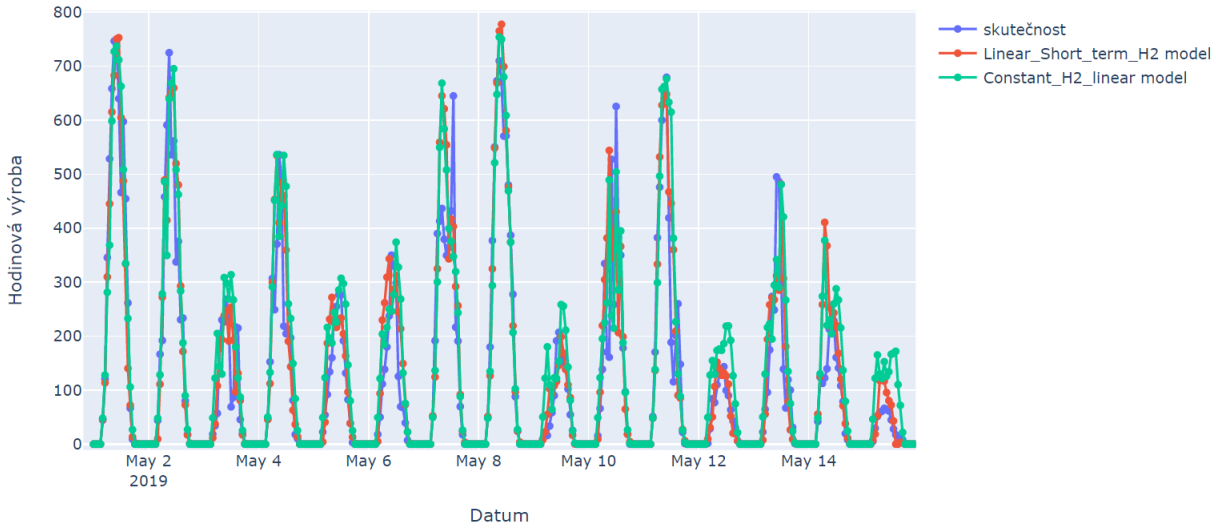



Obrázek 2.77: Grafické zobrazení parametrů modelu *Linear_Short_term_H2*, ve kterém se berou v úvahu i nepotřebná data o počasí, tj. *TOT_PREC* a *CLCT*.

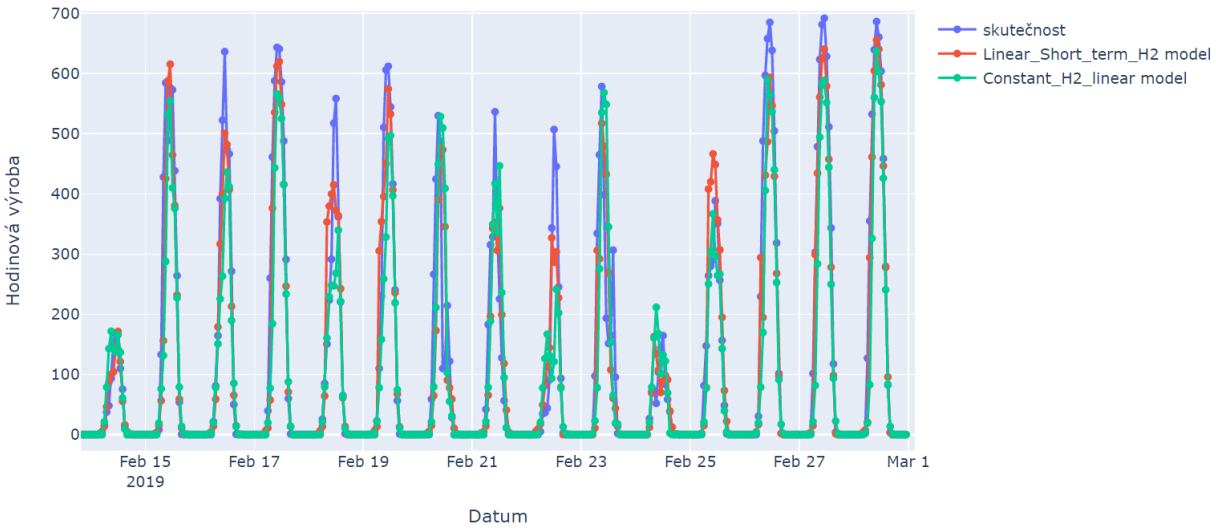



Obrázek 2.78: Grafické zobrazení parametrů modelu *Linear_Short_term_H2*, ve kterém se berou v úvahu jen potřebná data o počasí, tj. *SWDIR_S*, *SWDIFD_S* a *T_2M*.

Z grafů je vidno, že krátkodobý model s počasím nabývá vyšší přesnosti.




Obrázek 2.79: Srovnání predikce modelu *Linear_short_term_H2* se skutečnou výrobou elektřiny a predikcí modelu *Constant_H2_linear* v první polovině měsíce května




Obrázek 2.80: Srovnání predikce modelu *Linear_short_term_H2* se skutečnou výrobou elektřiny a predikcí modelu *Constant_H2_linear* v druhé polovině měsíce února

Tabulka 2.9: Srovnání chybových funkcí modelů *Linear_short_term_H2* a *Constant_H2_linar* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Constant_H2_linear* | Testovací množina | 61,04 | 9904 | 30,98 |
| | 1. polovina května | 73,87 | 10831 | 53,27 |
| | 2. polovina února | 62,08 | 11356 | 21,96 |
| *Linear_Short_term_H2* | Testovací množina | 61,07 | 9906 | 31,00 |
| | 1. polovina května | 56,69 | 7742 | 31,16 |
| | 2. polovina února | 58,35 | 8612 | 32,69 |

Chyby modelu jsou menší než ve všech předchozích modelech, opět jsme sestavili lepší model než předtím a ukázalo se, že znalost hodnoty výroby elektřiny v hodině $h - 2$ má na předpověď pozitivní vliv.

## 2.4 Neuronové modely

V této kapitole si ukážeme, jak předpovídat výrobu solární elektrárny pomocí bayesovských dopředných vícevrstvých neuronových sítí jak pro $D + 1$, tak pro $H + 2$. Ke zjištění parametrů modelu jsem využil Python knihovnu *Pymc3*, stejně jako v případě lineárních modelů, které jsem už popsal. V případech, kdy k predikci využíváme i hodnoty počasí, budeme pracovat jen se 3 zmíněnými vstupy (*SWDIR_S*, *SWDIFD_S* a *T_2M*). V této kapitole byly využity zdroje [1], [2], [4], [6], [7], [8].

### 2.4.1 *Constant_D1_neuron*

Tento model predikuje výrobu solární elektrárny na den dopředu jen ze znalosti měsíce a hodiny, pro kterou chceme předpovídat výrobu, stejně jako v případě nejjednoduššího modelu pro $D + 1$ předpověď *Constant_D1_linear*. Nejdříve si do proměnné *vstupy_3* uložíme názvy sloupců vstupních dat z tabulky *info* a definujeme matice vstupních dat a vektory výstupů pro trénovací a testovací množinu.

```
vstupy_3 = ['měsíc', 'hodina_jinak']

vstupy_train  = np.reshape(train[vstupy_3].values, [-train[vstupy_3].shape[1], train[vstupy_3].shape[1]])
vystupy_train = np.reshape(train['hodinová_výroba'].values, [-1, 1])

vstupy_test  = np.reshape(test[vstupy_3].values, [-test[vstupy_3].shape[1], test[vstupy_3].shape[1]])
vystupy_test = np.reshape(test['hodinová_výroba'].values, [-1, 1])
```

Zdrojový kód 10: Definování vstupních a výstupních dat trénovací a testovací množinu modelu *Constant_D1_neuron*

Následně do proměnných *n_hidden_1* a *n_hidden_2* uložíme čísla, která znamenají počet neuronů v první a druhé vrstvě neuronové sítě. Veličiny *init_1*, *init_2*, *init_3* jsou matice, které v případě bayesovských neuronových sítí zlepšují nafitování vah.

```
n_hidden_1 = 10
n_hidden_2 = 2          # Počty neuronů v jednotlivých vrstvách

init_1   = np.random.randn(vstupy_train.shape[1], n_hidden_1)
init_2   = np.random.randn(n_hidden_1, n_hidden_2)             # Náhodné váhy v jednotlivých vrstvách
init_out = np.random.randn(n_hidden_2, 1)
```

Zdrojový kód 11: Definování pomocných polí *init_1*, *init_2*, *init_out*

Níže uvedený kód definuje neuronovou síť, která nepracuje s biasy, kde *s* je parametr věrohodnostní funkce, *weights_1*, *weights_2* a *weights_out* jsou apriorními rozděleními vah

neuronové sítě, kterým je dáno normální rozdělení, *layer_1* a *layer_2* jsou vektory perceptronů ve vstupních vrstvách. Věrohodnostní funkci je dáno logistické rozdělení pravděpodobnosti.

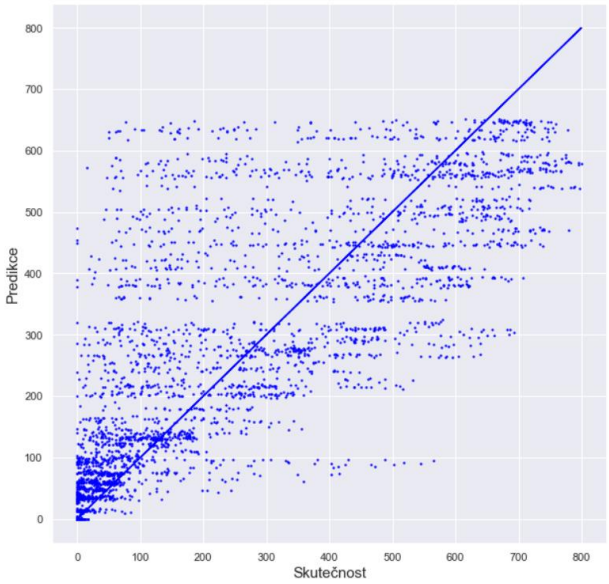
```
SD = 1
with pm.Model() as Constant_D1_neuron:

    data_vstup  = pm.Data("data", vstupy_train)

    sd_1 = pm.HalfNormal('sd_1', sd = 5)

    weights_1 = pm.Normal('layer1', mu=0, sd=SD, shape=[vstupy_train.shape[1], n_hidden_1], testval=init_1)
    weights_2 = pm.Normal('layer2', mu=0, sd=SD, shape=[n_hidden_1, n_hidden_2], testval=init_2)
    weights_out = pm.Normal('out' , mu=0, sd=SD, shape=[n_hidden_2, 1], testval=init_out)

    layer_1   = pm.math.tanh((data_vstup @ weights_1))
    layer_2   = pm.math.tanh((layer_1 @ weights_2))
    layer_out = layer_2 @ weights_out

    likelihood = pm.Logistic('likelihood', mu = layer_out , observed=vystupy_train,s = sd_1)
```

Zdrojový kód 12: Definování apriorních rozdělení parametrů a věrohodnostní funkce modelu *Constant_D1_neuron* v *Pymc3*

Následně definujeme typ výpočetního algoritmu, který vypočítá posteriorní rozdělení jednotlivých vah modelu. Poté definujeme, s jakými daty chceme pracovat a nakonec napíšeme řádek kódu, který znamená, že chceme zjistit posteriorní rozdělení jednotlivých předpovědí výroby elektřiny.

```
with Constant_D1_neuron:
    inference    = pm.ADVI()                              # definice výpočetního algoritmu
    approx       = pm.fit(n=400000, method=inference)     # definice výpočetního algoritmu
    trace_nn_1   = approx.sample(draws=4000)              # 4000 sample hodnot ke každé váze

with Constant_D1_neuron:
    pm.set_data({"data": vstupy_test})                     # vstupní data - testovací množina
    post_pred_9 = pm.sample_posterior_predictive(trace_nn_1)  # predikce testovací množina
```

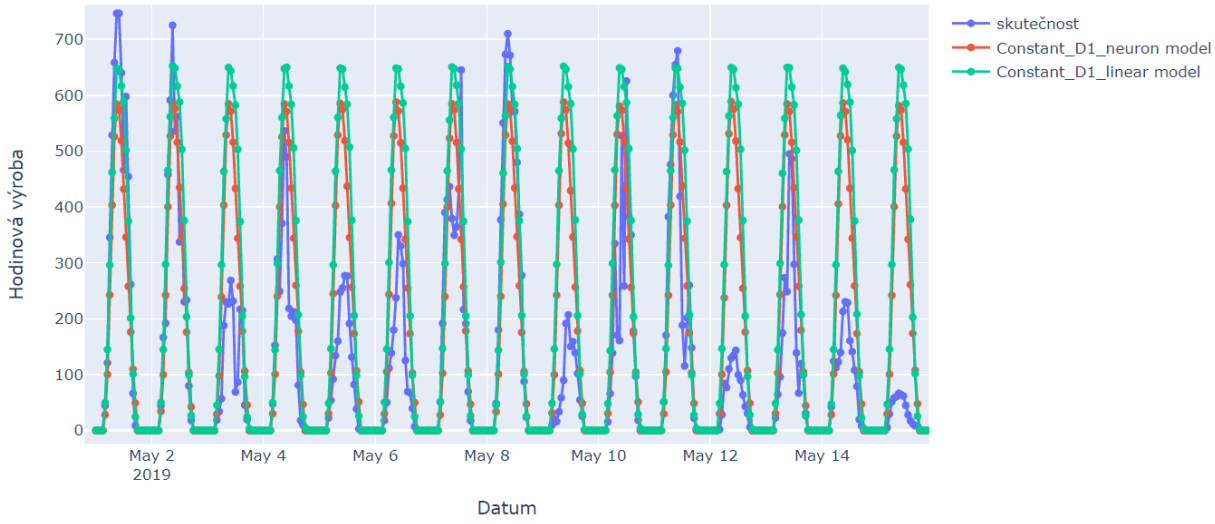
Zdrojový kód 13: Definice výpočetního algoritmu, dat použitých pro otestování modelu *Constant_D1_neuron* a uložení posteriorních hodnot predikcí do pole *post_pred_9* v *Pymc3*



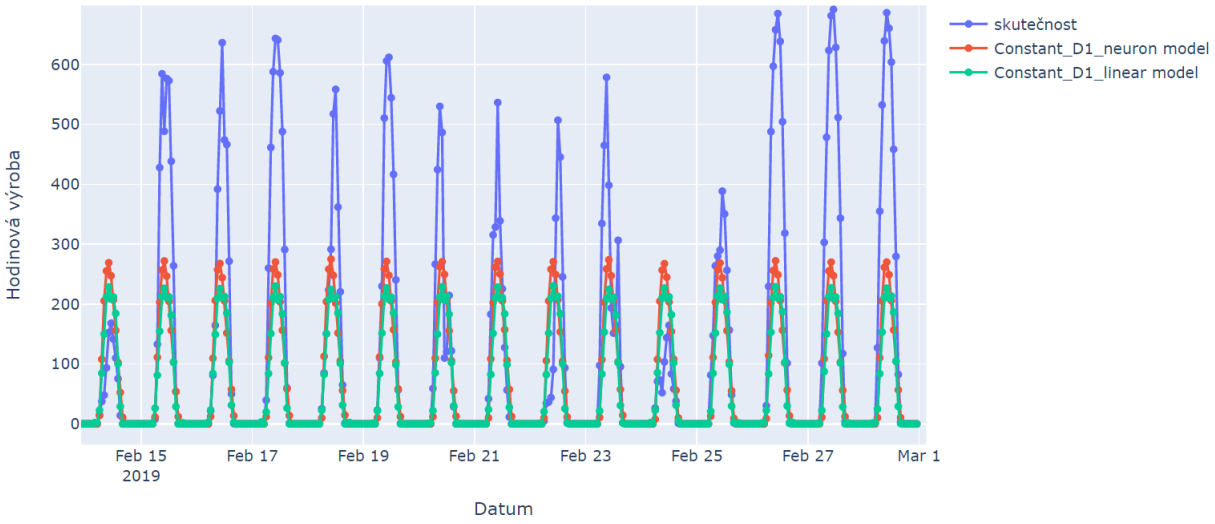

Obrázek 2.81: Grafické vyhodnocení modelu *Constant_D1_neuron*

Grafické vyhodnocení vypadá velmi podobně jako u modelu *Constant_D1_linear*. Můžeme se tedy domnívat, že tento model bude mít podobnou chybu jako v případě tohoto modelu? Pojďme se přesvědčit i jinými způsoby.

Jak je vidno z obrázků, tento model neuronových sítí pro daný měsíc nabývá konstantních solárních křivek stejně jako u porovnávaného konstantního lineárního modelu.




Obrázek 2.82: Srovnání predikce modelu *Constant_D1_neuron* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_linear* v první polovině měsíce května




Obrázek 2.83: Srovnání modelu *Constant_D1_neuron* se skutečnou výrobou elektřiny a modelem *Constant_D1_linear* v druhé polovině měsíce února

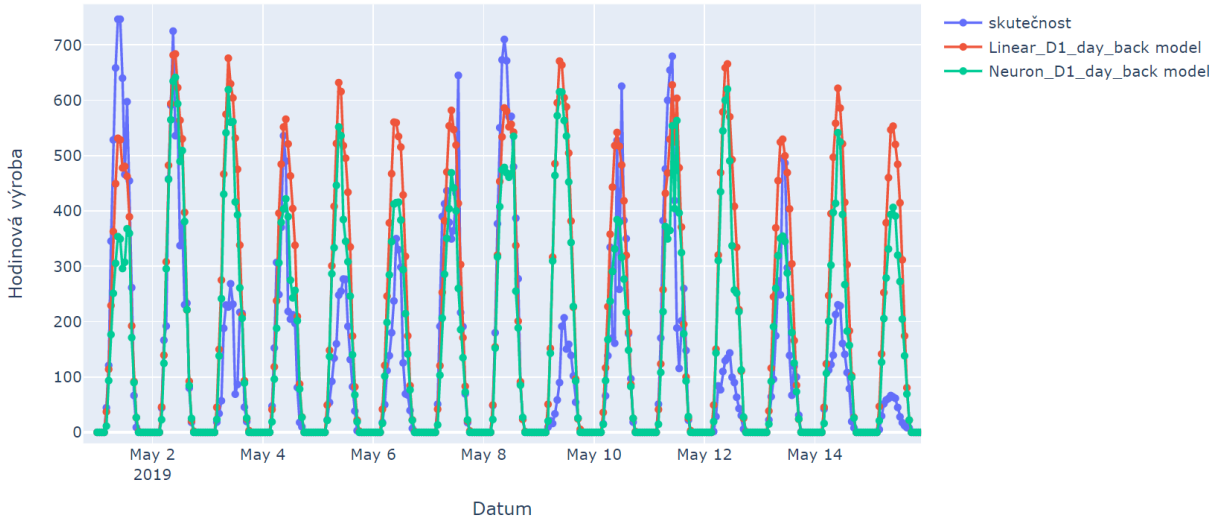
Tabulka 2.10: Srovnání chybových funkcí modelů *Constant_D1_neuron* a *Constant_D1_linear* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Constant_D1_linear* | Testovací množina | 95,57 | 23015 | 49,41 |
| | 1. polovina května | 170,9 | 56722 | 111,22 |
| | 2. polovina února | 113,66 | 32920 | 47,05 |
| *Constant_D1_neuron* | Testovací množina | 95,96 | 20708 | 56,81 |
| | 1. polovina května | 139,23 | 35844 | 96,27 |
| | 2. polovina února | 108,69 | 29352 | 38,77 |

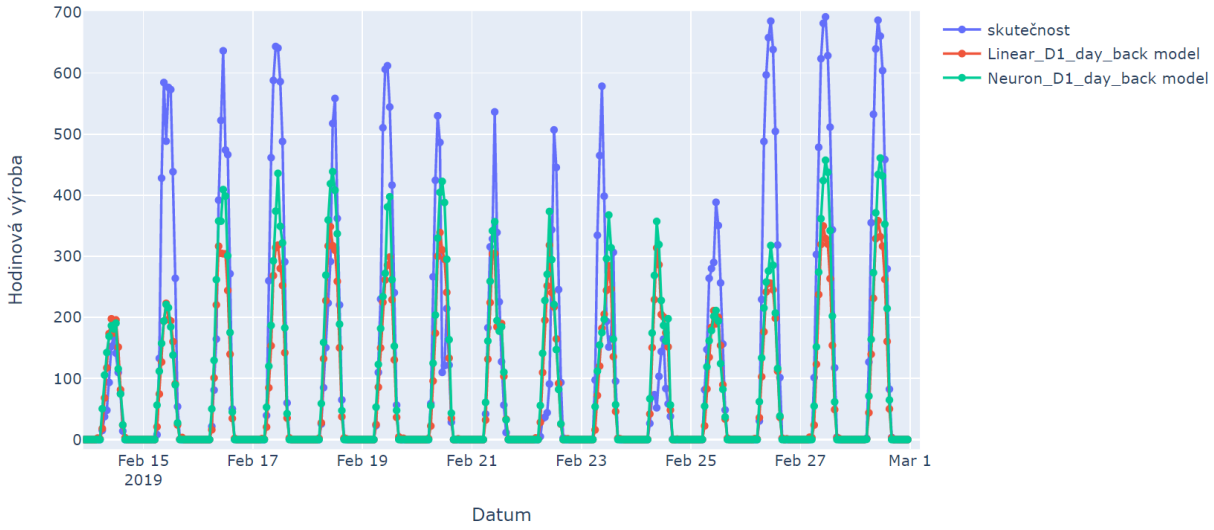
Tento model nabývá téměř stejné přesnosti na celé testovací množině jako model *Constant_D1_linear*. Zkusme tedy vytvářet další předpovědi za pomoci neuronových sítí i pro složitější modely. V následujících modelech už nebudeme ukazovat celý kód, který je pro všechny další modely neuronových sítí stejný, zachováváme i stejný počet skrytých vrstev s danými počty neuronů. Měnit budeme jen vstupy těchto neuronových sítí.

### 2.4.2 *Neuron_D1_day_back*

Cílem tohoto modelu je předpovídat výrobu elektřiny na den dopředu ze znalosti hodnot dnešní výroby elektřiny v jednotlivých hodinách dne. Tento model srovnáváme s lineárním modelem *Linear_D1_day_back*.




Obrázek 2.84: Srovnání predikce modelu *Neuron_D1_day_back* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_linear* v první polovině měsíce května



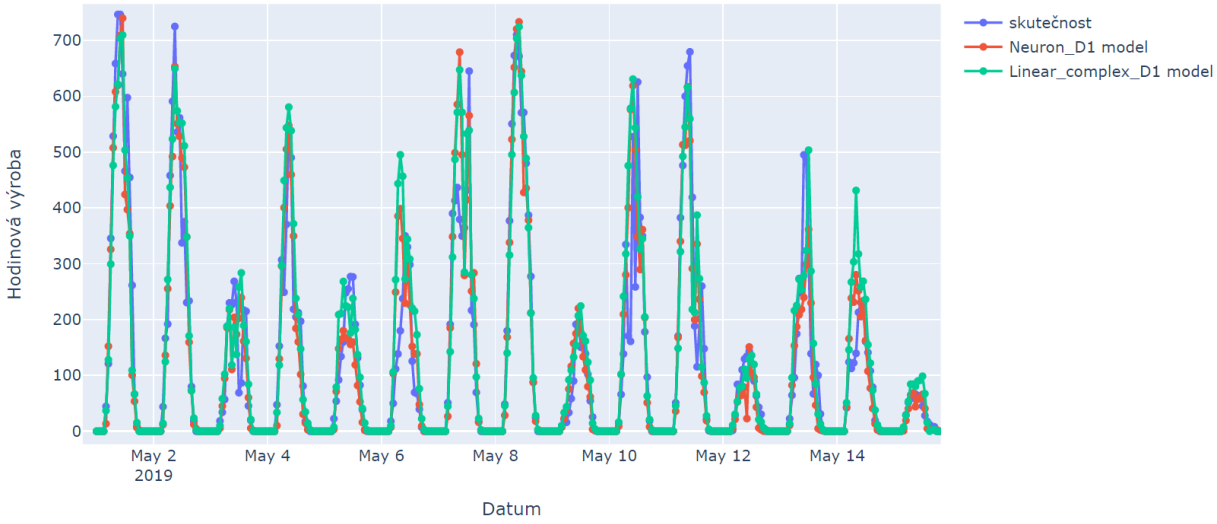

Obrázek 2.85: Srovnání predikce modelu *Neuron_D1_day_back* se skutečnou výrobou elektřiny a predikcí modelu *Constant_D1_linear* v druhé polovině měsíce února

Tabulka 2.11: Srovnání chybových funkcí modelů *Linear_D1_day_back* a *Neuron_D1_day_back* pro celou testovací množinu a dva vybrané intervaly

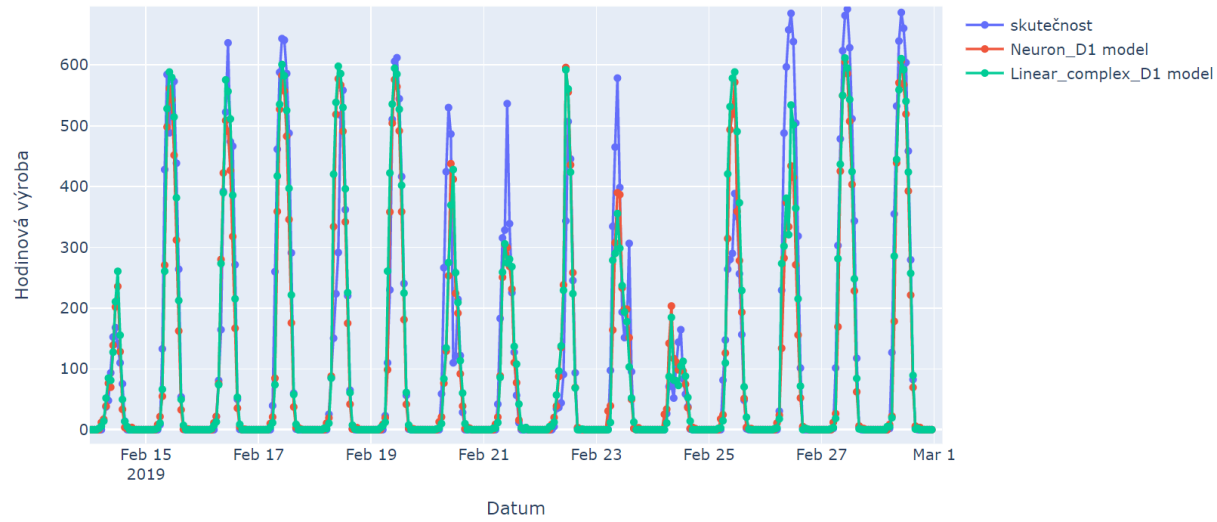
| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Neuron_D1_day_back* | Testovací množina | 88,66 | 18689 | 46,90 |
| | 1. polovina května | 128,02 | 31675 | 94,86 |
| | 2. polovina února | 92,40 | 20354 | 46,66 |
| *Linear_D1_day_back* | Testovací množina | 89,13 | 19914 | 46,74 |
| | 1. polovina května | 151,40 | 44619 | 111,91 |
| | 2. polovina února | 102,51 | 25176 | 49,34 |

### 2.4.3 *Neuron_D1*

Tento model predikuje výrobu elektřiny pro každou hodinu dalšího dne ze znalosti předpovědi počasí (v této práci bohužel jen ze znalosti skutečných hodnot počasí). Vstupy neuronové sítě jsou téměř stejné jako v případě D + 1 lineárních modelů, jen s výjimkou toho, že zde nepracujeme s hodnotami ve sloupci *svit*, protože se ukázalo, že tento sloupec, stejně jako sloupec *měsíc*, nemá vliv na lepší predikci.




Obrázek 2.86: Srovnání predikce modelu *Neuron_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_complex_D1* v první polovině měsíce května




Obrázek 2.87: Srovnání predikce modelu *Neuron_D1* se skutečnou výrobou elektřiny a predikcí modelu *Linear_complex_D1* v druhé polovině měsíce února

Tabulka 2.12: Srovnání chybových funkcí modelů *Neuron_D1* a *Linear_complex_D1* pro celou testovací množinu a dva vybrané intervaly

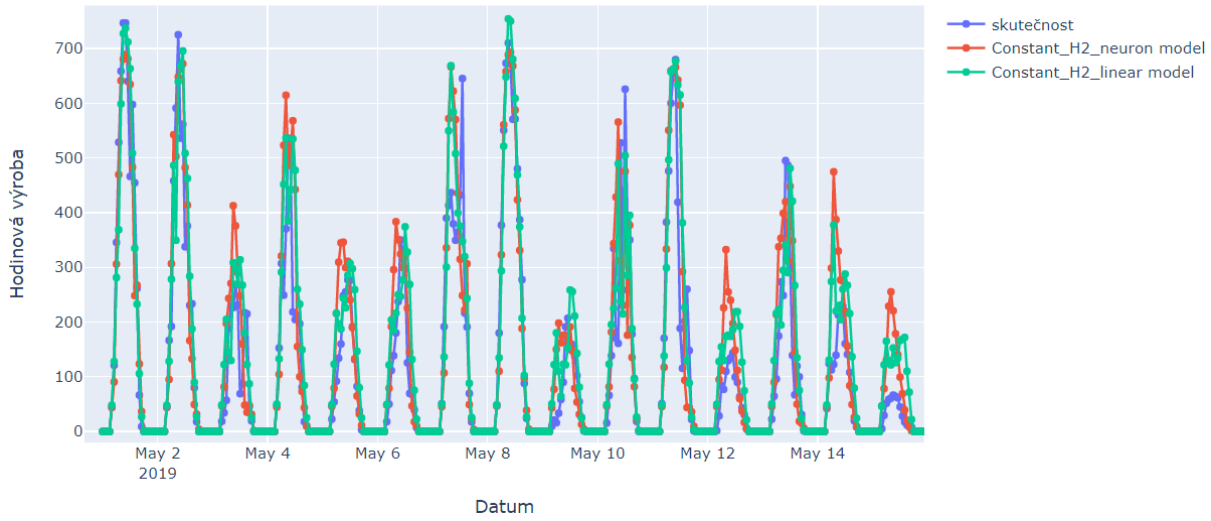
| | | MAE kWh | MSE kWh$^2$ | MAO kWh |
|---|---|---|---|---|
| *Linear_complex_D1* | Testovací množina | 55,42 | 8278 | 27,75 |
| | 1. polovina května | 58,35 | 8612 | 32,69 |
| | 2. polovina února | 52,55 | 8069 | 22,52 |
| *Neuron_D1* | Testovací množina | 55,27 | 8184 | 27,33 |
| | 1. polovina května | 53,38 | 6822 | 30,49 |
| | 2. polovina února | 55,46 | 8342 | 21,38 |

Dle výše uvedených hodnot nabývá tento model téměř totožné přesnosti jako nejsložitější a nejpřesnější lineární model, tj. *Linear_complex_D1*. Ukázalo se, že neuronová síť umí výborně zužitkovat vstupní data.

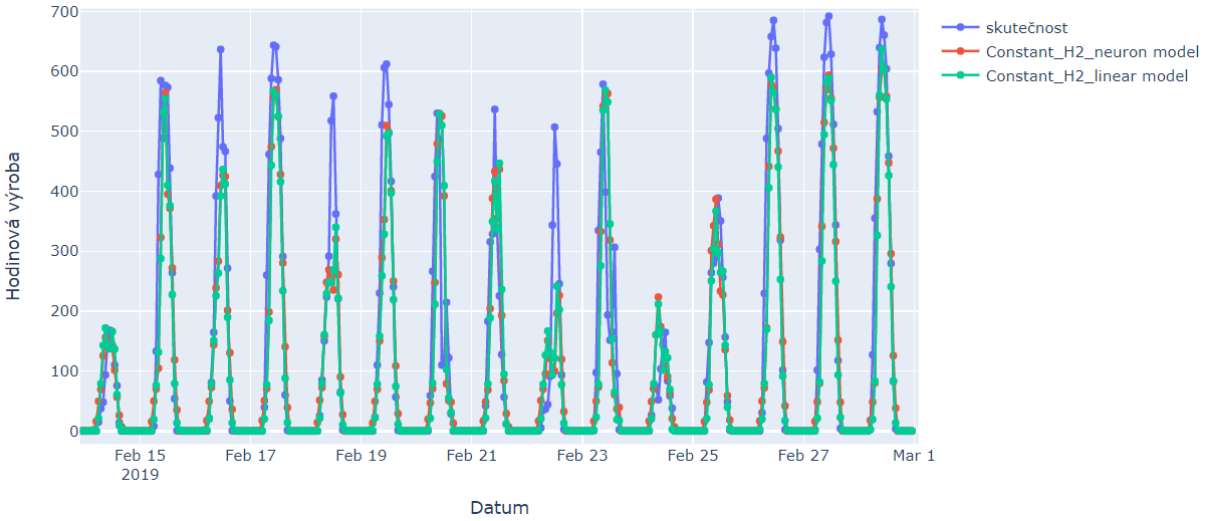
V následujícím textu budeme pokračovat s predikcemi modelů $H + 2$ pomocí neuronových sítí.

### 2.4.4 *Constant_H2_neuron*

Tento model má za cíl předpovědět výrobu elektřiny v hodině h ze znalosti hodinové výroby v hodině $h - 2$. S počasím jako vstupem zde nepracujeme. Vstupy tohoto modelu jsou stejné jako u modelu lineární regrese *Constant_H2_linear*, se kterým porovnáváme zkoumaný model.




Obrázek 2.88: Srovnání predikce modelu *Constant_H2_neuron* se skutečnou výrobou elektřiny a predikcí modelu *Constant_H2_linear* v první polovině měsíce května.




Obrázek 2.89: Srovnání predikce modelu *Constant_H2_neuron* se skutečnou výrobou elektřiny a predikcí modelu *Constant_H2_linear* v druhé polovině měsíce února

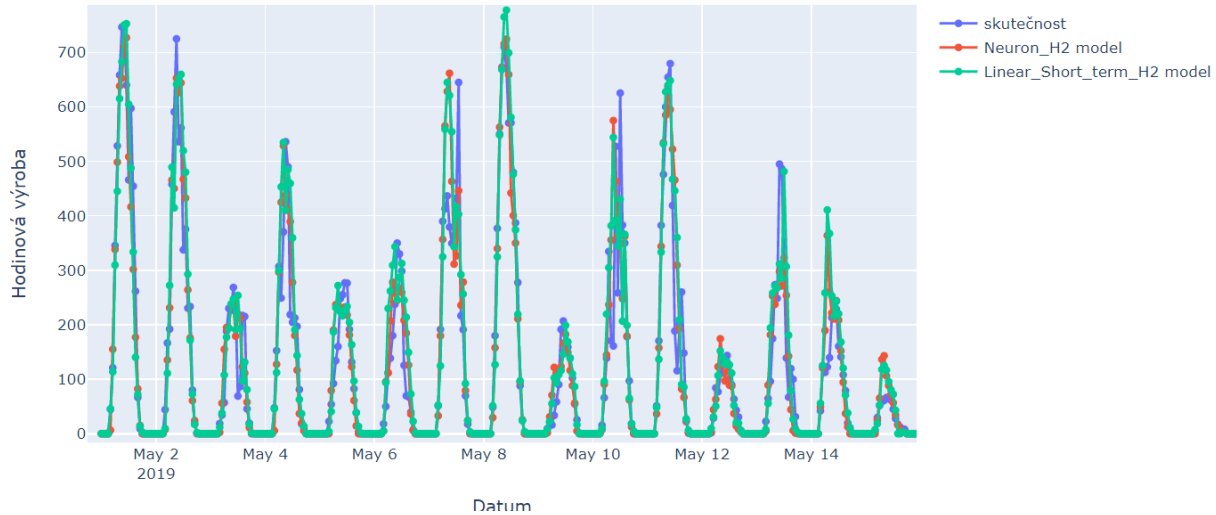
Tabulka 2.13: Srovnání chybových funkcí modelů *Constant_H2_neuron* a *Constant_H2_linear* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Constant_H2_linear* | Testovací množina | 61,07 | 9906 | 31,00 |
| | 1. polovina května | 73,87 | 10831 | 53,27 |
| | 2. polovina února | 62,08 | 11356 | 21,96 |
| *Constant_H2_neuron* | Testovací množina | 63,12 | 10467 | 32,99 |
| | 1. polovina května | 75,23 | 12295 | 48,02 |
| | 2. polovina února | 63,02 | 10933 | 31,76 |

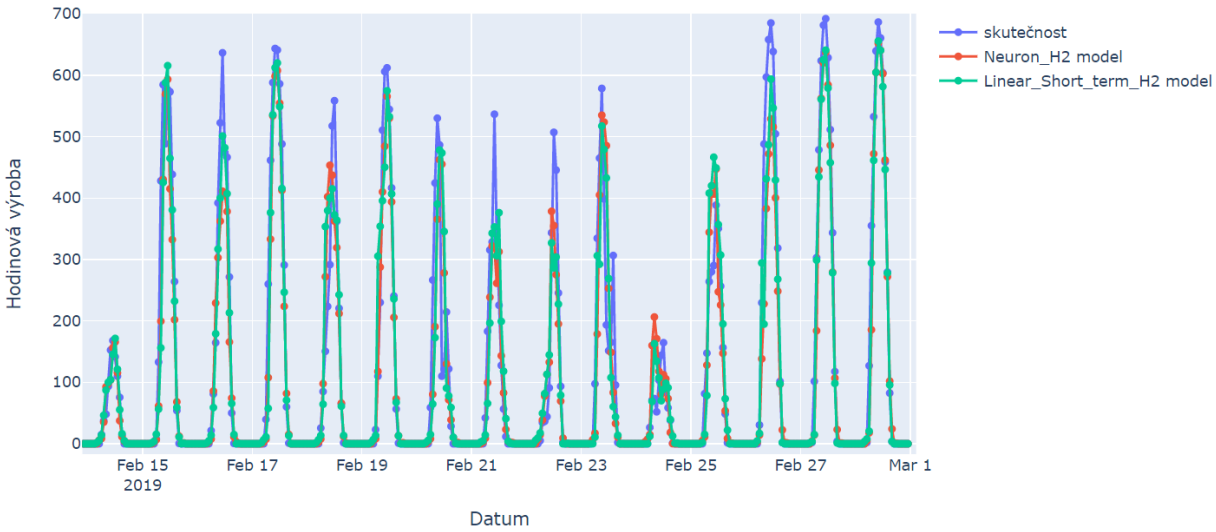
Tento model je jen nepatrně horší než model *Constant_H2_linear*, tedy model, který předpovídá výrobu elektřiny pro hodinu h ze znalosti výroby v hodině $h - 2$.

### 2.4.5 *Neuron_H2*

Cílem tohoto modelu je predikovat výrobu elektřiny v hodině h ze znalosti výroby elektřiny v hodině h − 2 a také ze znalosti předpovědi počasí pro hodinu h. Lineární model, který použijeme ke srovnání modelu *Neuron_H2* se jmenuje *Linear_Short_term_H2*. Model *Linear_Short_term_H2* je nejpřesnější model ze všech uvažovaných modelů lineární regrese.




Obrázek 2.90: Srovnání predikce modelu *Neuron_H2* se skutečnou výrobou elektřiny a predikcí modelu *Linear_Short_term_H2* v první polovině měsíce května




Obrázek 2.91: Srovnání predikce modelu *Neuron_H2* se skutečnou výrobou elektřiny a predikcí modelu *Linear_Short_term_H2* v druhé polovině měsíce únor

Zkoumaný model je celkově skoro stejně přesný jako s ním srovnávaný lineární model. Na dvou vybraných vzorcích dat (viz. Tabulka 2.13) je ovšem přesnější neuronová síť.

Tabulka 2.14: Srovnání chybových funkcí modelů *Neuron_H2* a *Linear_Short_term_H2* pro celou testovací množinu a dva vybrané intervaly

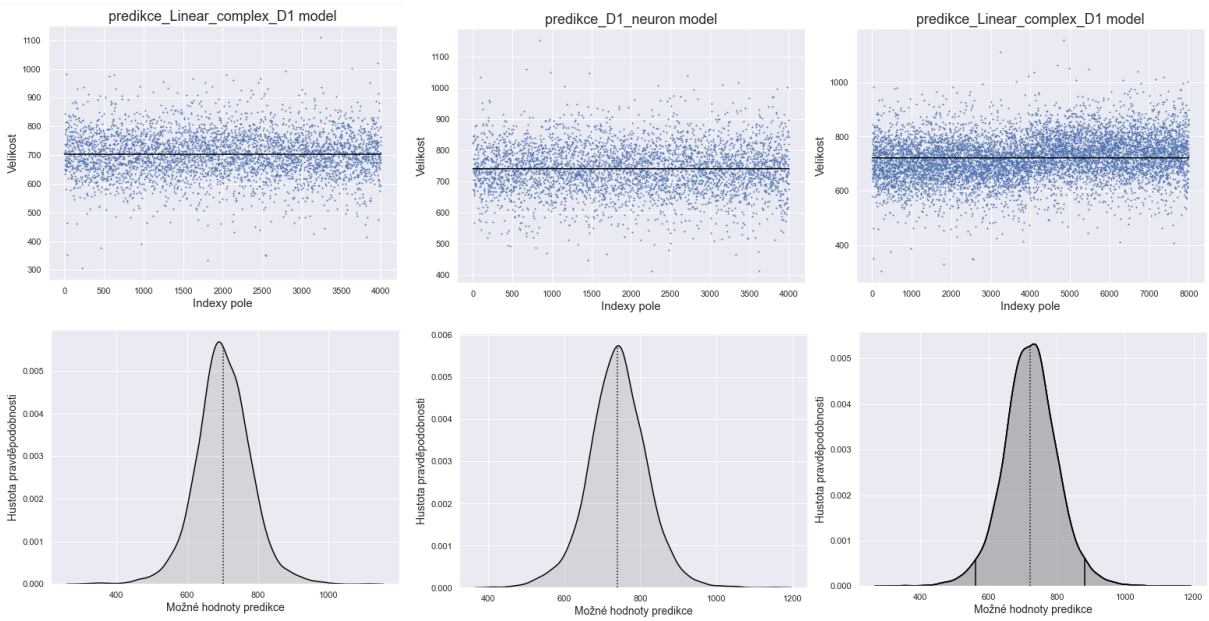
| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Linear_Short_term_H2* | Testovací množina | 51,42 | 7501 | 23,31 |
| | 1. polovina května | 56,69 | 7742 | 31,16 |
| | 2. polovina února | 58,35 | 8612 | 32,69 |
| *Neuron_H2* | Testovací množina | 51,00 | 7568 | 22,23 |
| | 1. polovina května | 52,11 | 6542 | 29,66 |
| | 2. polovina února | 49,72 | 6583 | 19,74 |

## 2.5 Výsledné modely

### 2.5.1 *Combine_D1*

Výslednou předpověď pro D + 1 výrobu elektřiny $v_{\mathrm{m,h}}^{\mathrm{d}}$ pro měsíc m, den d a hodinový interval h zjistíme tak, že spojíme pole čísel odpovídající m, d a h, kde jsou uloženy posteriorní hodnoty z neuronové sítě a nejlepšího lineárního modelu pro D + 1. To v našem případě znamená, že spojíme pole s 4000 čísly, které jsme získali z modelu lineární regrese a 4000 čísly získaných z modelu neuronové sítě. Jde o sample hodnoty jednotlivých předpovědí výroby elektřiny. Z těchto 8000 čísel uděláme průměr a toto průměrné číslo budeme brát jako předpověď, kterou použijeme k vyhodnocení modelu. Celý proces neprovádíme samozřejmě jen pro jednu předpověď, ale pro celou testovací množinu.

Obrázky znázorňují výše uvedený popis pro 11.5.2019 a hodinový interval mezi 11–12. hodinou. První dva horní obrázky zleva ukazují hodnoty v poli, kde jsou uloženy posteriorní hodnoty predikce, třetí horní obrázek zprava je spojení obou polí, černé úsečky znázorňují průměr v jednotlivých polích. Tři spodní obrázky ukazují posteriorní distribuce předpovědí, kde přerušovaná úsečka označuje průměr těchto čísel. Ve třetím spodním obrázku zleva jsou navíc dvě černé nepřerušované úsečky. Mezi těmito úsečkami je součet plochy pod křivkou 0,95, tuto oblast nazýváme 95% interval spolehlivosti. Celkový součet plochy pod křivkou je roven jedné.




Obrázek 2.92–2.97: Grafické vyobrazení sample hodnot jedné dané predikce pro modely *Linear_complex_D1, Neuron_D1, Linear_complex_D1* a z těchto hodnot vytvořených hustot pravděpodobnosti

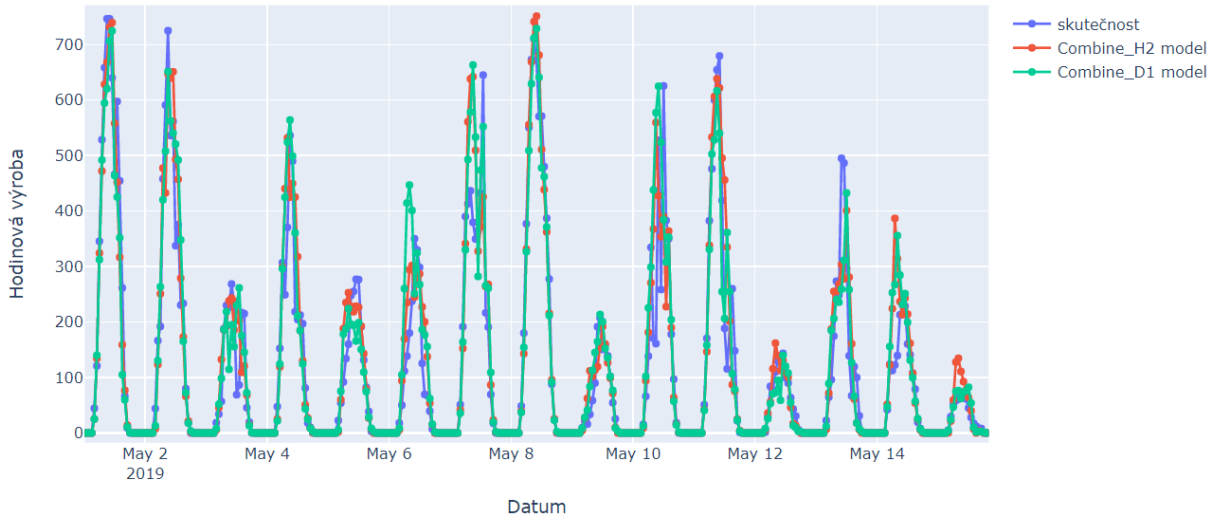
U obou kombinovaných modelů navíc vyhodnotíme, zdali skutečná hodnota výroby elektrické energie spadá do tohoto predikovaného 95% intervalu. Ukázalo se, že skutečná hodnota do tohoto intervalu spadá v **92,02 %** případů.

### 2.5.2 *Combine_H2*

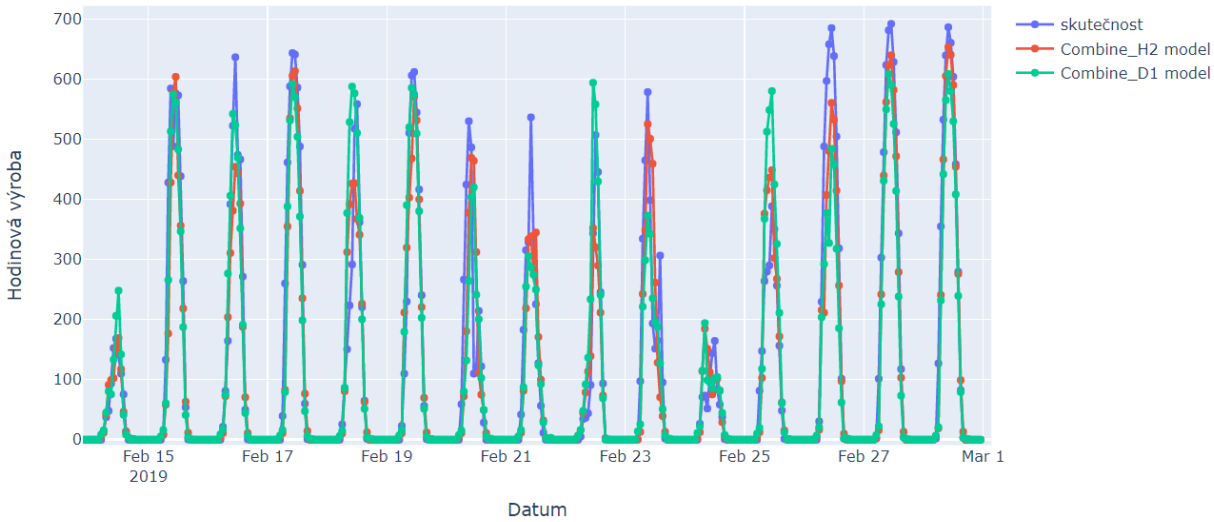
Celkové číselné vyhodnocení modelu pro H + 2, je následující. Model sestavíme stejně jako v případě modelu *Combine_D1*, jen s předpověďmi pro krátkodobou predikci.

Skutečné hodnoty v **90,49 %** případů spadají do 95% intervalu spolehlivosti.

Z obrázků můžeme usoudit, že oba modely byly relativně přesné pro zimní i letní měsíce.




Obrázek 2.98: Srovnání predikce modelu *Combine_D1* se skutečnou výrobou elektřiny a predikcí modelu *Combine_H2* v první polovině měsíce května




Obrázek 2.99: Srovnání predikce modelu *Combine_D1* se skutečnou výrobou elektřiny a predikcí modelu *Combine_H2* v druhé polovině měsíce února

Tabulka 2.15: Srovnání chybových funkcí modelů *Combine_H2* a *Combine_D1* pro celou testovací množinu a dva vybrané intervaly

| | | MAE kWh | MSE $kWh^2$ | MAO kWh |
|---|---|---|---|---|
| *Combine_D1* | Testovací množina | 54,25 | 7921 | 26,40 |
| | 1. polovina května | 54,23 | 7345 | 29,61 |
| | 2. polovina února | 53,57 | 6840 | 28,37 |
| *Combine_H2* | Testovací množina | 49,97 | 7297 | 21,17 |
| | 1. polovina května | 52,99 | 7889 | 20,60 |
| | 2. polovina února | 48,14 | 6316 | 17,45 |

## 2.6 Chyby všech modelů

Níže uvedená tabulka souhrnně ukazuje velikosti chyb všech zkoumaných modelů.

Tabulka 2.16: Celkové srovnání chybových funkcí všech modelů na celé testovací množině

| | model | MAE **[kWh]** | MSE **[$kWh^2$]** | MAO **[kWh]** |
|---|---|---|---|---|
| Modely bez znalosti počasí | *Constant_D1_linear* | 95,57 | 23015 | 49,41 |
| | *Linear_D1_day_back* | 89,13 | 19914 | 46,74 |
| | *Constant_D1_neuron* | 95,96 | 20708 | 56,81 |
| | *Neuron_D1_day_back* | 88,66 | 18689 | 46,90 |
| | *Constant_H2_linear* | 61,07 | 9906 | 31,00 |
| | *Constant_H2_neuron* | 63,12 | 10467 | 32,99 |
| | | | | |
| Nejlepší modely D1 a H2 (známe předpověď počasí) | ***Combine_D1*** | **54,25** | **7921** | **26,40** |
| | ***Combine_H2*** | **49,97** | **7297** | **21,17** |
| | | | | |
| Ostatní D1 a H2 modely se znalostí předpovědi počasí | *Linear_base_D1* | 65,85 | 10154 | 36,23 |
| | *Linear_shine_D1* | 58,76 | 8823 | 31,69 |
| | *Linear_hours_D1* | 59,43 | 9334 | 30,68 |
| | *Linear_complex_D1* | 55,42 | 8278 | 27,75 |
| | *Linear_Short_term_H2* | 51,42 | 7501 | 23,31 |
| | *Neuron_D1* | 55,27 | 8184 | 27,33 |
| | *Neuron_H2* | 51,00 | 7568 | 22,23 |

# Závěr

Cílem této práce bylo seznámení se s bayesovskou lineární regresí a bayesovskou neuronovou sítí. Tyto modely strojového učení jsem aplikoval na predikci výroby solární elektrárny. Předností bayesovské regrese je, že každý regresní parametr je funkcí hustoty pravděpodobnosti, ne jen jedno číslo. Z této hustoty lze zjistit i nejistotu celé predikce, která je taky hustotou pravděpodobnosti/posteriorní distribucí.

Tato práce byla zaměřena velmi prakticky, do první kapitoly – teoretické části – jsem napsal jen nezbytně nutnou matematiku k pochopení bayesovského učení, které jsem poté prezentoval na příkladech. V praktické části jsem se detailně věnoval předpovědím výroby elektřiny solární elektrárny, a to jak na den dopředu, kdy jsem predikoval výrobu elektřiny pro každý hodinový časový interval, tak na dvě hodiny dopředu. Předpovědi jsem dělal jak bez znalosti předpovědi počasí, tak i se znalostí počasí. Začal jsem s nejjednoduššími modely lineární regrese, které jsem postupně vylepšoval, dokud jsem nenašel dva nejpřesnější pro predikci výroby elektřiny na den dopředu a na dvě hodiny dopředu. Měl jsem k dispozici data o přímém slunečním svitu, difuzním svitu, teplotě vzduchu ve dvou metrech nad zemí, oblačnosti a celkovém úhrnu srážek. Zjistil jsem, že z dat, se kterými jsem pracoval, jsou pro předpovědi vstupy o celkovém úhrnu srážek a oblačnosti zcela nepotřebnými údaji, protože chyby na testovací množině nabývaly stejných hodnot jak s těmito vstupy, tak i bez nich. Poté jsem pokračoval bayesovskými dopřednými vícevrstvými neuronovými sítěmi, které jsem srovnával s lineárními modely. Nejzajímavější zjištění této práce je podle mě velmi podobná (téměř totožná) přesnost dvou nejlepších a nejsložitějších lineárních modelů pro $D + 1$ a $H + 2$ predikcí ve srovnání s predikcemi neuronovými sítěmi, které mají stejná vstupní data jako nejlepší lineární modely. Lineární regrese je tedy pro tento problém jako šitá na míru, protože z dat dovede určit závislosti stejně dobře jako neuronová síť. Nakonec jsem vytvořil dva kombinované modely pro $D + 1$ a $H + 2$, které vznikly spojením dvou nejlepších lineárních modelů a dvou neuronových sítí. Celkový počet modelů je 15.

Vzhledem k tomu, že jsem se modelům pro předpověď výroby solární elektrárny věnoval velmi pečlivě, nevidím už žádnou jinou možnost, jak hmatatelně (více než o 2 % z pohledu chybových funkcí) zlepšit přesnost mých nejlepších modelů.

# Literatura


[1] BISHOP, Christopher M. *Pattern recognition and machine learning*. [New York]: Springer, c2006. Information science and statistics. ISBN 978-0387-31073-2.

[2] Struktura dvojvrstvé neuronové sítě. In: *Visualizing Artificial Neural Networks (ANNs) with just One Line of Code* [online]. 2018 [cit. 2022-05-02]. Dostupné z: https://miro.medium.com/max/1400/1*gTwmrLh1aYLzayMylHGIeg.jpeg

[3] Jak Thomas Bayes vlastně odvodil ten svůj teorém?. *Centrum pro bayesovskou inferenci* [online]. 2017 [cit. 2022-05-02]. Dostupné z: https://www.4bin.org/odvozeni-bayese/

[4] Variational Inference: Bayesian Neural Networks. *While My MCMC Gently Samples* [online]. 2013, 2016-06-01 [cit. 2022-05-02]. Dostupné z: https://twiecki.io/blog/2016/06/01/bayesian-deep-learning/

[5] Bayesian Linear Regression in Python via PyMC3. *Towards data science* [online]. 2020-12-30 [cit. 2022-05-09]. Dostupné z: https://towardsdatascience.com/bayesian-linear-regression-in-python-via-pymc3-ab8c2c498211

[6] Continuous. *PyMC3* [online]. 2018 [cit. 2022-05-02]. Dostupné z: https://docs.pymc.io/en/v3/api/distributions/continuous.html#continuous

[7] Neural Networks. *IBM* [online]. 2020-08-17 [cit. 2022-05-02]. Dostupné z: https://www.ibm.com/cloud/learn/neural-networks

[8] 12 Types of Neural Network Activation Functions: How to Choose?. *V7labs* [online]. 2022-05-07 [cit. 2022-05-02]. Dostupné z: https://www.v7labs.com/blog/neural-networks-activation-functions

[9] Continuous Probability Distributions. *Duke NICHOLAS SCHOOL of the ENVIROMENT* [online]. [cit. 2022-05-02]. Dostupné z: https://sites.nicholas.duke.edu/statsreview/continuous-probability-distributions/

[10] Introduction to Bayesian Regression. *StatsWithR* [online]. [cit. 2022-05-02]. Dostupné z: https://statswithr.github.io/book/introduction-to-bayesian-regression.html

[11] YAN, Lisa. *Maximum A Posteriori* [online]. 2019-11-13, 3 [cit. 2022-05-02]. Dostupné z: https://web.stanford.edu/class/archive/cs/cs109/cs109.1202/lectureNotes/LN23_map.pdf

[12] Common Loss functions in machine learning. *Towards data science* [online]. 2018-9-2 [cit. 2022-05-02]. Dostupné z: Common Loss functions in machine learning | by Ravindra Parmar | Towards Data Science