BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INFORMATION SYSTEMS
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DATA MINING IN SMALL BUSINESS

METODY ZÍSKÁVÁNÍ ZNALOSTÍ V MALÉM PODNIKÁNÍ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **FRANTIŠEK SABOVČIK**
AUTOR PRÁCE

SUPERVISOR **Doc. Ing. JAROSLAV ZENDULKA, CSc.**
VEDOUCÍ PRÁCE

BRNO 2018

Abstract

This thesis has as an objective to evaluate techniques of data mining for use in small business. By examining data and consultations with domain experts, two approaches were chosen: market basket analysis and sales forecasting. For market basket analysis, Relim algorithm together with interestingness measurment of association rules was used. For prediction task, decompostion model, SARIMA, TSMARS and time-lagged neural network models were implemented. Acceptable results were obtained by hyper-parameter optimization. In contrast to expectation, additional weather data and use of non-linear model did not improve accuracy above SARIMA model. Forecasting was implemented as a backend service for testing in production.

Abstrakt

Tato práce si klade za cíl vyhodnotit techniky získávání znalostí pro využití v prostředí malého podnikání. Po prozkoukámí dat a konzultace s doménovými experty byly vybrány dvě úlohy: analýza nákupního košíku a predikce prodejů. Pro analýzu nákupního košíku byl využit algoritmus Relim pro vyhledávání častých itemsetů a metriky určující zajímavost asociačních pravidel. Pro úlohu predikce prodejů byl implementován dekompoziční model, SARIMA, TSMARS a neuronové sítě s časovým oknem. Modely byly vyhodnoceny. Pomocí optimalizace hyper-parametrů bylo dosaženo přijatelných výsledků. Oproti předpokládům nedošlo při dodání dat o počasí a využití nelineárních modelů ke zlepšení oproti SARIMA. Predikce byla implementována jako služba na straně serveru pro testování v produkčním prostředí.

Keywords

Data mining, small business, machine learning, regression, association rules, time series forecasting.

Klíčová slova

Metody získávání znalostí, malé podníkaní, strojové učení, regrese, asociativní pravidla, predikce časových řad.

Reference

SABOVČIK, František. *Diploma thesis.* Brno, 2018. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Doc. Ing. Jaroslav Zendulka, CSc.

Rozšířený abstrakt

V prostředí velkých společností je standardem používání metod získávání znalostí. Tyto metody jsou používány k přesnému cílení na zákanickou skupinu. Patří mezi ně clustering, klasifikace, dolování asociačních pravidel, predikce atd. V prostředí malého podníkání se tyto metody prozatím neprosadily, kvůli náročným požadavkům na technologii a lidské zdroje.

Tato práce se zaměřuje na metody získávání znalostí s cílem podpory malého podnikání v prostředí amerického trhu. Data poskytnuté pro tuto práci pocházejí z POS systému distribuovaném na americkém trhu. Cílem této práce bylo také ověřit, zda je možní některé z metod použít pro produkční nasazení v rámci daného systému. Byl implementován serverový program, který poskytuje predikční data klientské produkční aplikaci.

Dané metody musí být prezentovatelné v uživatelsky přívětivé podobě tak, aby byly snadno intepretovatelné. Také musí pracovat automaticky či poloautomaticky, aby jej mohli využívat uživatelé bez hlubších statistických znalostí.

Tato práce má se skládá z části průzkumné a části implementační. V prvé části jsou prozkoumány obecné metody získávání znalostí s cílem vybrat potencílní úkoly pro následnou implementaci. Dále je provedena analýza konkurečních produktů. V následující části jsou analyzovány vhodné nástroje a struktura poskytnutých datových zdrojů. V poslední části jsou dané úkoly implementovány a vyhodnoceny.

Vybrány byly dva úkoly – analýza nákupního košíku a predikce prodejů. První slouží k analýze chování zákazníků s cílem přizpůsobit nabídku potřebám zákazníků. Druhý úkol má za cíl předpovídat budoucí vytíženost podniku, aby bylo možné lépe plánovat organizaci pracovníků.

Analýza nákupního košíku je provedena pomocí vyhledání částých itemsetů pomocí algoritmu Relim, které jsou následně vyhodnoceny podle různých metrik zajímavosti a prezentovány.

Část predikce nejprve zkoumá teorii k předpovídání časových řad, kterou dále využívá k analýze dostupných dat. Z nich vyplývá silná týdenní periodicita, kterou je možné využít k předpovídání budoucích dějů. Na základě vhodných dat je dále zkoumáno několik modelů, jedná se o jednoduchý dekompoziční model, model SARIMA, TSMARS a neuronové sítě.

Pro vyhodnocenování daných modelů jsou vybrány především chyby měření MAPE (mean absolute percentage error), MAE (mean absolute error) a MASE (mean absolute scaled error). MAE a MASE mají vhodnější teoretické vlastnosti, ovšem jsou méně dobře interpretovatelné.

Jednoduchý dekompoziční model nedosahuje uspokojivých výsledků. Ostatní modely dosahují navzájem podobných výsledků, které lze považovat za dobré. Byl zkoumán vliv dat o počasí. Bylo provedeno optimalizování modelů TSMARS a neuronových sítí, s cílem přízpůsobit model těmto datům, data ovšem nezlepšily daný výsledek.

Kromě vyhodnocení daných prostředků je výsledkem této práce implementace modelů v programovacím jazyce Python, který je kromě vyhodnocování daných modelů využit jako program na straně serveru poskytující data produkční aplikaci.

Diploma thesis

Declaration

Hereby I declare that this bachelor's thesis was prepared as an original author's work under the supervision of Doc. Ing. Jaroslav Zendulka, CSc. All the relevant information sources, which were used during preparation of this thesis, are properly cited and included in the list of references.

.....
František Sabovčik
May 25, 2018

Contents

Chapter 1

Introduction

Traditionally big companies used techniques of Data Mining for improving their sales performance by better targeting on customers by analyzing their behavior based on techniques of clustering, classification, association rules mining, prediction etc.

Although of existence of these methods, small businesses not used these techniques due to lack of knowledge and affordable technology. This thesis is focusing on use of Data Mining techniques to support small businesses as they face new challenges in today economy of United States.

Small business is defined by *Small Business Administration (SBA)* as a business with less than 500 employees [7]. Another definition is "an independently owned and operated company that is limited in size and in revenue depending on the industry." [8]. In context of this work, term small business is used for type of business represented by cafés, bars, restaurants, private doctors, retail stores, dry cleaners etc.

The Point of Sale is a logical (possibly physical) point in sale process, where a retail transaction is completed, purchased items are counted, the price is calculated and a customer is asked to make the payment, also an invoice for a customer can be provided.

POS systems are computer systems, which enable this process in a more automatic way. Particularly outside of the retail industry, POS systems developed into more powerful tools providing more features, covering stock management, orders, customer relations, and for context this theses most importantly, offering reporting capabilities. Reports allows an owner to analyze performance of particular components of his business, providing a tool to improve it's efficiency by performing an appropriate actions.

1.1 Objectives

Use of a POS systems to manage different aspects of a small business generates data possibly suitable for data mining techniques, which would produce a useful knowledge for the business owners.

For every small business, particularly ones starting, it is crucial to make proper decisions facing a competition, an economic situation and customer demands. The objective of this thesis is to support decision making of a manager by extracting a useful knowledge and presenting it in a user-friendly way. Also, this system has to be automatic or at least semi-automatic, since it is intended to be part of the POS system and used by users without any deeper statistical knowledge.

The method of solving the objective will be done in these steps:

1. The research of *Data Mining techniques*, their *use cases* and proposing potential solutions based on the problem domain knowledge
2. The analysis of existing products for *decision support* of small businesses, assessing their capabilities in relation with the previous step
3. Performing a sustainability analysis of possible solutions
4. For every chosen task, a deeper research of particular techniques and performing data mining on a given dataset, resulting in implementation of a solution and evaluating the results

1.2 Structure

In Chapter 2, general methods of data mining are researched to choose methods suitable for small business environment. Consequently (Chapter 3), existing solutions for small business are examined. In Chapter 4 and 5 available data source is examined and data mining tools are chosen. Consequently, in 6, appropriate tasks to achieve are selected. Chapters 7 and 8 are each dedicated to research and implement one of the chosen tasks.

Chapter 2

Data Mining Methods

Data mining is process of automatically discovering useful information in large data repositories, it is used for reveling novel, previously unknown patterns [9]. It relies on the intersection of methods from machine learning, statistics, and database systems. Data mining is an integral part of knowledge discovery in databases (KDD), Figure 2.1.

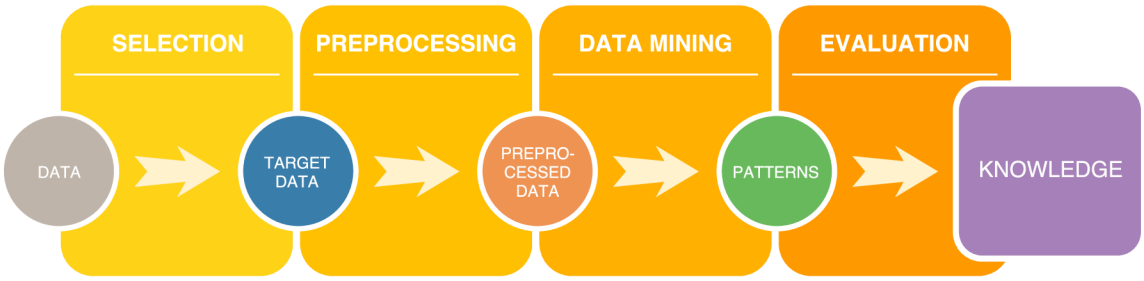


Figure 2.1: Knowledge discovery in databases

There are many techniques used in data mining for particular business problems, this section is providing overview of the methods, some of which can be potentially used for archiving given goals. We can divide these methods into these categories:

- Clustering
- Association Rule Mining
- Prediction
- Anomaly Detection
- Dimensionality Reduction

2.1 Clustering

Method used for dividing instanced of data into two or more groups, called clusters, based on similarity between members of particular clusters. These methods are useful in variety of fields, such as social sciences, biology, statistics or pattern recognition [9, pp. 487-491].

There are two categories of output information gained from clustering:

1. **Clustering for Understanding** - an application, which can comprehend the reality in a structured way, which allows to understand it in a better way. On of the application is biology when it is possible to generate taxonomy of a living organism in automatic way. Other example is use in business, where it is useful to divide

customers into groups, which can be approached individually and target marketing activities specifically.

2. **Clustering for Utility** - clusters represents an abstract data class, which can be useful for increased performance of further process, e.g. when finding nearest neighbors in large dataset, first clustering into subgroups and finding neighbors among them can be beneficial

Among techniques used for the cluster analysis are mentionable:

- **K-Means** - partitions n observations into k clusters when distance of every observation to it's partition centroid is minimal. The task is NP-hard, the approximation for this problem is Loyd's algorithm, which iteratively assigns observation to nearest clusters based on preselected distance measure and then recomputes partition's centroids. Downside of this method is the requirement to specify k parameter.
- **Agglomerative Hierarchical Clustering** - group of techniques merging iteratively nearest clusters and producing hierarchical graph, in which is visible every iteration with the corresponding number of clusters. Some of the methods have interpretation in graph-based clustering, some have interpretation in prototype-based interpretation. The advantage over K-Means is that it is not required to define k parameter, downside is whole clusters have to be merged.
- **DBSCAN** - density based algorithm taking two parameters ϵ and $minPts$. For every unprocessed point, number of points in his ϵ distance is considered, when it exceeds $minPts$, it is considered as part of dense cluster, then for every point in ϵ distance is done the same, points not satisfying this condition are marked as noise and thus not belonging to any cluster. Main advantage of the algorithm is to generate non-convex clusters (Figure 2.2) , it is not needed to define k parameter and also, performance can be facilitated by use of R* trees. Disadvantage is non-determinism, the output of the algorithm can differ according to the starting point.

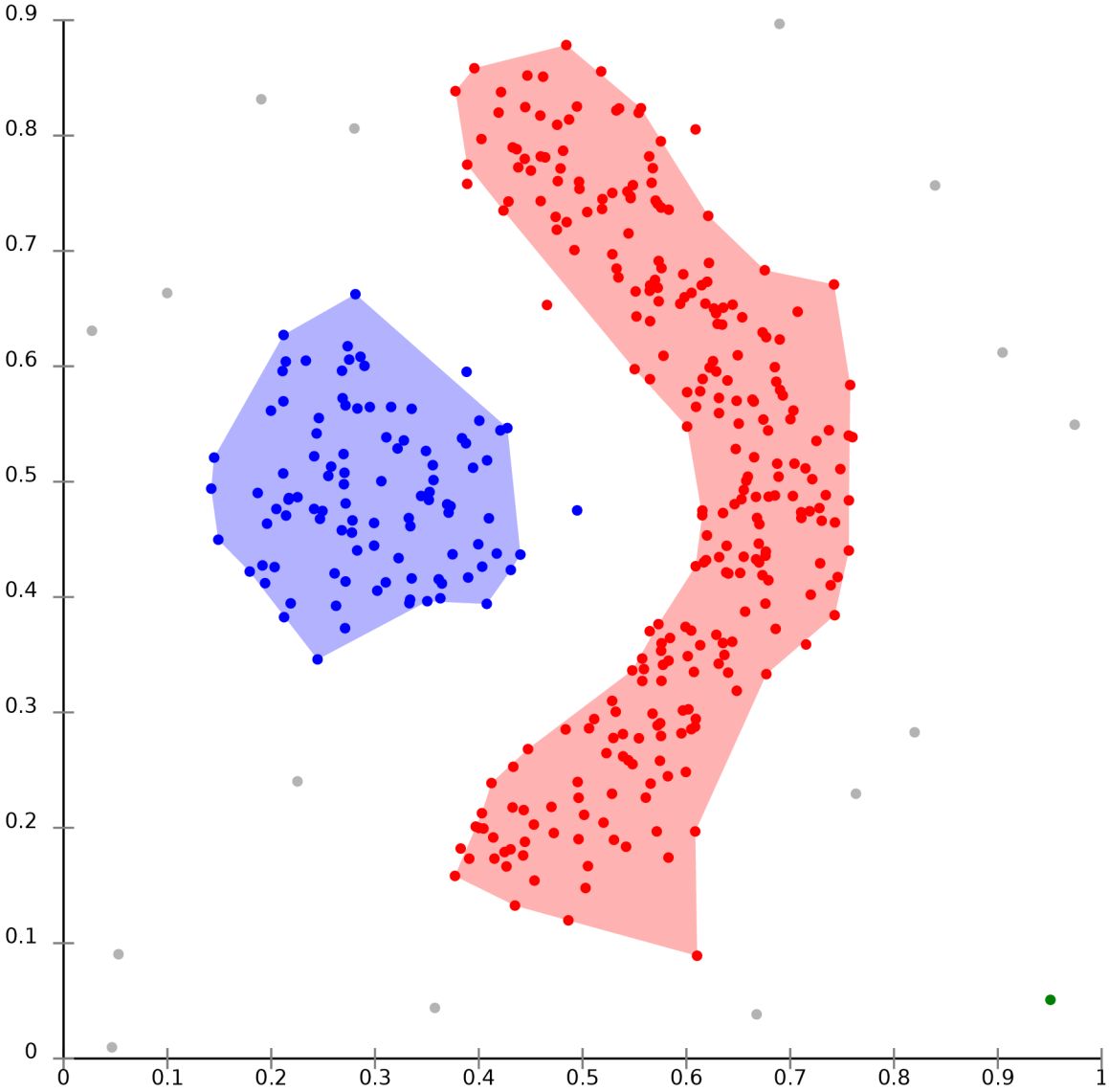


Figure 2.2: Clustering using DBSCAN

2.2 Association rules

The method used in the basket analysis to find the patterns in items purchased among transactions by generating rules in form of $A \rightarrow B$, where A and B are item sets. The rule generally states that there is an implication between the antecedent and the consequent. Many metrics exists to describe "goodness" of a rule based on the desired type of information and the nature of the data.

The result of the association rules analysis is used by retail stored for promotional pricing or product placements. Also, association rules can be used in other fields like Web usage mining, intrusion detection, continuous production, and bioinformatics.

2.3 Predictive Analysis

A prediction is a statement about an uncertain event, often occurring in the future. A related term forecasting is often used interchangeably, the alternative definition describes forecasting as a subset of prediction, producing estimation for every step in the time range, rather than generally event that can occur any time. When predicting continuous variable, the term regression is used, predicting categorical values is called classification. [10]

The prediction is one of the main domains of data mining, it's applications can be found in many fields, including:

- optimization of shipping cost by predicting orders [12]
- estimate credibility of a client asking for loan
- stock market forecast
- causal relationships between parameters in biological systems
- causes of diseases in epidemiology
- analyzing effects of pollution in environmental science

The prediction and the classification are forms of the supervised learning, for a model to learn it is necessary to provide training data, which are structured in series of observations, each observation contain values of independent variables (input) and dependent variable (output), then model is capable of providing expected output for provided unseen input.

The methods for predicting the continuous output variable includes for example **linear regression**, **polynomial regression** or other types of regression. The forecasting of the time series includes **Kalman filtering, Exponential smoothing, Autoregressive integrated moving average (ARIMA), Trend estimation based models or STL.**

For the classification are used e.g. these algorithms: **Logistic regression, Naive Bayes classifier, Quadratic classifiers, Decision trees or Boosting**.

More advances methods like **Support Vector machines** or **Artificial Neural Networks (ANN)** can be used for both cases. There is a particular class of ANN called **Recurrent Neural Networks**, e.g. **Long Short Term Memory Networks** that are intended for Time Series Forecasting.

2.4 Anomaly Detection

The field dedicated to uncover unusual observations, often for security reasons, but in general any unusual data can be detected. The anomaly detection (or outlier detection) is closely related to the clustering, differences between the two is that anomaly detection is focusing rather on observations to differing from the rest [10, pp. 544-553].

The distinction has to be made between noise and outliers, the noise is generally an ubiquitous phenomenon in data modeling, it is usually ignored and removed from the model. Outliers are generated by another process, distinct from the original one, thus to detect outliers, it is necessary to define properly the original process.

We can divide anomaly detection techniques according to whether expert provided information is available. When there is expert provided information we talk about **supervised methods**, when there is not, we talk about **unsupervised methods**. Also it is possible to use **semi-supervised** methods for labeling some observations and then complete the process by labeling similar observations to those labeled by expert by some proximity metrics.

Second criterion is to divide anomaly detection according to the method used to process a particular observation:

1. **Statistical Methods** relay on defining statistical model of "normal" observation with appropriate distribution, subsequently test observation are evaluated according to this model. If they manifest statistically significant deviation from this model, it is categorized as an anomaly.
2. **Proximity based methods** assume that anomalies has it's neighbors far away in feature space, in different words, the proximity of anomalies significantly deviates from proximity of other objects. The difficulty of this methods consists in use of appropriate proximity measure, which can be hard to archive in some datasets. Also, in some cases, when many anomalies are clustered, they can be overlooked.
3. **Clustering-Based Methods** assumes that "normal" observations belongs to large and dense clusters, whereas anomalies are found apart from them. There are many supervised and unsupervised algorithms for clusterings (see above). Downside of this approach is computational cost.

2.5 Proposals of application for given problem

By research of common data mining tasks done previously, concrete solutions leading to a practical advantage for the client can be proposed:

- **Personalized recommendation** of complementary products to motivate customer to purchase related items and therefore increase sales.
- **Associated products** analysis with output used for recommendation of complementary products, price adjustment or enhancing a menu by grouping related items
- **Forecasting of sales** in a particular day, month or a year the period for stocking up and for special offers
- **Clustering customers** by their typical orders and thus dividing customers into certain categories and targeting these groups individually
- **Anomalies in sales** of particular product, which can lead to uncovering hidden problems, or to validate special pricing. Possible extension is to incorporate more variable into the process, resulting in explanatory approach to an anomaly, e.g. location or day time.

Chapter 3

Competition Analysis

Research of competition in regards of business analysis will be done in two different types of software:

1. Point of Sales systems
2. Business Intelligence tools

Both groups will be evaluated and software solutions will be investigated in regards of support for small businesses.

Square

Currently the most popular POS with over 2 millions of users [2]. It has report capabilities limited to charts displaying sales over particular period of time (day, week, month etc.). User can filter the data by custom defined filters. System offers more deep insight into actual performance by comparing sales during actual period with according past period (day week ago, day year ago etc.).

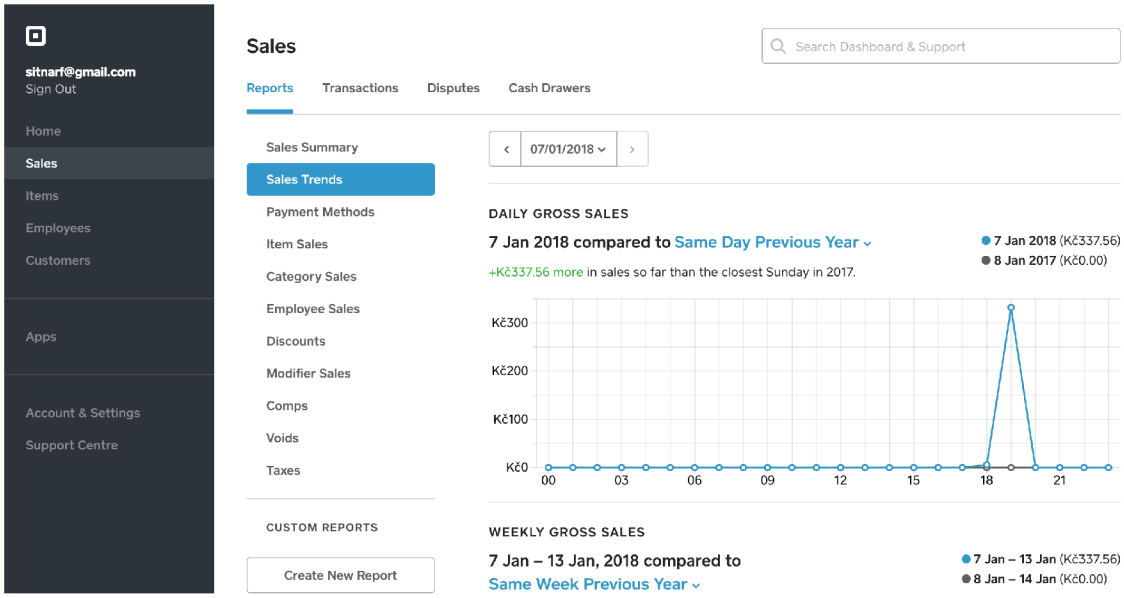


Figure 3.1: Square: Trend visualization

Shopify

System with almost 2 millions of users [2], although only $\frac{1}{3}$ paying customers compared to Square. It provides mostly bar and line charts with different metrics, useful for small businesses are mostly sale charts over period of time. Every chart is customizable with custom filters and with possibility do add another data point to the chart (e.g. location).

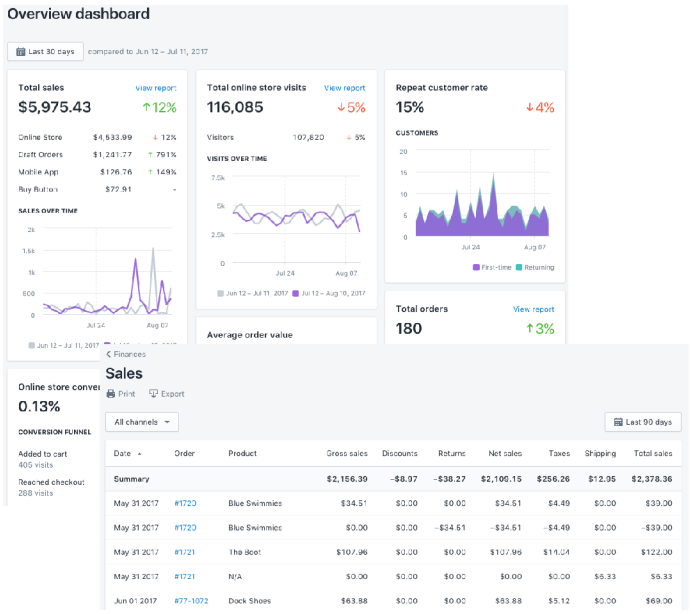


Figure 3.2: Shopify: Visualisations

Sisense

Company with thousands of customers, some of them are better-established companies as Airbus, Wix or Hewlet Packard, other small startups

It allows in very user-friendly way to easily import and clean data, create charts and reports, and present the results. Programming is not needed, although for more advanced analytics like regression, there is a possibility to use R function in equations.

There is also possibility to extend the functionality by writing JavaScript plugins, which can define new types of widgets or customize visualization of dashboards.

Also, related product Sisense Pulse allows to detect anomalies by settings threshold or by use of artificial intelligence.

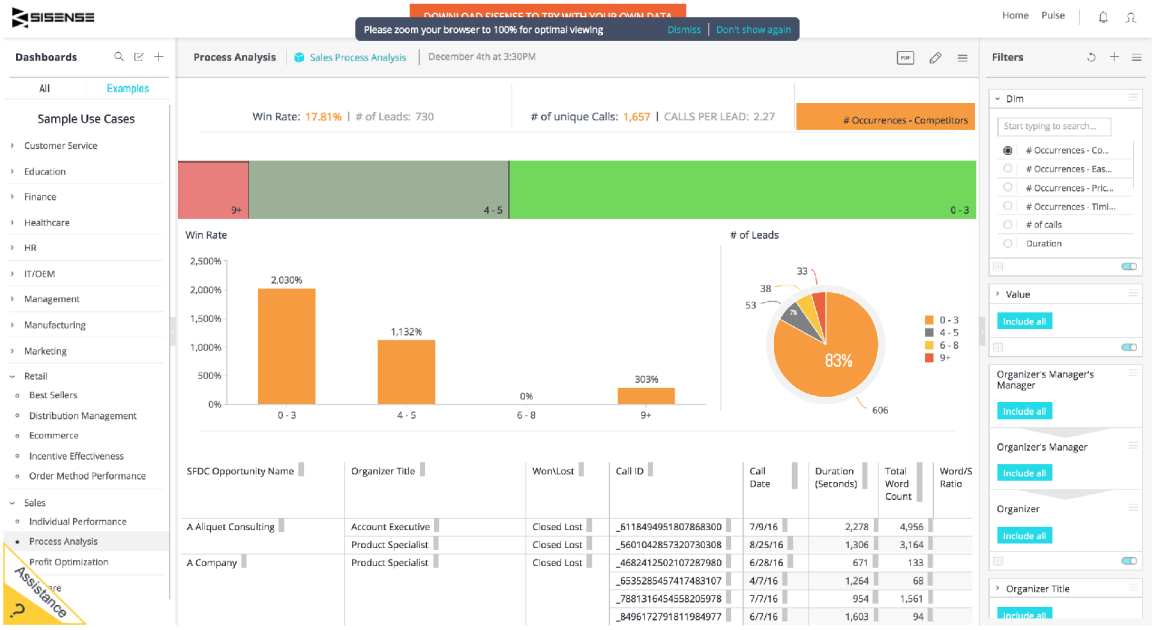


Figure 3.3: Output dashboard

Figure 3.4: Edit mode for data visualization

Conclusion

By research of the products targeting similar goals, it can be said, that exists two disjunct set of applications. POS systems offers very basic, yet simple to use reports, which includes visualization of sales related to some other criteria (e.g. time or product category).

Second group, business intelligence offer more advanced methods for analyzing and visualizing data, their disadvantage is that use of these tools require theoretical and practical knowledge of analysis, making it difficult for unskilled user.

Conclusion of this section is, that solution offering tools that are part of BI, but doing so in automatic way, as part of product user already use, could be providing functionality so far missing on the market.

Chapter 4

Datasets

Sales data collected by POS was provided, it contains sales data from several clients, the data time range is from several weeks to years.

Data are stored in the PostgreSQL 10.x database system, were anonymized for purpose of research and downloaded into a local database system.

4.1 Database Structure

For better analysis of important data, structure is stripped of unnecessary data. These are primarily ones designated for management of user accounts, taxes etc.

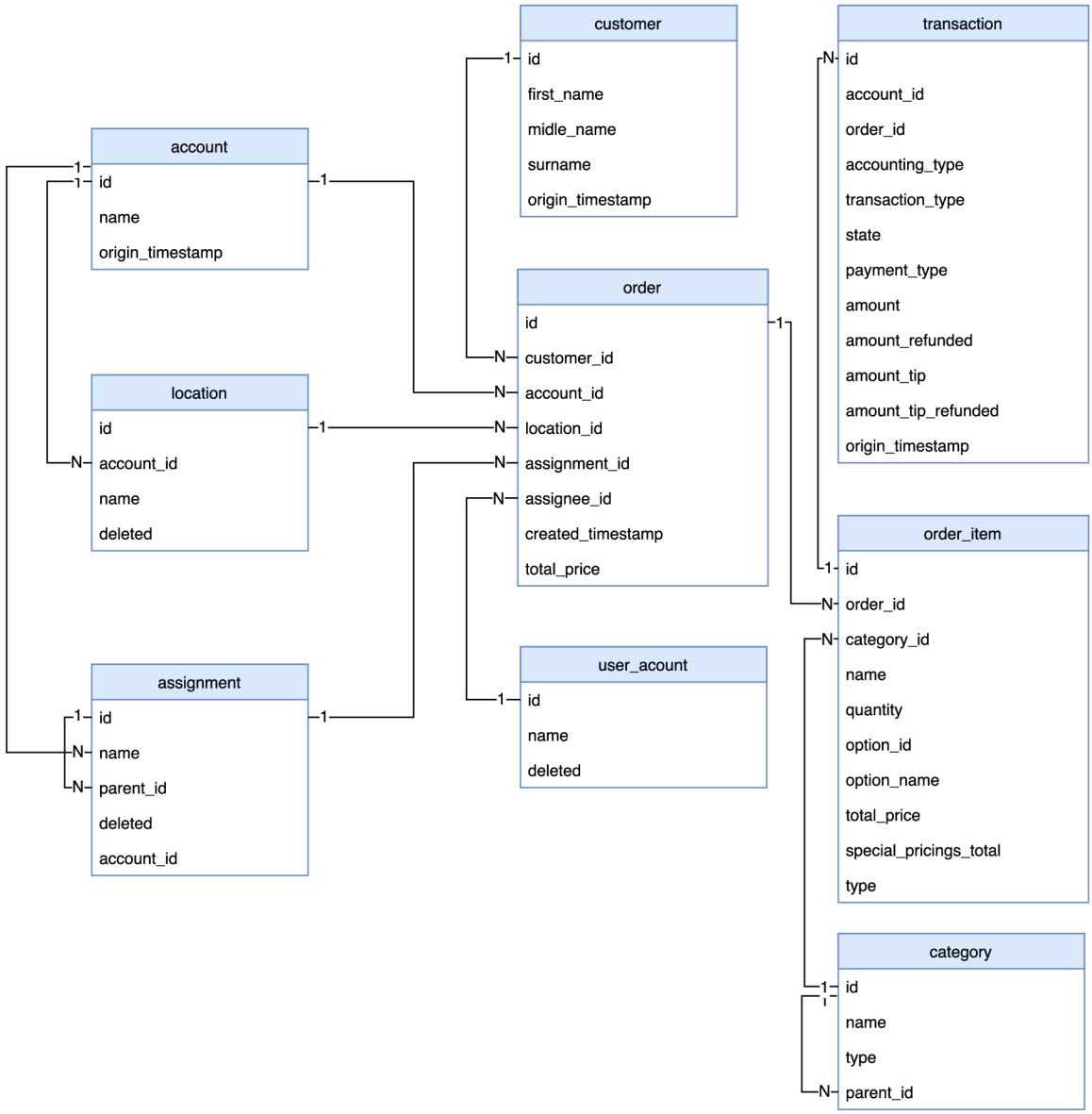


Figure 4.1: Database diagram

Every business is represented by the `account` entity. Tables are shared by all accounts, `account_id` column is distinguishing particular business data from others.

A central entity is `order` representing order containing one or more `order_items`, it is assigned to `customer` purchasing set of items and `assignee` responsible for delivering the order, e.g. server in a restaurant. Customer data are fetched from credit card payment or can be entered manually. When paying in cash, customer data can be missing.

`category` in this data model defines both categories containing products (e.g. Meals, Drink, Grill) and definition of products (e.g. Cola, Rice, Cappuccino).

Particular order items listed on the order are are very customizable, there can be some special pricing or price alone can be entered completely apart of definition of product.

Chapter 5

Data Mining Tools

In this section are analyzed tools used for data mining to perform initial part of the process which consists of analyzing the data and finding potential solutions to incorporated into final product.

Ideally same tool set would be suitable for production deployment producing knowledge automatically as a service. Because of the previous point, tools offering commercial license, would be preferred, also price criterion is considered. Another important criterion is availability of documentation and community support for given platform.

Data analysis is dominated by three programming languages: **R**, **Java** and **Python** [3] [14]. Although there are other choices, for the sake of amount of information sources, community support and robust libraries, these three were considered.

Very popular tool for data analysis is **R**, it contains variety of functions for wide range of statistical and data mining applications, also containing integrated graphical environments like RStudio. Most important downside according to chosen criteria is lack of R capability to run as service, existing Microsoft Machine Learning Server is non-multiplatform solution requiring purchase of Microsoft SQL Server.

Because server-side of Figure POS is written in Java, it would be the first chose for simple interoperability.

Weka is one of the most popular Java Data Mining libraries, it is developed at University of Waikato in New Zealand. It contains tools for data mining, data analysis,predictive modeling, preprocessing, classification, clustering, visualization, regression, and feature selection. Weka is released under GPL license, therefore not suitable for commercial use, commercial license is somehow available, but purchasing process is unclear [11].

RapidMiner is popular tool composed of RapidMiner Studio allowing visual creation of model, RapidMiner Server for collaboration and RapidMiner Radoop for parallel computations over Spark and Hadoop. Most prevalent downside for use for this application is price, which is for commercial use in thousand of dollars.

Apache Mahout, maintained by Apache Foundation is promising fast GPU-powered computations, but amount of provided algorithms is very humble [13]. **JDMP** is library possible use for commercial use, but it's website states it is not finished and not well documented [5]. Another libraries (**Deeplearning4j**) is suited only for machine learning, other (**ELKI**, **SPMF**) are distributed under Free Software licenses not suitable for commercialization.

Third option, **Python**, has community support and well documented libraries suited for different types of analysis. There is no single tool or library incorporating all functionality,

instead of that there is range of libraries dealing with particular task in process of knowledge discovery.

Core tool for Data Mining in python is **SciPy stack**, consisting of:

1. **NumPy** adds support for large multi-dimensional arrays and matrices in efficient manner and providing large collection of mathematical function over these structures. Decrease in space and time complexity compared to native python implementation is archived by storing data as byte array with items of same byte length rather than high-level objects containing additional overhead and operating upon them in byte-wise manner [15].
2. **Matplotlib** is library for creating graphs providing API for several GUI toolkits (Tkinter, wxPython, Qt, or GTK+), it has also procedural pylab interface.
3. **SymPy** provides functions for symbolical computations.
4. **Pandas** contains data structures for numerical tables and time series, as well as operation structuring the data. It allows loading into in-memory representation from different file formats (csv, sql, HDF5 etc.)
5. **SciPy library** provides many user-friendly and efficient numerical routines such as routines for numerical integration and optimization.

For use of Machine Learning, most prominent library is **scikit-learn** depending on SciPy stack, offering algorithms for in preprocessing, model selection, dimensionality reduction, classification, regression and clustering.

Keras is modular library intended for use of deep learning with Neural Networks, it is one of the fastest growing libraries [16]. It is written in pure python, but for computational intensive functions relies on MXNet, Deeplearning4j, TensorFlow, Microsoft Cognitive Toolkit or Theano backends [17].

According to previously set criteria, **Python** was chosen for further use for it's well-established libraries, community support and good documentation, also it is possible to use it on server in easy way. Consequent reason is also that language itself, as well as provided libraries, is suitable for commercial use.

Chapter 6

Initial analysis and task choice

In this section, proposed Data Mining application will be evaluated to choose appropriate tasks for further analysis and potential consequential implementation.

Name	Order count
Korean Restaurant	76720
Nail Saloon	26779
Deli	15882
Italian Ice Cream	10883
Sushi Restaurant	10619
Café	5149
Dry Cleaners	5083
Bar	1639
Thai Restaurant	1612
Optics	912

6.1 Customer preferences

This section is aiming to analyze possibility to extract customer preferences to obtain interesting patterns describing customer behavior. This type of knowledge describes several previously suggested tasks:

- Associated products
- Clustering customers
- Personalized recommendation

This kind of task uses information about what customers purchase during some period of time, in first case these purchases are aggregated per order, in second customers are clustered into groups and in third purchase habits are focused on particular customer.

When examining particular datasets, important *problem was discovered* complicating this kind analysis, in current data structure, there is not distinction between items ordered by different customers, only single reference to `user_account` is created for customer performing payment for whole order, therefore one order contains several customer order, see Table 6.1, where it is possible to see several products of same category (e.g. coffee) in one order, indicating multiple customers.

Divide of order into suborder for particular customers will be one of the features implemented in the POS in future. In the meantime, further investigation whether is possible

Table 6.1: Example of Transaction of Café dataset

Tea
Tea | Americano
Waffles | Coffee | Eggs Special
Bubble Tea
Bubble Tea | Waffles | Pastry | Pastry | Cappucino
Cappucino | Cappucino
Waffles | Latte | Latte | Eggs Special
Latte | Fried Dumplings | Rice Dish | Egg Sandwich
Bubble Tea
Coffee | Hot Chocolate | Espresso | Waffles | Omelette | Egg Sandwich
Egg Sandwich
Omelette | Coffee | Coffee | Waffles | Custom +1
Burrito | Rice Dish | Coffee
Coffee | Tea
Omelette | Coffee | Egg Sandwich | Omelette | Omelette
Orange Juice | Egg Sandwich
Pastry | Pastry | Pastry
Latte | Egg Sandwich | Egg Sandwich | Egg Sandwich | Pastry | Coffee
Latte

to circumvent this problem is needed. Also, analysis of orders of Nail Salon shows that customers share their order very rarely, thus it is suitable for this kind of analysis.

Because of problematic data structure, only **association rules analysis** is suggested for further investigation. Association rules between purchased product provides information about associated products, which can be used to choose products for special offers or organize menu and shop board.

6.2 Sales forecasting

Forecast of future sales is beneficial for business owner, since he can more efficiently plan stock management, human resources or make better decision in general.

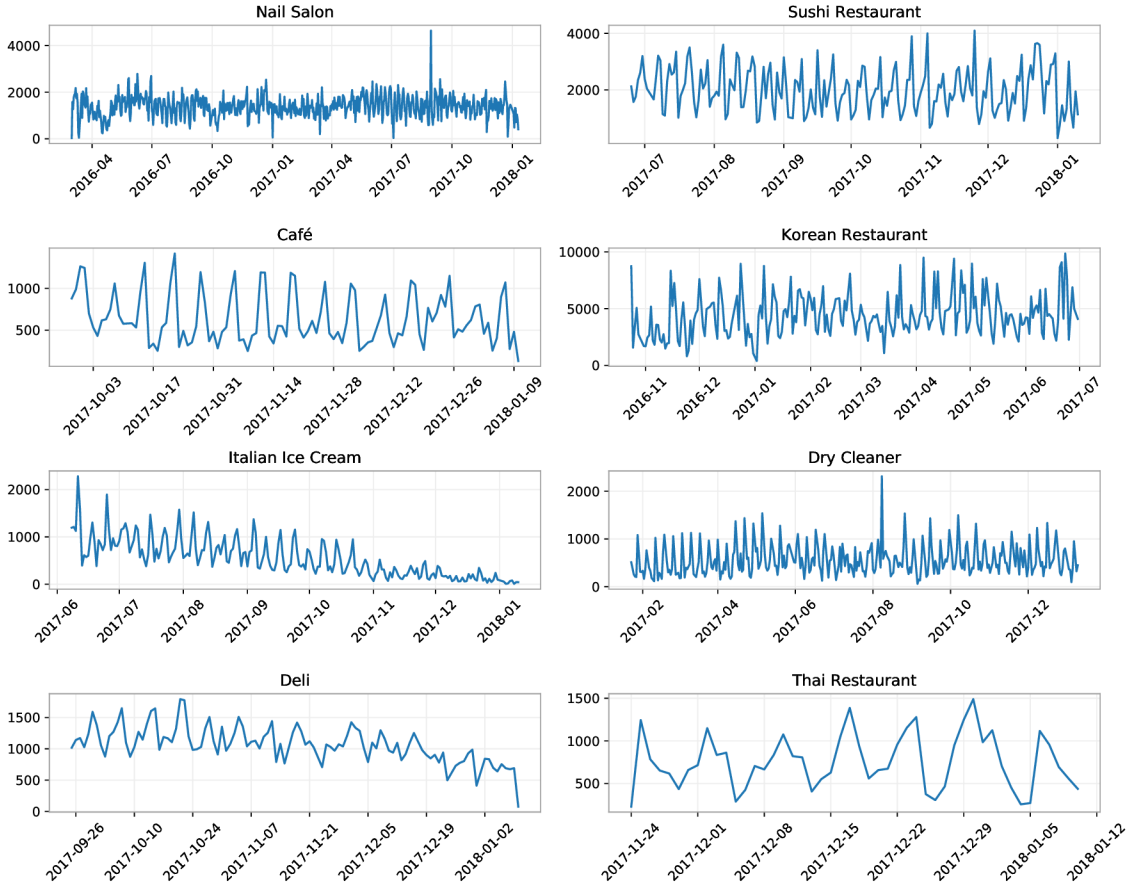


Figure 6.1: Time series of given datasets: Time / $ Revenue

Some of the sales time series are available in Figure 6.1. Time period of collected data ranges from several month to approximately two years (Table 6.2).

Table 6.2: Time period of collected data

Data set	Start date	End date	Months
Nail Salon	2016-03-01	2018-01-10	22
Dry Cleaner	2017-01-23	2018-01-09	11
Korean Restaurant	2016-10-24	2017-06-30	8
Gelato	2017-06-08	2018-01-10	7
Sushi Restaurant	2017-06-25	2018-01-10	6
Café	2017-09-28	2018-01-10	3
Deli	2017-09-25	2018-01-10	3
Thai Restaurant	2017-11-24	2018-01-10	1

Chapter 7

Market basket analysis

Task elaborated in this section aims to provide manager of the business with information about interesting patterns in purchasing behavior of customers. Behavior is inferred from combination of products being frequently purchased in customer transactions. This knowledge can be used for example to better organize menu or perform special pricing for given combinations of associated products.

Market basket analysis is commonly used in big retail industry to optimize business offer to increase sales. Hypothesis for this section is that practice, which is common in big industry, is suitable also for small business environment. To test this hypothesis, algorithms for computing frequent itemsets and association rules are evaluated.

7.1 Pattern mining

This section utilizes information from book Frequent Pattern Mining [33]. First, transaction and itemset has to be defined to further define frequent itemset and association rules. Let $\mathcal{T} = \{T_1, T_2, \dots T_n\}$ be transaction database, representing database of orders. Each $T_i \in \mathcal{T}_i, \forall i = \{1 \dots n\}$ consists of set of items $T_i = \{x_1, x_2, x_3 \dots x_l\}$. A set $P \subseteq T_i$ is called an itemset. **Support** of an itemset (Equation 7.1) is occurrence of this itemset among transactions, relative to number of all transactions. Itemset P is considered **frequent**, when $\text{supp}(P) \geq q$, where q is minimum threshold.

$$\text{supp}(P) = \frac{|\{t \in T; P \subseteq t\}|}{|T|} \tag{7.1}$$

Based on support of given itemsets, association rules, describing relation between datasets, can be defined. **Confidence** (Equation 7.2) represents conditional probability, that itemset X is present under condition, that itemset Y is present. In other words, it describes how many transaction containing X, also contain Y.

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} \tag{7.2}$$

Confidence does not take into aspect of number of transactions when Y is present, but X is not. For that reason, another measure was invented for purpose of this work. Let it be denoted as **Symmetric confidence**. It is representing ratio between amount of transactions, where X and Y co-appeared to amount of transactions, where appeared at least one of the pair. Equation is multiplied by two to normalize the value to be in approximately

same dimension as confidence.

$$\text{sconf}(X \Rightarrow Y) = \frac{2 \cdot \text{supp}(X \cup Y)}{\text{supp}(X) + \text{supp}(Y) - \text{supp}(X \cap Y)} \tag{7.3}$$

Class of static background models, which introduce notion of expected measurement and relates observed measurement to this expectation. Expectation are most often calculated based on some probabilistic model, which arises from background knowledge of the problem.

$$\text{ind(X, Y)} = \text{supp}(X) \times \text{supp}(Y) \tag{7.4}$$

Independence model (Equation 7.4) assumes that two items are independent and their co-appearance will be proportional to their occurrence in whole dataset. Using Independence model as expectation lead to measurement called **lift** (Equation 7.5).

$$\text{lift}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{ind(X, Y)}} \tag{7.5}$$

Another approach to the interestingness of an itemset is to use a statistical test. Probability of generating itemset X under independent model is $q = \text{ind}(X)$. Denote, that Z is random variable stating in how many transactions X occurs. Binomial distribution then describes probability of given number of transaction M containing X (Equation 7.6).

$$P(Z = M) = \binom{N}{K} q^M (1-q)^{N-M} \tag{7.6}$$

Statistical test for significance of M is then

$$P(Z \geq n \text{ supp}(X)), \tag{7.7}$$

which is used to compute p-value during evaluation. Algorithms for finding frequent itemsets use principle of mononicity, which states, that support of itemset, when adding new items is non-increasing, thus support can only remain same or decrease. Apriori was first algorithm that exploited this principle. The algorithm iteratively generates the candidate itemsets from the already existing ones, but when encounters an itemset, which is already not frequent, new itemsets are not generated from it [34].

For further investigation, **Recursive Elimination** (**RELIM**) algorithm for mining frequent itemsets was chosen. It is very quick and simple algorithm, which is not requiring generating candidate itemsets. First it removes items, which itself does not meet minimal support condition. Then, least frequent item is chosen.

With this item, only transaction containing this item are considered. The item is removed from them and they are considered as a new transaction set on which is recursively applied this procedure. When a recursive sub-branch is evaluated, second least-frequent item is taken for the procedure, but with already processed item kept removed. This procedure generates itemsets to check for minimal support, but also prevents from evaluating a priori known unfrequent itemsets [35].

7.1.1 Implementation

Research on existing libraries suitable for frequent pattern mining were performed and Python library **pymining**, containing implementation of Relim, was chosen. First, positive

association are going to be extracted. Relim is used to mine frequent itemsets with minimal support s_{min}. Consequently, lift, symetric confidence and p-value for binomial distribution is computed. Itemsets in following tables are represented simply as list, because all used metrics are symmetrical.

Table 7.1: Preliminary result for Deli, $s_{min} = 5$

items	p-value	lift	sym. confidence	rel. supp	supp
Burger, Basic Burger, 3. Potato	3.70574e-21	5084.07	45.0%	0.5%	6
Wheat, Coffee, 09. Peanut Butter, Bagel	8.29859e-17	956.921	2.7%	0.5%	6
Egg And Cheese, Default, Egg Whites	6.0197e-14	843.58	15.8%	0.4%	5
Wheat, 09. Peanut Butter, Bagel	7.13376e-18	580.84	10.7%	0.5%	7
Egg And Cheese, Default, On a croissant	3.77161e-15	506.148	18.0%	0.5%	6
Egg And Cheese, Default, Bagel, ...	1.6151e-12	436.519	6.0%	0.4%	5

From preliminary results (Table 7.1) is clear, that settings of s_{min} is crucial. Itemsets with low support can achieve very high lift, due to random fluctuations. Sufficient amount samples of given dataset has to be provided to be statistically significant.

Suggested heuristic adjusts trade-off between support and lift by defining minimal lift ($\text{lift}_{min} = 1.1$). For this filtered dataset, n itemsets with maximal support is taken and are consequently sorted by lift (see Table 7.2).

This approach will not achieve the maximal lift, but will provide a sufficient confidence in obtained results. Also, generic items (like "Extra + 2"), were omitted, since they do not represent real products.

Table 7.2: Result for Café 2, lift = 1.1, ordered by support

items	p-value	lift	sym. confidence	rel. supp	supp
Sweet Soy Singer Chicken, Rice - Brown	6.1147e-08	2.40915	56.6%	13.6%	45
Sunny Egg, Rice - Brown	3.85977e-06	2.10435	52.7%	13.3%	44
Sunny Egg, 1/2 Avocado	6.8775e-09	2.66479	58.9%	13.0%	43
1/2 Avocado, Rice - Brown	1.21691e-05	2.07134	50.3%	12.4%	41
Sweet Soy Singer Chicken, Sunny Egg	3.56561e-08	2.62687	56.3%	12.1%	40
Sunny Egg, Rice - White	1.36465e-07	2.57231	54.3%	11.5%	38
Sunny Egg, Chili Honey Pork	6.1473e-07	2.57931	51.1%	10.3%	34
Rice - White, 1/2 Avocado	2.86185e-06	2.4312	50.0%	10.3%	34
Sweet Soy Singer Chicken, 1/2 Avocado	4.6674e-06	2.35863	49.3%	10.3%	34
Chili Honey Pork, Rice - Brown	0.000161611	2.04085	44.0%	10.0%	33
Rice - White, Chili Honey Pork	2.30608e-07	2.80106	52.0%	9.7%	32
Shark Burger, Standard	1.03456e-11	4.71429	58.6%	8.8%	29
1/2 Avocado, Chili Honey Pork	3.21526e-05	2.32394	45.0%	8.8%	29
Tygerchycken Sandwich, Standard	2.38666e-11	4.71429	57.1%	8.5%	28
Sweet Soy Singer Chicken, Rice - White	0.00107254	1.97015	39.4%	7.9%	26

7.2 Conclusion

By investigation of the task, it seems, that market basket analysis could be indeed potentially useful for small business owners and managers to discover interesting patterns in the customer behavior. Nevertheless, for final conclusion, it is needed to validated the feature

by users of the system. During the implementation emerged two problems, that have to be addressed.

First is to prepare algorithm to automatically remove helper products, which would clutter resulting output. Second, question of statistical importance of given association risen, which also relates to choice of minimum support. Possible solution is suggested in [36], where parametric multi-hypothesis test is used, employing approximation of the Poisson distribution to choose statistically significant s_{min}.

Chapter 8

Sales prediction

8.1 Methodology

According to standard approach to time series modeling [1, pp. 21], modeling can be decomposed into these basic steps:

1. **Problem definition:** The step involves initial question for purpose of the task, it involves interviewing participants, what is their intended use of the model, how it will be used, but also how it will be integrated into the existing infrastructure of the company or product.
2. **Gathering information:** There are two types of accessible information: (a) raw statistical data represented by table structure or any other structured information, containing numerical or nominal data and (b) knowledge from domain experts, reveling hidden relations between data, which can be incorporated into the model
3. **Preliminary (exploratory) analysis:** Involves different types of statistical tests and data visualization with aim to discover valuable information for use of model selection, choice of parameters etc. It is typically focused, e.g. on discovering seasonal components, outliers or correlations between variables. Also, domain expert insight on particular phenomena can be beneficial.
4. **Choosing and fitting the model:** Best model for given data depends on many factors, availability of the historical data, character of the time series and explicit or implicit assumptions about the time series. Usually several models has to be tested to discover proper representation of the given problem. Nevertheless the process of the fitting the data and choosing parameters manually, semi-automatically or automatically has to be performed.
5. **Using and evaluating a forecasting model:** When the model is prepared, it is ready to make prediction in production environment. Only way to assess it's real value, is to compare forecasted values with real observations.

8.2 Problem Definition

The goal of this task is to determine future sales per day, providing customer estimation of performance of the business, enabling him to estimate necessary amount of human resources to allocate, also known as **staffing**. This task emerged as practical need from customers using the POS, as representatives of the company were approaching clients using the system.

Sales per day are aggregated orders performed over period of a day. Order is action unit within the system, representing finished or unfinished transaction, containing items, special pricing, taxes, information about monetary transactions made etc. For purpose of this task is suitable primary **net revenue** of the order, representing pure income performed by monetary **transactions** (cash, card etc.) with values of cancellation actions subtracted from it[1].

8.3 Gathering Information

Business data are available from production database. There is need to choose right datasets containing correct data to perform preliminary analysis and consequently build a model. There are currently 84 accounts in the database, many of them being testing accounts, unused accounts or newly created accounts not containing necessary amount of data.

For this choice, only datasets with enough amount of historical data were taken, this also excluding testing or unused accounts. Data for past 60 days were considered to asses whether account is viable to further use. Only accounts with more than 40 active days[2] and sufficient average order count per day, as well as total number of orders considered:

```
select *, orders_count / active_days as order_per_day from (
  select
    account.name,
    account.id,
    count(*) as orders_count,
    count(distinct "order".closed_timestamp :: date) as active_days
  from "order"
  left join account on "order".account_id = account.id
  where (now() - "order".closed_timestamp) < interval '59 days'
  group by account.id
  order by active_days desc
) base;
```

[1] Transaction can be marked as voided, when mistakenly entered into the system or refunded, when clients are not satisfied with the service.

[2] Active day means there was at least 1 order at that day.

Table 8.1: Chosen datasets according amount of historical data

Name	Orders / Day	Total Count	Active Days
Café	50	3056	60
Thai Restaurant 2	16	971	60
Thai Restaurant	24	1493	60
Gelato	33	2037	60
Nail Salon	37	2222	59
Spa	7	466	59
Sushi Restaurant	49	2928	59
Deli	150	8906	59
Nails Salon 2	39	2253	57
Café 2	29	1617	54
Dry Cleaners	17	852	50
Dry Cleaners 2	14	694	49
Thai Restaurant 3	14	714	49
Dry Cleaners 3	6	306	44
Mexican Restaurant	38	1651	43

From the table it is visible, that some of the datasets has very few orders per day, which can be problematic for the modeling, since there is not enough statistical certainty, but explicit modeling process has to be performed to support this view.

8.3.1 Assumed predictors

From domain-knowledge expertise view, domain experts were interviewed to obtain information about influences on business performance. These were break down into these points:

1. **Seasonality**: For instance, during summer, there are lot of tourist in New York, it is supposed to be similar in most of major cities in United Stated.
2. **Weather**: Expected weather is more important than actual. Weather affects suburban areas much more than urban areas Precipitation is perhaps more important than temperature. Especially for snow, suburban businesses suffer considerably. Contrary to the relative business, delivery orders (for businesses that offer it) shoots up tremendously when there is precipitation First warm day coming out of winter usually helps sales in a big way.
3. **Nearby Events**: E.g. one of the businesses is across the street from a major stadium and whenever they have events, venue gets busier.
4. **Holiday**: When holidays border Friday or Monday and making total sum larger usually increase in sales for certain businesses like Deli.
5. **Festivals**: Events not based on holidays but based upon regions.
6. **Lack of Parking**: When parking is not sufficient in the space, adversely affects business.
7. **Non-annual events**: Elections, major movie releases, Olympics, sporting events (during American football season, some restaurants slow down considerably on Sundays that do not show football)
8. **Nearby Competition**: Opening up of a new nearby business in the same industry.

Major holidays and special days were collected in collaboration with domain experts to address them in modeling phase:

- May, 2nd Sunday – **Mother's day:** This is the busiest day. Many families like to go out to eat instead of cooking from home on this holiday. If Mother's day is on a weekend, the whole weekend is very busy.
- February, 14th – **Valentine's day**. Second busiest day of the year. Many restaurants are fully booked for this holiday. If they don't have reservations, many restaurants fill up quickly. The week before and after may be busier because people like to avoid the massive Valentine's day crowds.
- June, 3rd Sunday – **Father's day**. Usually the 3rd busiest day of the year. For the same reasons, but lesser extent, as Mother's day, people do not want to cook at home for Father's day.
- December, 31th – **New Year's Eve**: People go out to eat before they drink or celebrate the new year. Also, they spend more because they might start with dinner and remain in the bar or restaurant until midnight.
- March, 22th to 25th, April – **Easter**: Many families eat out for brunch, lunch, or dinner on this holiday.
- January/February – **Super Bowl**: Any bar or restaurant with a TV gets significant amount of business this day. If they do not have a TV, many people order food for delivery or to go on this day.
  March, 17th – **St. Patrick's Day**: Bars and restaurants usually get a larger drinking crowd on this night.
  December 24th – **Christmas Eve/Day**: Christmas may get busy for businesses right before or after this day, but not usually on that day. People like to go out the days leading up to it or the days after these holidays.
  November, 22th – **Thanksgiving**: Thanksgiving may get busy for businesses right before or after this day, but not usually on that day. People like to go out the days leading up to it or the days after these holidays.
  July, 4th – **Independence Day**: For most of the country non-working day. The standard plans are to grill food at home that day but many choose to go out to eat sometime during that whole week.
  October, 31th – **Halloween**

Special days has to be scripted into the model, some of them has to be manually entered (Super Bowl), since they cannot be derived purely from calendar. Especially problematic is indirect effect of some special days (e.g. Valentine's day), which can impact any day in proximity of the nominal event.

8.3.2 Weather data

Data about weather has to be collected. There are several services providing API for retrieving weather data. Services were chosen from according to availability of historical data, as well as ability to forecast and also, with regards to pricing.

Four services providing weather data API were examined. **OpenWeatherMap** has relatively costly solution for historical data compared to other free or cheaper solutions providing same service. **Weather Underground** does not offer standard historical API service and there is need to contact sales department. **Dark Sky** provides very reasonable prices with relatively great amount of free requests per day. It also offers good API and has better forecasts than **World Weather Online**, which only offers limited trial.

Dark Sky provides HTTP API requests with URL in form of:
`https://api.darksky.net/forecast/[key]/[latitude],[longitude],[time]`.

Response in JSON format is received, containing aggregated daily data and also per hour breakdown describing important weather metrics for specific hour [3]:


```
{
    "latitude": 42.3601,
    "longitude": −71.0589,
    "timezone": "America/New_York",
    "hourly": {
      "summary": "Snow \(6-9 in.\) and windy starting in the afternoon.",
      "icon": "snow",
      "data": [
        {
          "time": 255589200,
          "summary": "Mostly Cloudy",
          "icon": "partly-cloudy-night",
          "precipIntensity": 0,
          "precipProbability": 0,
          "temperature": 22.8,
          "apparentTemperature": 16.46,
          "humidity": 0.73,
          "windSpeed": 4.83,
          "cloudCover": 0.78,
          "visibility": 9.62
        },
        ...
      ]
    },
    "daily": {
      "data": [
        {
          "time": 255589200,
          "precipIntensity": 0.0354,
          "precipProbability": 1,
          "precipAccumulation": 7.337,
          "precipType": "snow",
          "pressure": 1016.41,
          "windSpeed": 22.93,
          "cloudCover": 0.95,
          "visibility": 4.83,
          "apparentTemperatureMin": 11.13,
          "apparentTemperatureMax": 20.47,
         ...
        }
      ]
    },
    "offset": −5
  }
```


Attributes, that were chosen for the forecasting are summarized in this table:

[3] API response was simplified and only substantial attributes are shown.

Table 8.2: Chosen attributes from DarkSky API

Attribute name	Description
apparentTemperatureHigh	Highest apparent temparture in $°F$
cloudCover	Cloud cover [0,1]
humidity	Humidity [0,1]
windSpeed	Wind speed in knots
visibility	Visibility in miles
icon	String representing overall weather on that day
precipType	Type of precipration (rain, snow, none)
precipIntensity	Intensity of precipration in meters
precipProbability	Probability of precipration [0,1]

Since aggregated sales data are used, solution how to relate dynamically changing weather to this aggregated data has to be found. Proposed solution is to extract per hour metrics by opening hours of particular business. Then, represent these metrics as real value vector describing percentage representation in examined period.

Since there are no data describing opening hours in the system, it has to be inferred from timestamps of performed orders (results can be seen in Table 8.3):

```
select account_id, avg(first_hours) as first ,stddev(first_hours) as first_dev, avg(
    last_hours) as last,stddev(last_hours) as last_dev from
  (
    select
      dt,
      account_id,
      EXTRACT(MINUTE from first) / 60 + EXTRACT(HOUR from first) as first_hours,
      EXTRACT(MINUTE from last) / 60 + EXTRACT(HOUR from last) as last_hours
    from account a
      left join (
        select
          o.created_timestamp :: date as dt,
          account_id,
          (min(o.created_timestamp) FILTER (WHERE EXTRACT(HOUR from o.created_timestamp) >
              3)) as first,
          max(o.created_timestamp) as last
        from "order" o
        where NOW() - o.created_timestamp < INTERVAL '7 days'
        group by
          o.created_timestamp :: date,
          o.account_id
      ) base on base.account_id = a.id
    where a.id in ('daxq', 'wtu2', 'n8wm', 'tu84', '8gbg', ...)
  ) to_agg
left join account on account.id = account_id
group by account_id, account.name
```

Table 8.3: Expected opening hours with corresponding standard deviations

Account Name	μ_{open}	σ_{open}	μ_{closed}	σ_{closed}
Dry Cleaners 2	13:50	02:22	21:19	01:15
Dry Cleaners	12:35	02:34	21:50	01:23
Thai Restaurant	16:35	00:49	22:58	02:22
Sushi Restaurant	15:50	00:58	22:59	02:06
Thai Restaurant 3	16:22	01:26	22:47	01:60
Deli	12:05	02:44	21:58	01:26
Café 2	16:27	00:36	22:30	02:01
Café	12:35	04:18	22:50	01:50
Nail Salon	14:47	01:36	22:55	01:53
Mexican Restaurant	15:53	00:59	23:04	01:56
Nails Salon 2	14:45	01:30	22:37	01:43
Dry Cleaners 3	15:52	03:50	22:16	01:53
Thai Restaurant 2	16:56	01:39	22:56	02:01
Spa	16:33	01:43	21:38	01:39
Gelato	16:47	00:34	23:01	01:56

8.3.3 Sales data

Data were obtained in two ways. First, used previously, was to manually dump the data from remote PostgreSQL database to local file in binary format and then import it to local database. Pandas allows to directly load data into `DataFrame` structure used for manipulation with the data through database connection using **SQLAlchemy**. Therefore, several aggregation SQL queries were used to obtain the data suitable for further processing.

Due to security reasons and also formal requirements of **Payment Card Industry Data Security Standard** (**PCI DSS**), which also had to be applied to backend application generating the data for use in production application, another solution had to be made. Backend of the POS system communicates with frontend application through the custom JSON API, which was used to obtain the relevant data. The API was developed ad-hoc for particular reports available in the frontend, but contains also some general aspects, which could be exploited to retrieve necessary data. Both approaches are available in file `prediction/datasets.py` and API approach particularly in classes in `prediction/api.py`.

API requires user credentials, also used for logging into the application, for accessing the authorized data. Only data, related to the account used to log in with, are available. Thus, generic account, which was permitted to access given data, is used. Credentials in attached software are naturally omitted, but offline version of the anonymized data is available.

8.4 Time Series Modeling

Formally, time series is defined as special case of **random process**, measured values are called **sample path**:

Definition 8.1. Random (stochastic) process is nonempty family ($T \neq \emptyset$) of (real) random variables defined on the same probability space $(\omega, \mathcal{A}, P)$. We write $X = \{X_t \mid t \in T\}$ or simply $\{X_t\}$.

Special cases:
$T \subseteq \mathbb{R}$ ... **continuous time process** or **random function**
$T \subseteq \mathbb{Z}$ ... **discrite-time process** or **time series**

Definition 8.2. For fixed $\omega \in \Omega$ we get function $x : T \to \mathbb{R}$ as an outcome of random experiment $x(t) := X_t(\omega)$. This function is called **sample-path** (**trajectory**, **realization**, **observation**) of X.

For many modeling techniques, condition of stationarity has to hold:
Definition 8.3. Time series $X = \{X_t \mid t \in \mathbb{Z}\}$ is called **(weakly) stationary** if the following three conditions are fulfilled:

1. X has finite second moments,
2. $\gamma_X(r, s) = \gamma_X(r + h, s + h)$ for each $r, s, h \in \mathbb{Z}$,
3. $\mu_X(\,\cdot\,) \equiv \mu_X$is a constant function,

where γ is auto-covariance function and μ is mean. Related term is **covariance stationarity**, for which only first two conditions hold.

Consequently, stationary time series has constant mean and variance. Also, covariance function is shift-invariant and is dependent only on distance h between two points in time series. For inference about auto-regression used in auto-correlation plot and partial auto-

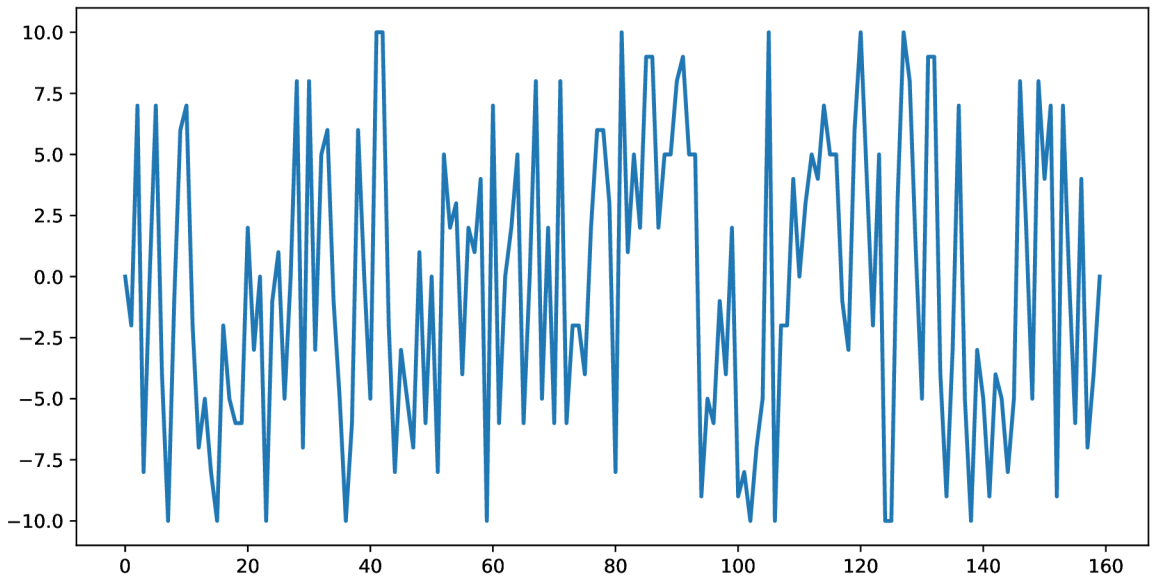


Figure 8.1: White noise

correlation plot in preliminary analysis are important **auto-correlation function (ACF)** and **partial auto-correlation function (PACF)**. For these two metrics, **Pearson's correlation coefficient** has to be defined:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\mathrm{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{8.1}$$

where:

- *cov* is the covariance

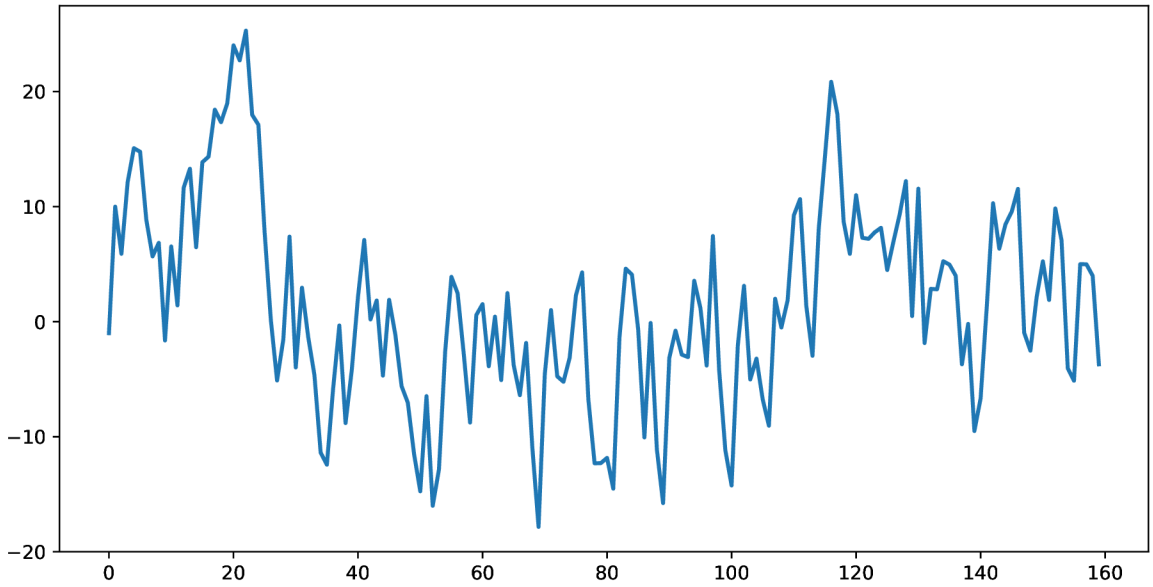


Figure 8.2: Stationary time series

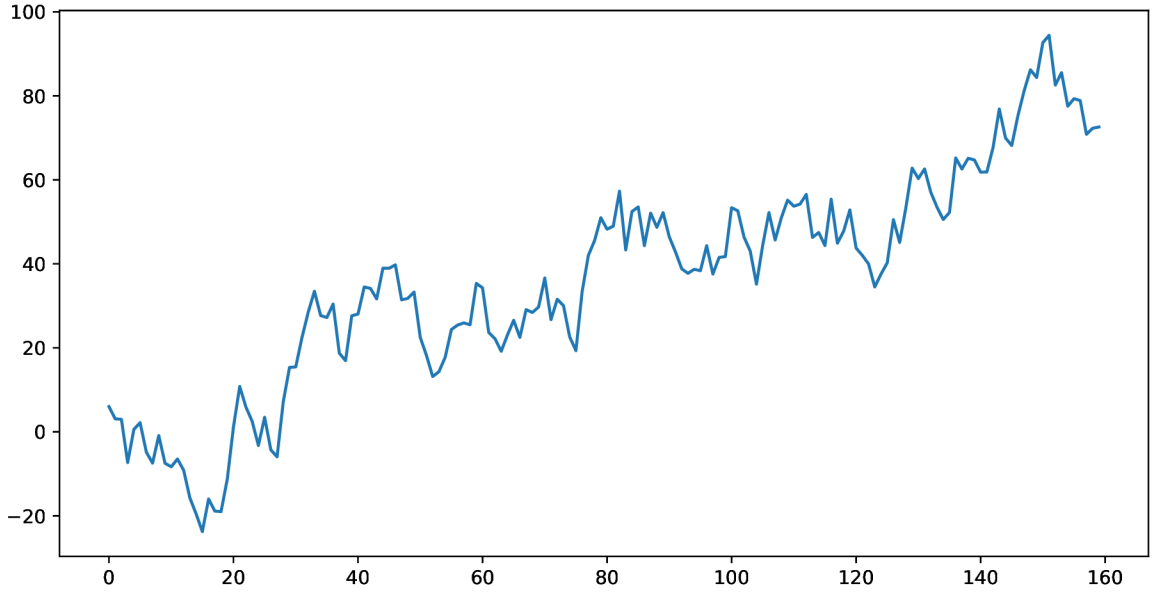


Figure 8.3: Covariance stationary time series

- μ_x mean of X
- μ_y mean of Y
- σ_X standard deviation of X
- $\sigma_Y$standard deviation of Y

Pearson's correlation coefficient is defined for two random variables, measuring linear correlation between them, ranging from -1 for negative correlation to $+1$ for positive correlation, 0 meaning no correlation. For measured samples, different equation is to be used:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{8.2}$$

where:

- n is the number of samples
- x_i, y_i are the measured values of the two random variables
- $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ (the sample mean); analogously for $\bar{y}$

With Pearson's correlation coefficient applied to time series X, it is possible to define auto-correlation function for given two points in time s and t:

$$R(s,t) = \rho(X_s, X_t) = \frac{\gamma_X(s,t)}{\sqrt{\gamma_X(s,s)}\sqrt{\gamma_X(t,t)}} \tag{8.3}$$

For stationary time series, from Definition 8.3, covariance between two points in time depends only on their distance $h = |s - t|$, from that we can define auto-correlation function as function of one parameter τ (lag) [18, 2.1.2.]:

$$R(\tau) = \frac{\gamma_X(\tau)}{\sqrt{\gamma_X(0)}^2} = \frac{\gamma_X(\tau)}{\gamma_X(0)} \tag{8.4}$$

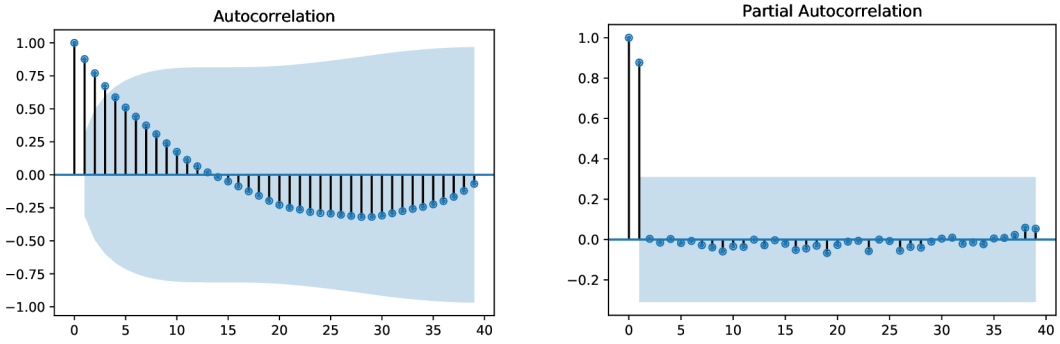
This function is used for **auto-correlation plot** (see Figure 8.6), it is suitable for order estimation of MA (see section 8.8) process.

As opposed to auto-correlation function, **partial auto-correlation function** eliminates correlation propagated further indirectly by controlling correlation with lag shorter than examining one, thus is suitable for degree estimation of AR process [19, Section 3.4]:

$$\alpha(1) = \rho(z_{t+1}, z_t), \tag{8.5}$$

$$\alpha(k) = \rho(z_{t+k} - P_{t,k}(z_{t+k}),\ z_t - P_{t,k}(z_t)),\ \text{for } k \geq 2, \tag{8.6}$$

where $P_{t,k}(x)$ denotes the projection of x onto the space spanned by $x_{t+1}, \ldots, x_{t+k-1}$.



(a) Auto-correlation function

(b) Partial auto-correlation functopn

Figure 8.4: Comparison of ACF and PACF of AR(1) process

Figure 8.6 shows auto-correlation and partial autocorrelation function for AR(1) process displayed in Figure 8.8. First column is always of magnitude 1, since it captures autocorrelation with itself. ACF exhibits slow decay since ϵ is propagated to further steps and causing correlation with previous steps, on the other hand PACF is showing significant only lag at position 1, explaining order of the AR process.

8.4.1 Evaluating accuracy

For performance evaluation of suggested models, measurement metrics has to be established. One of the criteria is good comprehensibility for the user and the representatives of the company. Other criterion is scale-independence for comparing model among datasets, but also for for evaluating performance on particular dataset, since it can be unclear, what absolute error is acceptable for the dataset. Also, symmetry of the metric is considered beneficial.

There are many methods for error measurement, although particular advantages and disadvantages has to be examined. It is possible to divide error measurements into these categories:

- Scale-dependent errors
- Percentage errors
- Scaled errors

Among often used scale-dependent errors belong **Mean Absolute Error** (MAE) and **Root Mean Squared Absolute Error** (RMSE) [21]:

$$\text{MAE} = \frac{\sum_{t=1}^{n} |\hat{x}_t - x_t|}{n} = \frac{\sum_{t=1}^{n} |e_t|}{n} \tag{8.7}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n} (\hat{x}_t - x_t)^2}{n}}. \tag{8.8}$$

RMSE represents standard sample deviation between observed values and predicted ones. It has key attribute to evaluate more deviated errors unproportionally worse, in other words more deviated but less frequent errors cause larger RMSE error. This is subject of critique [21], arguing that RMSE is not dependent only on magnitude of error itself, but is affected by other factors – number of observations ($n^{\frac{1}{2}}$) and distribution of error magnitudes. On the other hand, penalization for more deviated errors can be desirable, as user can accept more frequent, less serious error more indulgently than less frequent, more serious error. That said, MAE has additional advantage in being more comprehensible for user or executives. As a conclusion, MAE can be used for presentation of the results, while RMSE can be used as a target function intended for minimization.

Assessing percentage errors, most commonly used percentage measurement is **Mean Absolute Percentage Error** (MAPE):

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{x_t - \hat{x}_t}{x_t} \right| \tag{8.9}$$

MAPE is ratio of absolute error to observed value, in other words, it states how many percents prediction deviates from observed value. Most significant advantage is interpretability. Although MAPE is most commonly used percentage error measurement, it suffers from many disadvantages [22]. Clearly, it cannot be used in cases, when observed value is zero.

According to Tofallis, when MAPE is used to select model or estimate model parameters, it constantly underestimates, which origins from fact, that for predicted values higher that observed values, MAPE is significantly larger, it holds that $lim_{y \to 0} MAPE(y, \hat{y}) = \infty$.

Also, it is not symmetrical, same absolute error can implicate different MAPE, depending on which component is observed and which is predicted. This problem is trying to solve

SMAPE (Symetric mean absolute percentage error):

$$\text{SMAPE} = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|\hat{x}_t - x_t|}{(|x_t| + |\hat{x}_t|)/2} \tag{8.10}$$

It although does not remove other above mentioned problems. Furthermore, because the reference value to compare to will be always closer to comparing one due to averaging the predicted and observed value, resultant error will be lower and thus possibly to describe as biased.

Tofallis suggests taking logarithm of relative accuracy[4] for error measurement. It resolves above problems of MAPE, although compared to it, it is not such clearly interpretable. It is defined by:

$$ln\ Q = \frac{1}{n} \sum_{t=1}^{n} ln\ \frac{\hat{x}_t}{x_t} \tag{8.11}$$

Another interesting error measurement belonging to the scaled errors – **Mean Absolute Scaled Error** (MASE) was introduced by Hyndman and Koehler [23]:

$$\text{MASE} = \frac{1}{n} \sum_{t=1}^{n} \left(\frac{|e_t|}{\frac{1}{n-1} \sum_{t=2}^{n} |x_t - x_{t-1}|} \right) = \frac{\sum_{t=1}^{n} |e_t|}{\frac{n}{n-1} \sum_{t=2}^{n} |x_t - x_{t-1}|}, \tag{8.12}$$

as well as for seasonal time series:

$$\text{MASE} = \frac{1}{n} \sum_{t=1}^{n} \left(\frac{|e_t|}{\frac{1}{n-m} \sum_{t=m+1}^{n} |x_t - x_{t-m}|} \right) = \frac{\sum_{t=1}^{n} |e_t|}{\frac{n}{n-m} \sum_{t=m+1}^{n} |x_t - x_{t-m}|} \tag{8.13}$$

Above error measurement uses naive forecast method as normalization factor to produce relative error and this way solving above mentioned problems with MAPE. It is interpretable as ratio, how well is model doing in comparison to naive method, which expects future steps to equal past values.

In conclusion, from above mentioned reasons, for evaluating models, Mean absolute error (MAE) and Mean Absolute scaled error (MASE) were chosen. Furthermore Mean absolute percentage error (MAPE), was chosen as a complement for interpretable explanation of model performance in comprehensive terms.

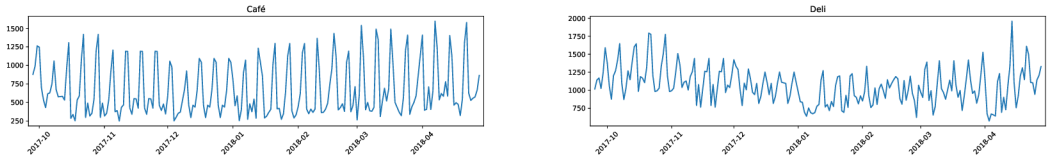
8.5 Preliminary Analysis

One of the first steps of preliminary analysis of time series is to visualize series in form of chart. Purpose of the preliminary chart is to visually analyze character of the data, observe obvious patterns and identify components of the time series. Complete graph listing can be found in Attachment C.

Most of the datasets exhibit seasonal weekly character 8.5. Some of them exhibit trend (Gelato, Thai Restaurant 2), even in latter, time period is too short to explicitly state that.

From course of the many series, it is clear that there is great amount of stochasticity from perspective of time series itself, thus modeling will be difficult without providing external descriptive input.

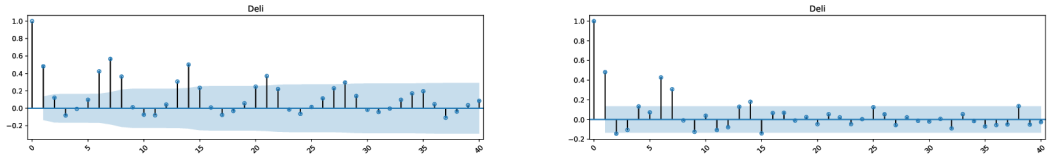
[4] complement of relative error



(a) Café (b) Deli

Figure 8.5: Examples of preliminary plot

From auto-correlation and partial auto-correlation plot is possible to determine linear correlation between value at particular time and it's precedent values. This can be used to estimate parameters of some of the following models. For ARIMA model (see below), one of the ways to estimate parameters is Box-Jenkins method [18, Chapter 6], which utilizes ACF and PACF.



(a) Auto-correlation function (b) Partial auto-correlation function

Figure 8.6: Examples of ACF and PACF for real data (Deli)

From given ACF plots can be observed, that for many datasets (Café, Café 2, Deli, Gelato, Nail Salon, Nail Salon 2, ...), there is weakly seasonality (lag 7), as well as peaks on lag 6 and lag 1. Both strong lag 6 and lag 7 can be interpretable as result of correlation with both weekly lag and previous day lag.

muSome datasets exhibits weaker autocorrelation (Dry Cleaners 2, Mexican Restaurant, Spa). It is interesting, that some datasets (e.g. Dry Cleaner's) exhibits strong autocorrelation with period 6 days, but almost no correlation at period 7 days. With many datasets there is also progressive decline over given periods, representing trend.

Examining PACF plots, for many datasets with weakly period, lag 7 diminishes since it is also correlated with shorter lags - lag 1 and lag 6. Also, with PACF, more distance period diminishes, since they are correlated with closer periods.

With above said, it can be concluded, that many datasets will be probably possible to model and predict based on auto-correlation with certain error, but for better results there should be found other explanatory variables.

8.5.1 Preprocessing

Obtained data are relatively consistent, since originates from singular system, although preprocessing is still crucial to allow models to perform well. The aim was to enable this procedure to be performed automatically, since the system will be provided with new data and the system should adapt to it. By investigation of the data, following preprocessing task were suggested and consequently implemented:

1. Outliers regularization

2. Testing period removal[5]
3. Special days regularization
4. Filling missing values

All of the above steps are intended for improving model capability to be trained, but when evaluating the data, original data were used, so results is not biased by decreasing error with introducing above mentioned regularizations into series.

Preliminary results revealed, that data contain unfitting extreme values, caused by data migrating or testing. These values are removed from the data. Outliers are tested by comparing deviation of tested data point with standard deviation of it's τ-neighborhood, described by Equation 8.14.

$$\epsilon_\tau(x_t) = (x_{t-\tau}, x_{t-\tau+1}, \ldots, x_{t-1}, x_{t+1}, \ldots, x_{t+\tau-1}, x_{t+\tau}) \tag{8.14}$$

Given value is said to be outlier, if following inequation holds:

$$|x_t - \mu(\epsilon_\tau(x_t))| \geq \ \delta(\epsilon_\tau(x_t)) \cdot h, \tag{8.15}$$

where $\mu(\cdot)$ is sample mean, $\delta(\cdot)$ is sample standard deviation and h is threshold constant. Constants τ and h are set by default to 30 and 3 respectively. Outliers are substituted with naive seasonal forecast:

$$x_t \leftarrow x_{t-s}, \tag{8.16}$$

where s is dominant period of the dataset. Although, some of further employed models are capable of dealing with missing values[6], substituting with naive forecast led to better forecasts during experimentation.

Further, portion of the data, that is not describing proper business process, is to be removed. This period is characterized by occasional sales data, separated by period of non-activity. As these periods would disrupt the model training process, they ought to be removed. Window filter for testing subsequent days for activity is suggested:

$$x_t \leftarrow \emptyset \text{ if } \Lambda > \sum_{\tau=1}^{k} c(x_{t+\tau}), \tag{8.17}$$

$$c(h) = \begin{cases} 1 & \text{if } x \geq \lambda \\ 0 & \text{otherwise,} \end{cases} \tag{8.18}$$

where k is number of days to look ahead, Λ describes minimum number of active days in a window and λ is minimal amount of sales to consider day as active.

Last step is to deal with missing values, although, not many missing values are present, there are days when business was for example closed, this interferes with learning process. For this occurrences, Equation 8.16 is used. Closed days occurring within week regularly has to be taken into account and excluded from this process. Regular closed day are identified as occurring $\geq 50\%$ instances within week. Also, identified regular closed days are used on different places further in the process of modeling for automatic weekly period inference.

[5] A time period, when the system was not fully used, but yet there exist sparse data used for testing.
[6] e.g. by state space model, SARIMA implantation used further

8.6 Implementation Framework

For implementation part, Python programming language was chosen in Chapter 5. Many of the implemented models use common libraries, particularly:

- **Pandas**: used for loading data, preserving and performing operations over these data, including operations commonly used in DBMS like aggregation, selection etc.
- **Statsmodels**: statistical functions, including different statistical tests, as well as implemented different types of model used for regression and time series modeling
- **NumPy**: efficiently store n-dimensional arrays and perform operations over them, NumPy structures are primarily used as input to other libraries
- **Matplotlib** and **Bokeh**: libraries for performing visualizations, while the former is primarily used for export visualizations for purpose of this paper, since it is suited more for static formats, the latter is used for examination of the data, because it allows export into HTML, containing JavaScript for interactivity and thus making it possible for interactive examination of the data.
- **SciPy**: provides another statistical and time series functions, like Savitzky–Golay filter or Pearson's correlation coefficient.

For development and evaluation of the models, simple framework was developed to make this process more efficient. Basic idea is to develop model generally and then execute it and evaluate it on arbitrarily number of datasets with output of performance metrics for particular datasets, as well as aggregated metrics for all datasets.

Also there is possibility to modularly add more (error) metrics without interfering with any of other code. Furthermore, output error metrics are stored in cache and then used in computing difference with new values for instant feedback during edit – run – evaluate cycle. For improving efficiency, forecasting is performed concurrently for different time periods, which are then used separately for evaluation. Since Python implementation CPython is using GIL (Global Interpreter Lock), it is not possible to use threads, processes are used instead, but because processes are created initially and then reused, this approach is sufficient.

Figure 8.7: Diagram of modelling framework

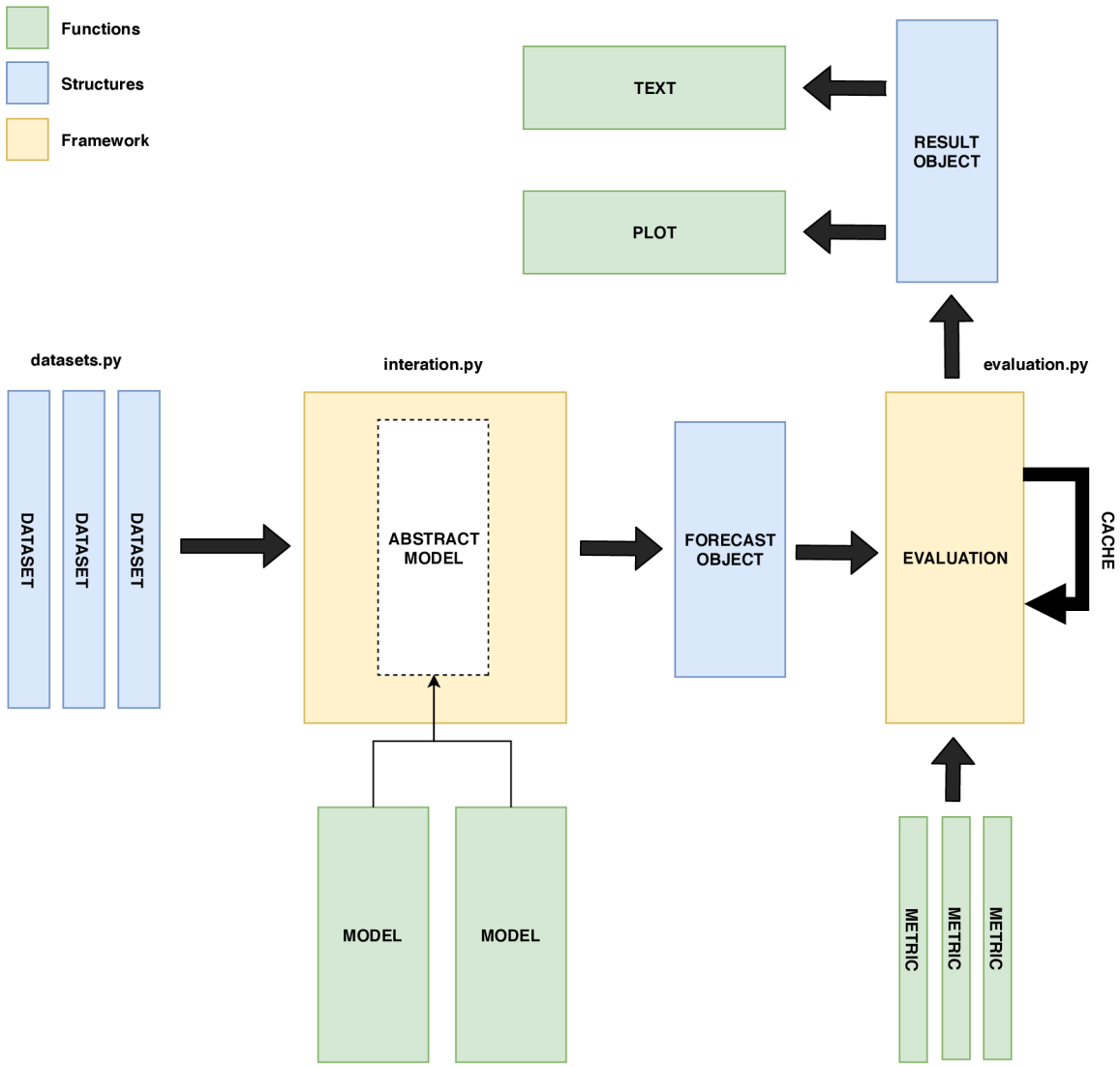


Figure 8.7 schematically describes proposed framework. Datasets are loaded by `datasets` module, producing preprocessed conventionally labeled Pandas `DataFrame`-s. Then, chosen model is passed into `iteration` procedure, which executes the model using common utility functions, designed e.g. for structuring data into train and test sets according to particular model. Model outputs `forecast object`, containing partitioned[7] predicted and test data for all datasets. Result object are then passed into `evaluation` module, which applies defined error measurements and computes aggregated values. Resulting `result object` contains this data in abstract way, so they are suited for further processing. Functions, chosen by input procedure[8], designed for displaying results as a graph or in a table are called upon these data.

Framework exploits possibilities of Python functional programming paradigm in defining most of the modular parts as pure functions to archive greater flexibility. Models are

[7]For greater conclusiveness, models are evaluated on greatest range of inputs possible by partitioning
[8]Program is callable with different intentions, specified by `-run-function` parameter.

simply functions which receives input and outputs evaluated result. This is also useful for parallelization of the process.

8.7 Model 1 – Decomposition

One of the classical approaches to model time series is decomposition model. Decomposition model deconstructs time series into four components:

- Trend – Tr_t
- Seasonality component – Sz_z
- Cycle component – C_t
- Residual – E_t, $E_t \approx WN$ or $E_t \approx IID$, where WN is white noise and IID is independent and identically distributed

Trend component increases (or decreases) in long term (or dynamically changes slope) linearly or nonlinearly. Seasonal and Cycle components have common that in both period of the component can be identified. Cycle component differs from seasonal component in having non-fixed period and thus harder to model. When trend, seasonal and cycle component is subtracted, the reminder is called residual, which is supposed be white noise.

There are three basic types of decomposition model according to relation between particular components:

- Additive model: $X_t = Tr_t + Sz_t + C_t + E_t$
- Multiplicative model: $X_t = Tr_t \cdot Sz_t \cdot C_t \cdot E_t$
- Other: $X_t = m(Tr_t, Sz_t, C_t, E_t)$

There are several generic ways to decompose to these components, the classical decomposition [1, Section 6.3] applies rolling average filter two times to obtain obtain trend with cycle component and then simply averaging corresponding values of the periods from detrended series to extract seasonal component. The reminder is residual.

Other approach that was studied is X-13ARIMA-SEATS, developed by United States Census Bureau, government organization responsible for different national and international statistics. It it complete modeling tool, not only decomposing method, but integrates also ARIMA (see further) and SEATS (Signal Extraction in ARIMA Time Series), which model time series components by ARIMA. It contains many ad-hoc procedures inferred from practical knowledge, but it is adapted to monthly socioeconomic data, thus not suitable for current task. [1, Section 6.4] [20]

For purpose of this thesis, simple implementation of **STL** (Seasonal and Trend decomposition using LOESS) procedure was examined [24]. LOESS (Local Regression) is fitting polynomial using weighted least squares, giving more weight to the points closer to the evaluated one. For trend forecasting, drift method was chosen, since it can capture change of a trend and forecast it into future. Because this particular implementation does not contain ability to change parameters over time, based on hyper-parameter optimization, training period of 30 days was chosen, as it was evaluated as containing best error rate.

Table 8.4: Results for decomposition model

Name	MAPE	MASE	MAE	Mean
Nail Salon	28.07%	1.09	$342.7	$1375.3
Café	41.11%	1.71	$248.3	$680.0
Gelato	59.76%	1.44	$196.6	$422.6
Deli	16.24%	1.09	$160.7	$1038.9
Sushi Restaurant	32.0%	1.17	$536.3	$1950.0
Dry Cleaners	69.91%	1.53	$305.5	$566.0
Thai Restaurant	37.09%	1.22	$232.0	$787.3
Nails Salon 2	41.25%	1.39	$558.8	$1661.1
Mexican Restaurant	31.89%	1.07	$299.7	$970.9
Café 2	54.59%	1.21	$222.7	$582.1
Thai Restaurant 3	85.86%	1.32	$394.0	$783.7
Thai Restaurant 2	47.29%	1.15	$114.2	$287.8
Dry Cleaners 2	72.32%	1.25	$420.9	$649.0
Spa	87.86%	1.06	$327.3	$455.5
Dry Cleaners 3	47.55%	0.73	$69.3	$176.7
Aggregated	52.06%	1.3	$318.1	$917.5

As table shows, this approach does not archive very good results. There is possibility to use better implementation of STL, namely standard implementation in R [25]. Also, decomposition techniques are focused to description of the time series and lacks techniques for forecasting, e.g. STL can identify trend, but is not clear, how to forecast future trend from current observation.

8.8 Model 2 – SARIMA

SARIMA model (**Seasonal Autoregressive Integrated Moving Average**) is a generalization of autoregressive integrated moving average (ARIMA) model. ARIMA(p, d, q) consists of two components – moving average and auto-regression:

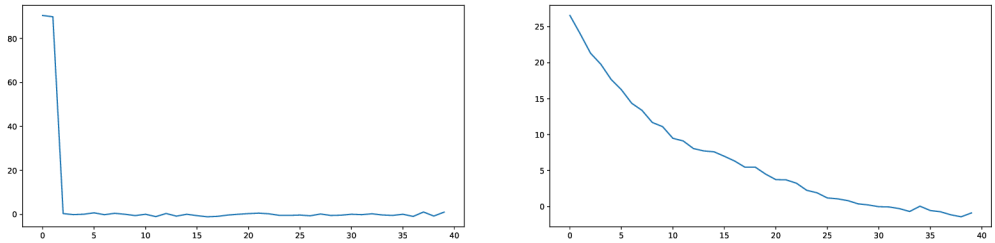
1. **Auto-regression (AR)**: $X_t^{AR} = c + \sum_{i=1}^{p} \phi_i X_{t-1} + \varepsilon_t$, where c is a constant, ϕ_i are parameters and ε_t is white noise.
2. **Moving average (MA)**: $X_t^{MA} = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t$, where μ is constant, θ_i are parameters and ε_t is white noise.

AR can bee seen as infinite signal response, since random shocks ϵ_t propagates infinitely into future as a component of X_t. On the other side, MA behaves like finite signal response, since ϵ_t affects output only for the q steps into future. As seen on Figure 8.8, showing response to unit impulse, MA exhibits fast decline and AR shows gradual decrease.

Equation 8.19 describes complete ARIMA(p, d, q) model with both AR and MA component:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \phi_i X_{t-1} + \sum_{i=1}^{q} \theta_i \tag{8.19}$$

Parameters p and q, denotes order of the model, describing number of previous values to consider, d parameter describes number of differencing steps to perform.



(a) MA(1), $\delta_\epsilon = 0.5$ (b) AR(1), $\delta_\epsilon = 0.7$

Figure 8.8: Examples of AR and MA processes response to unit impulse

Differencing is statistical transformation used to make time series stationarity by computing difference between current and previous value:

$$y_t\prime = y_t - y_{t-1} \tag{8.20}$$

SARIMA$(p,d,q)(P,D,Q)_s$ extends previous by duplicating ARIMA model by adding complementary ARIMA for seasonal component with parameter s denoting seasonal period and parameters P, D and Q denoting orders for AR, integration and MA.

8.8.1 Hyper-parameter optimization

One of the options to improve performance of the model is to use some automatic procedure to choose parameters of the model. **Hyper-parameter optimization** is general technique to search for optimal value of the parameters determining how model is fitted. In case of SARIMA model, they are p, d, q, P, D, Q. State space for reasonable range of parameters is possible to define this way:

$$S_{(p,d,q,P,D,Q)} = ([0,7]\times[0,3]\times[0,5]\times[0,4]\times[0,3]\times[0,3]) \setminus (\{0\}\times[0,5]\times\{0\}\times[0,3]\times\{0\}\times[0,3]) \tag{8.21}$$

In the second part of the Equation 8.21, unreasonable states, which have zero for all of p, q, P, Q parameters, are subtracted.

One of the approaches to choosing parameters is to perform exhaustive search (or **grid search**), which iterates over all valid parameter combinations in sequential manner. With above mentioned state space of parameters, this search would not terminate in reasonable time, since number of states is $|S_{(p,d,q,P,D,Q)}| = 15264$.

Another option is to perform **stochastic grid search**, which evaluates states randomly. After a given number of iterations, the algorithm is stopped and the best parameter combination is retrieved. This approach is further investigated. Grid search is embarrassingly parallel, thus an **EC2 instance** in **Amazon Cload Service (AWS)** with enough processor units was designated for this task. Type **c5.4xlarge** with 16 processor units and 32 GB memory was chosen.

Café, Thai Restaurant and Gelato datasets were chosen as representatives of different time series characteristics. Results from hyper-parameter optimization will be consequently examined to choose the appropriate parameters.

Table 8.5: Thai Restaurant: Grid search results showing best and worst parameters

(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)	(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)
[[5, 0, 0], [3, 0, 2]]	19.7%	0.797	166.9	37.8	[[6, 2, 3], [1, 2, 2]]	140.9%	6.363	1332.8	41.4
[[4, 0, 1], [3, 0, 0]]	19.9%	0.762	159.6	7.5	[[1, 2, 0], [1, 2, 0]]	148.9%	6.093	1276.4	2.3
[[4, 0, 0], [3, 0, 0]]	19.9%	0.763	159.9	6.4	[[6, 2, 3], [1, 2, 1]]	154.3%	10.876	2278.3	16.7
[[5, 0, 0], [3, 0, 1]]	20.0%	0.812	170.1	20.8	[[1, 2, 0], [0, 2, 0]]	181.3%	8.112	1699.3	1.0

Results for Thai Restaurant (Table 8.5) show, that the best settings for the dataset is no differencing, p and q describing 5 day lag and similarly seasonal component capturing approximately 3 week lagged value.

Table 8.6: Café: Grid search results showing best and worst parameters

(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)	(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)
[[2, 0, 3], [2, 1, 2]]	18.2%	0.724	132.2	17.9	[[0, 2, 0], [3, 0, 1]]	119.6%	3.66	667.9	12.1
[[0, 1, 4], [3, 0, 0]]	18.3%	0.705	128.7	10.1	[[0, 2, 0], [1, 2, 0]]	126.2%	6.157	1123.5	0.6
[[3, 1, 1], [3, 0, 2]]	18.4%	0.705	128.7	9.2	[[6, 2, 3], [0, 2, 1]]	128.4%	4.557	831.5	90.1
[[3, 1, 1], [3, 0, 0]]	18.4%	0.703	128.4	31.4	[[1, 2, 0], [0, 2, 0]]	143.2%	4.959	905.0	1.5

For Café dataset (Table 8.6), differencing of order 1 for non-seasonal component seems to be beneficial, with other parameters approximately same as for Thai restaurant. Also results for this datasets show significantly better performance then naive forecast (see MASE).

Table 8.7: Gelato: Grid search results showing best and worst parameters

(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)	(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)
[[3, 0, 2], [3, 2, 2]]	50.8%	0.841	160.4	49.2	[[6, 2, 3], [0, 2, 2]]	155.1%	3.976	758.8	99.4
[[6, 0, 0], [0, 2, 2]]	60.3%	0.841	160.5	5.2	[[6, 2, 3], [3, 2, 1]]	117.0%	3.097	591.0	107.4
[[2, 2, 1], [0, 1, 1]]	44.9%	0.841	160.5	2.7	[[6, 2, 3], [0, 1, 1]]	94.3%	4.957	945.8	211.8

Particularly for Gelato dataset (Table 8.7), MAPE is not considered to be best error measure. Scale of this dataset with combination of some values being distinctively small causes MAPE to be substantially high. Instead, MAE was used to evaluate this dataset. Differencing of greater order is used.

Table 8.8: Sushi restaurant: Grid search results showing best and worst parameters

(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)	(p,d,q), (P,D,Q)	MAPE	MASE	MAE	Time (s)
[[4, 0, 2], [2, 2, 2]]	25.1%	0.754	432.7	46.2	[[1, 2, 0], [1, 1, 0]]	81.9%	1.993	1142.9	1.6
[[1, 1, 4], [2, 1, 0]]	28.1%	0.798	457.5	8.3	[[6, 2, 4], [1, 2, 0]]	83.3%	1.53	877.6	43.3
[[1, 0, 3], [2, 2, 1]]	28.2%	0.8	459.0	16.5	[[2, 2, 0], [1, 2, 0]]	96.4%	2.495	1431.1	2.5
[[3, 0, 1], [2, 2, 1]]	28.3%	0.806	462.5	14.9	[[6, 2, 3], [1, 2, 1]]	338.5%	3.432	1968.6	68.3

8.8.2 Results

Optimized parameters from previous section were used to evaluate model on all datasets. Table 8.9 provides summary of achieved forecasting error.

Table 8.9: SARIMA final results with parameters optimized

Dataset	MAPE	SMAPE	MAE	μ_x
Nail Salon	16.72%	0.75	$218.5	$1356.2
Café	18.5%	0.75	$128.8	$729.0
Gelato	42.23%	0.76	$128.8	$374.2
Deli	15.04%	0.71	$145.3	$1047.6
Sushi Restaurant	31.18%	0.83	$444.3	$2065.2
Dry Cleaners	37.8%	0.77	$177.0	$580.6
Thai Restaurant	18.11%	0.74	$146.9	$782.5
Aggregated	25.65%	0.76	$198.5	$990.8

By consultation with domain experts, MAPE $\leq 25\%$ can be acceptable error for customers. SMAPE is significantly ≤ 1, which means results are significantly better than naive forecast. Gelato and Dry Cleaners are datasets, which exhibits higher error than others. More investigation has to be made to address this issue.

Autocorrelation function of residuals should show no autocorrelation above critical value. Indeed, when plotted residuals of test set (which was not used for learning phase), they exhibits autocorrelation below critical value. Example forecast can be seen in Figure 8.10.

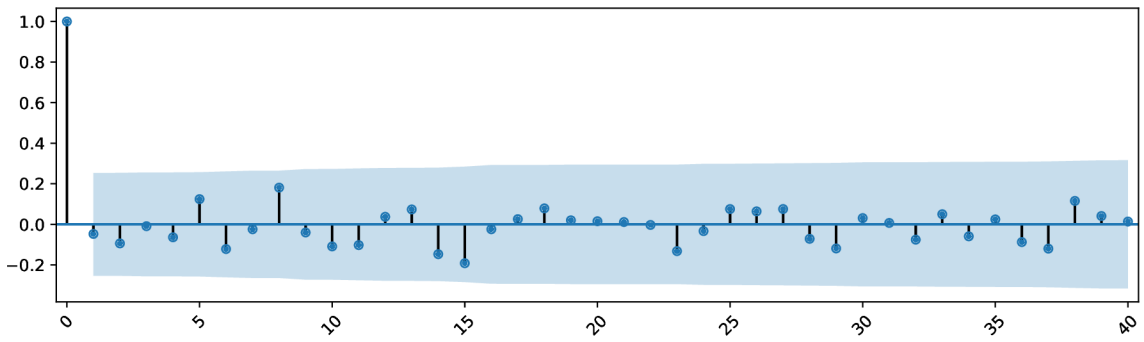


Figure 8.9: Nail Salon: autocorrelation plot of residuals

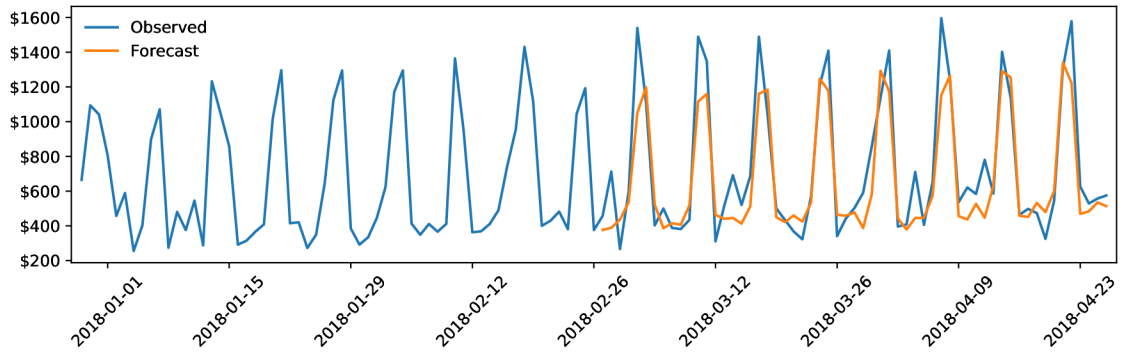


Figure 8.10: Café: SARIMA weekly forecast over period of two months

8.9 Model 3 - Neural Networks

Neural network (ANN) is model using network of artificial neurons, also called perceptrons. Neural networks can have difficulties with linear processes under some circumstances (noise ratio, amount of data), but generally well approximates non-linear processes, which is not possible with linear models (like ARIMA) [26].

Because of above mentioned, they are suggested for investigation, to model non-linear correlation, possibly with weather data. In this manner, they are evaluated as an alternative for classical models as ARIMA or Decomposition model.

Architecture of ANN consists of interconnected layers. In case of classical approach, called **Feed forward neural network**, each neuron is connected to all neurons of previous layer (see Figure 8.11).

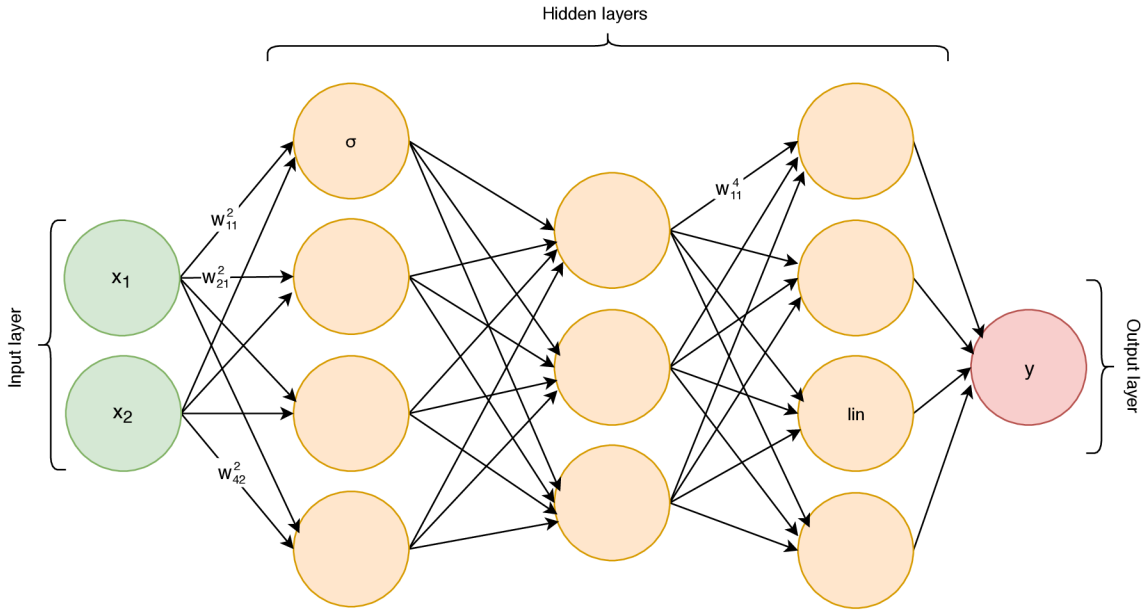


Figure 8.11: Feed forward ANN architecture

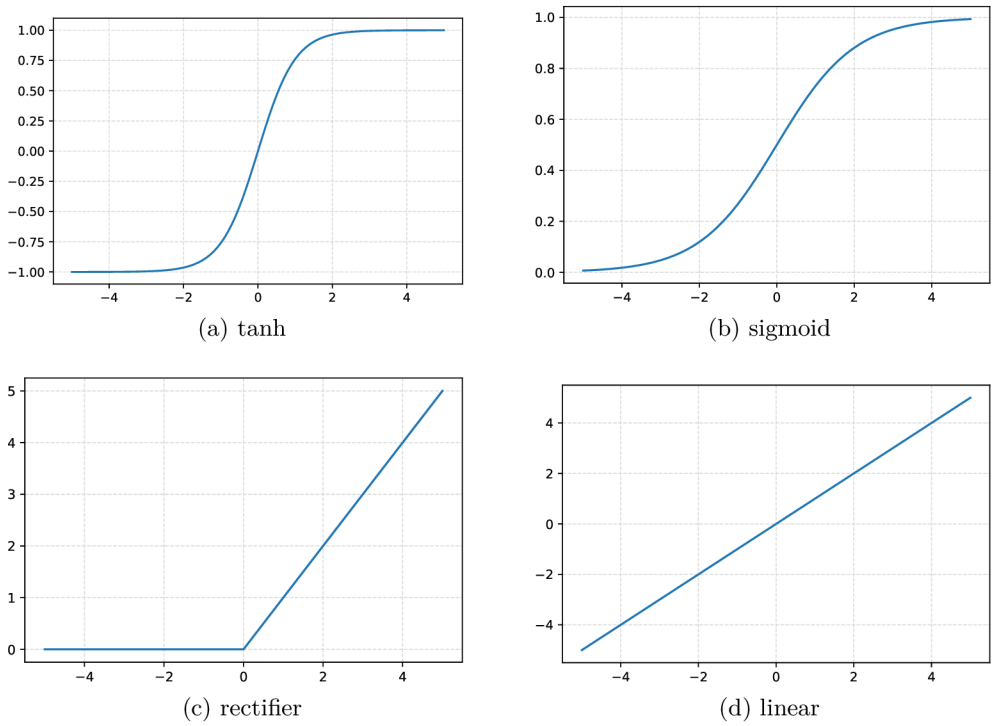
For j-th neuron in layer l is computed it's output (activation) a^l_j based on weighted input of neuron z^l_j. Weighted input is computed by entering activations of previous layer

and it's bias b_h^l into activation function σ:

$$z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l \tag{8.22}$$

$$a_j^l = \sigma(z_j^l) \tag{8.23}$$

Dependant on specific task, different activation function can be used for different layers. Most often used are sigmoid function, Hyperbolic Tangent (tanh), Rectifier (ReLu) and linear function (see Figure 8.12). Activation function, together with multi-layer architecture of neural networks, allows to model non-linear relations.



(a) tanh (b) sigmoid

(c) rectifier (d) linear

Figure 8.12: Examples of activation functions

Process of learning is enabled by fitting weights of neural network to provided data. In case of Feed forward neural networks, this process is called **back-propagation**. For that, notion of cost function C has to be defined. It is target function which is supposed to be minimized. For purpose of cost function is typically used some kind of error measurement, very often it is mean-squared error:

$$C = \frac{1}{2n} \cdot \sum_x ||y(x) - \hat{y}(x)||^2 \tag{8.24}$$

Output of the neural network from output layer is entered into the cost function, the result is then **backpropagated** in reverse order, gradually through individual layers. Intermediate product for computing weights and biases is error of the neuron:

$$d_j^l = \frac{\partial C}{z_j^l}, \tag{8.25}$$

Gradient of the connection weight is related to value of activation it connects, simply, more it participated in given error, more it is adjusted:

$$\frac{\partial C}{w_{jk}^l} = a_k^{l-1}\delta_j^l, \tag{8.26}$$

Analogically, bias equals, since it represents the error itself on the given neuron, to:

$$\frac{\partial C}{w_{jk}^l} = \delta_j^l, \tag{8.27}$$

For practical computations, **stochastic gradient decent** is used, it has parameter η, called learning rate, which adjust how quickly are weights changed:

$$w_{ij}(t+1) = w_{ij}(t) + \eta\frac{\partial C}{\partial w_{ij}} + \xi(t), \tag{8.28}$$

where $\xi(t)$ is stochastic term. In practical algorithms δ_j^l is computed from already known values and only derivations of C and σ are needed. Further, many improved optimization algorithms were developed. Among other it is Adam, Adagrad, SGD, RMSprop etc., but description of these is behind limits of this thesis.

Choice of optimization algorithm is one of the important parameters, which has to be chosen prior to archive correct results. Number of neurons, activation function and also optimization algorithm are subject of hyperparameter optimization through genetic algorithm described bellow.

8.9.1 Published papers

Many papers were published referring comparison of different types of ANN as representatives of more recent approaches in contrast to classical ARIMA based models. Further, some results from this research are presented.

In forecasting precipitation in Kenya was TLNN (Time-lagged Neural network) found superior to ARIMA. [27]. Also, when predicting stock market, TLNN supplied with ten variables (e.g. opening price, daily low price...) lagged by two days, outperformed ARIMA model [28].

For electricity consumption in Turkey, Feed-forward ANN also shown better results in comparison with ARIMA[9], different ANN architectures were evaluated, as well as different input data supplied, namely combinations of calendar data, previous load estimation plan, electricity, price, weather data, and currency. Interestingly, when supplied less input variables, namely only calendar and load data, performance increased.

In the other case, forecasting monthly streamflow of Kizil River in China resulted approximately in same performance for ARIMA and Jordan-Elman ANN [10]. In this case, ARIMA was preferred, since this model allows better interpretability. [29]

Long short-term memory neural network (LSTM) is type of network, which consists of some kind of memory, which is persisted across data in sequential manner. It can model advanced patterns, where some event in the past can affect output in distant future, downside of the approach is need for large amount of data to learn this patterns, also,

[9] ANN MAPE: 1.8%, ARIMA MAPE: 2.6%

[10] Simple type of recurrent neural network

LSTM can experience difficulties with learning more straightforward correlations, like auto-regression. So, use for some types of time processes, particularly with strong auto-regressive behavior is questionable [32]. For this reason, more straightforward approach to model auto-correlation was chosen.

Further, it will be investigated, whether above mentioned results can apply also for sales prediction for small business. This objective will be archived by building and evaluating model using **Feed-forward network with windowed input** in **Keras** library.

8.9.2 Encoding data

Time context is introduced by providing time lagged input to Feed forward neural network[11]. Time lag is defined similarly to differencing in in Section 8.8 .

$$\begin{aligned} \delta x_t &= x_{t-1} \\ \delta_n x_t &= x_{t-n} \end{aligned} \tag{8.29}$$

In this fashion, it can be defined lagged input $\mathbf{x_t}$ of the sampled time series x_t according to lag vector $\mathbf{l}$:

$$\begin{aligned} \mathbf{l} &= (l_1 \dots l_k \mid \forall j \leq k : \; l_j \geq 0, j \in N, l_j \in N) \\ \mathbf{x_t} &= (\delta_{l_k} x_t \mid \forall l_k \in \mathbf{l}) \text{ for } t > max \; l_k \in \mathbf{l} \end{aligned} \tag{8.30}$$

It should be noted, that this transformation removes maximal lag $max \; l_k \in \mathbf{l}$ observations from the begging of the series, since for them is not possible to compute their lagged input.

Also, data has to be preprocessed for neural networks. First, categorical data are encoded as integers by `LabelEncoder` from scikit-learn. Then, **one-hot approach** (or dummy variable approach) is applied. Categorical attributes are represented in binary format and every bit is encoded as single attribute. Last dummy variable is removed to prevent so called "dummy variable trap". Categorical data are automatically detected and above mentioned transformation applied to it.

Additionally, attributes has to be normalized to range [0,1] to fit activation functions. Scikit-learn contains class `MinMaxScaler`, but is not easy to use and also not integrated with Pandas, so custom class `Scaler` was developed.

Different combinations of parameters were manually estimated in relation to number of inputs and output and their types, but as results were not satisfiable, another solution was approached.

8.9.3 Implementation

Keras library was used to implement neural network model. Keras is designed to use several backends to perform computionaly intensive tasks. Backends are using GPU acceleration to perform fast parallel computations. For purpose of this thesis, **TensorFlow** backend was chosen, since it is most popular, thus having good community support and also support in cloud services.

First, arbitrary neural network architecture was suggested with mean_squared_error loss function and adam optimizer according to general guidelines:

[11] Alternatively called Multilayer perceptron (MLP)

```
model = Sequential()
model.add(
  Dense(
    12,
    activation="sigmoid",
    input_dim=train_X.shape[1]
  )
)
model.add(
  Dense(
    5,
    activation="sigmoid",
  )
)
model.add(Dense(1, activation='linear'))
```

This design had very poor performance with with MAPE $\tilde{6}0\%$. Because of that, as in previous cases, hyper-parameter optimization method has to be used. In case of neural network, **genetic algorithm** was chosen [37]:

> The genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population „evolves“ toward an optimal solution. You can apply the genetic algorithm to solve a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, nondifferentiable, stochastic, or highly nonlinear. The genetic algorithm can address problems of mixed integer programming, where some components are restricted to be integer-valued.
>
> - The genetic algorithm uses three main types of rules at each step to create the next generation from the current population:
> - Selection rules select the individuals, called parents, that contribute to the population at the next generation.
> - Crossover rules combine two parents to form children for the next generation.
> - Mutation rules apply random changes to individual parents to form children.

Population was chosen to be 30, generations to be 20 and genom (individual) was encoded in this way:

```
'neurons': [5, 8, 16, 32, 64,]
'layers': [1, 2, 3, 4]
'activation': ['relu', 'elu', 'tanh', 'sigmoid', 'hard_sigmoid', 'softplus']
'optimizer': ['rmsprop', 'adam', 'sgd', 'adagrad', 'adadelta', 'adamax', 'nadam']
```

In case of number of layers ≥ 2, consequent layers has half number of the neurons of the previous layer. Last linear layer with one neuron is implicit. EC2 instance **p2.xlarge** with NVIDIA Tesla K80 Accelerator GPU, containing 2,496 parallel processing cores and 12 GiB of memory (accessible via 240 GB/second of memory bandwidth).

Name	Neurons in first layer	Hidden layers	Activation	Optimizer
Gelato	5	1	relu	rmsprop
Thai Restaurant	32	3	relu	adadelta
Café	5	1	relu	rmsprop
Sushi Restaurant	32	2	relu	adadelta
Deli	64	2	relu	rmsprop
Dry Cleaners	8	1	relu	rmsprop
Nail Salon	16	1	relu	adadelta

Table 8.10: Parameters optimized by genetic algorithm

With above generated neural architecture (Table 8.10), model can be retrained to better fit the data. Final output can be seen in Table 8.11.

Table 8.11: Final result for neural network model

Dataset	MAE	MAPE	SMAPE	MAE	μ_x
Nail Salon	16.39%	16.9%	0.59	$228.3	$1343.3
Café	19.7%	20.34%	0.81	$140.4	$736.4
Gelato	35.3%	36.11%	0.56	$139.8	$376.1
Deli	18.17%	16.32%	0.78	$164.8	$1045.0
Sushi Restaurant	36.68%	27.63%	1.0	$490.9	$2061.6
Dry Cleaners	45.25%	35.71%	0.82	$195.1	$574.1
Thai Restaurant	33.86%	28.18%	0.95	$224.9	$793.1
Mean	29.34%	25.88%	0.79	$226.3	$990.0

8.10 Model 4 - TSMARS

Time series for MARS (TSMARS) is variation of MARS (Multivariate adaptive regression splines) applied to time series problem. MARS is non-parametric general-purpose model, which is able to automatically model non-linearities from given data. It also implicitly perform variable-selection by removing non-relevant variables (see bellow) [30].

8.10.1 Model description

MARS model is defined by:

$$\hat{f}(x) = \sum_{i=1}^{k} c_i B_i(x), \tag{8.31}$$

where c_i are constants and B_i being basis functions, which are combined into resulting model. Basis function can be:

1. constant
2. a hinge function, in form of $\max(0, x - const)$ or $\max(0, const - x)$
3. product of two or more hinge function, which are used to model association between variables

By combining hinge functions, it is possible to approximate course of target function by piecewise linear function (see Figure 8.13), but also achieve non-linearity by multiplication of the terms. The point, where both complementary hinge functions intersects, is called *knot*.

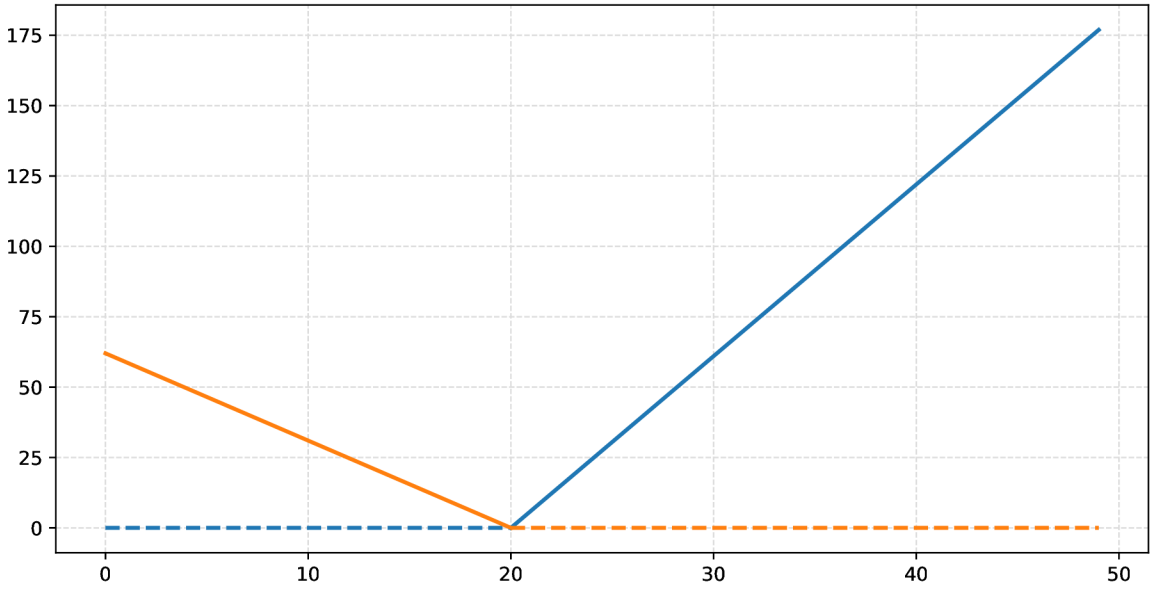


Figure 8.13: Linear combination of hinge functions: $y = 6.1 \cdot max(0, x-20) + 3.1 \cdot max(0, 20-x)$

The model is suitable for general non-linear regression problems, but it is also possible to use the model for time series modeling. This variant is known as Time series for MARS (TSMARS). It models time series by using lagged values of the series as an input for the model.

8.10.2 Algorithm

Algorithm for building the model equation (Equation 8.31) is divided into two steps – forward pass and backward pass. In forward pass, algorithm generates new terms of the equation in a greedy fashion. In backward pass, it optimizes the equation and removes terms to prevent overfitting.

First, only intercept term is introduced by computing mean value of the input. Then, new terms are repeatedly generated by multiplying new complementary hinge functions with existing term, which can be also intercept term. Hinge functions, which best minimizes sum-of-squares residual error is chosen. Generating new terms is performance intensive task, since several parameters can be taken into account:

1. Choice of existing term to multiply with
2. Variable to generate new hinge functions from
3. Value of the variable to be chosen as a knot

MARS uses linear regression to calculate the coefficients. When defined residual error or allowed number of terms is reached, the first step is terminated and it is approached to second step.

In the second step – backward pass – terms are evaluated one by one by their effectiveness. For that, **Generalized cross validation** (GCV) is used. GCV uses **residual**

sum-of-squares (RSS) as error metrics, which is supposed to be minimized, with additional penalization of number of terms, since without the penalization, algorithm would provide overfit model with large amount of terms. The name *generalized* cross validation references to general technique – cross validation, which evaluates model by leaving out portion of the data, which is then used for validation (see Section 8.6). GCV approximates this procedure by penalizing the amount of terms, it has the form of:

$$GCV = \frac{RSS}{(N \cdot (1 - \frac{n_p}{N})^2)}, \tag{8.32}$$

where n_p is effective number of parameters calculated as:

$$n_p = n_t + p * \frac{n_t - 1}{2}, \tag{8.33}$$

where n_t is number of terms and p is penalty parameter. The second part, $\frac{n_t - 1}{2}$, of equation 8.33, relates to number of knot parameters, since every knot describes two hinges and intercept parameter does not have a knot.

8.10.3 Modeling time series

Py-earth is an implementation of MARS for Python[12] [31]. It will be used to model the given datasets and to investigate hypothetical non-linear correlation between sales and weather data. Testing set was chosen to be 60 days, with weekly prediction made for this period.

First, default linear model (Table 8.12a) with lagged input is tested. Lags provided for the model are 6, 7, 14 and 13.

Dataset	MAPE	SMAPE	MASE	MAE
Nail Salon	17.52%	17.44%	0.8	$230.8
Café	20.38%	20.32%	0.8	$137.5
Gelato	42.82%	40.73%	0.86	$145.2
Deli	17.82%	17.18%	0.85	$174.7
Sushi Rest.	30.47%	23.81%	0.82	$438.4
Dry Cleaners	41.72%	35.27%	0.84	$190.8
Thai Rest.	26.79%	23.35%	0.9	$177.7
Mean	28.22%	25.44%	0.84	$213.6

(a) Nail Salon

Dataset	MAPE	SMAPE	MASE	MAE
Nail Salon	17.1%	17.54%	0.81	$234.9
Café	22.62%	22.74%	0.85	$145.6
Gelato	41.87%	43.22%	0.84	$142.8
Deli	16.51%	15.61%	0.78	$158.9
Sushi Rest.	31.53%	24.6%	0.83	$445.6
Dry Cleaners	40.9%	35.28%	0.82	$187.6
Thai Rest.	30.04%	25.21%	0.94	$185.6
Mean	28.65%	26.31%	0.84	$214.4

(b) Thai Restaurant

Table 8.12: Preliminary results for MARS model

Second step was to increase order of the model to be able to model interaction between variables and provide weather data mentioned in Table 8.2.

Results in Table 8.12b that increase in order and provided weather did not improve results. In attemt to improve model performance, similar hyper-parameter optimization method by genetic algorithm to Section 8.9.3 was employed. Following parameters are optimized:

[12]many open-source implementation are called Earth due to fact, that MARS is registered trade mark

1. max degree $\in \{1 \dots 2\}$: the maximal polynomial degree
2. penalty $\in \{1 \dots 4\}$: a coefficient penalizing additional terms
3. minspan$_\alpha \in$ [0.1,1]: the probability from which is inferred minimal number of data points between knots
4. maxspan$_\alpha \in$ [0.1,1]: the probability from which is inferred number of extreme data values of each feature not eligible as knot locations.

Parameters	MAPE	MASE
(2, 3, 0.01, 0.2)	16.2%	0.75
(2, 3, 0.01, 0.2)	16.2%	0.75
(2, 2, 0.61, 0.30)	16.3%	0.734
(2, 4, 0.01, 0.6)	16.4%	0.755
(2, 4, 0.01, 0.70)	16.4%	0.755
(2, 4, 0.01, 0.8)	16.4%	0.755

(a) Nail Salon

Parameters	MAPE	MASE
(1, 3, 0.01, 0.1)	23.0%	0.879
(1, 3, 0.01, 0.1)	23.0%	0.879
(1, 2, 0.01, 0.1)	24.1%	0.921
(1, 2, 0.01, 0.1)	24.1%	0.921
(1, 4, 0.01, 0.1)	23.2%	0.88
(1, 4, 0.01, 0.1)	23.2%	0.88

(b) Thai Restaurant

Table 8.13: Examples of genetic algorithm results for MARS model

Genetic algorithm outcomes for two chosen datasets is available in Table 8.13. From results follows, that optimized parameters improved performance of the model.

8.10.4 Results

Final results shows table 8.14. Error was reduced in with comparison to SARIMA model.

Table 8.14: Final results for MARS model

Dataset	MAPE	SMAPE	MAE	μ_x	
Nail Salon	16.43%	16.8%	0.58	\$223.9	\$1343.3
Café	19.72%	20.4	0.81	\$142.0	\$736.4
Gelato	40.91%	41.6%	0.59	\$145.6	\$376.1
Deli	14.57%	13.3%	0.64	\$134.6	\$1045.0
Sushi Restaurant	32.92%	25.3%	0.96	\$472.1	\$2061.6
Dry Cleaners	35.9%	36.2%	0.79	\$186.9	\$574.1
Thai Restaurant	32.31%	25.5%	0.85	\$200.5	\$793.1
Mean	27.54%	25.6%	0.75	\$215.1	\$990.0

In Figure 8.14 is displayed variable importance for one of the datasets. Following measurement attempt to interpret influence of the input variables to the model output. Three metrics are available – RSS, GCV and nb_subsests. The first one is computed from effect on root squared error during the initial phase of the algorithm. The second one measures influence on generalizes cross-validation and the third is inferred from number of terms in which is the variable present.

Most important variable is lagged time series, particularly 7 day lag, which corresponds to week period. The weather variable is also present, but it is removed during second phase of the algorithm in favor of other variables.

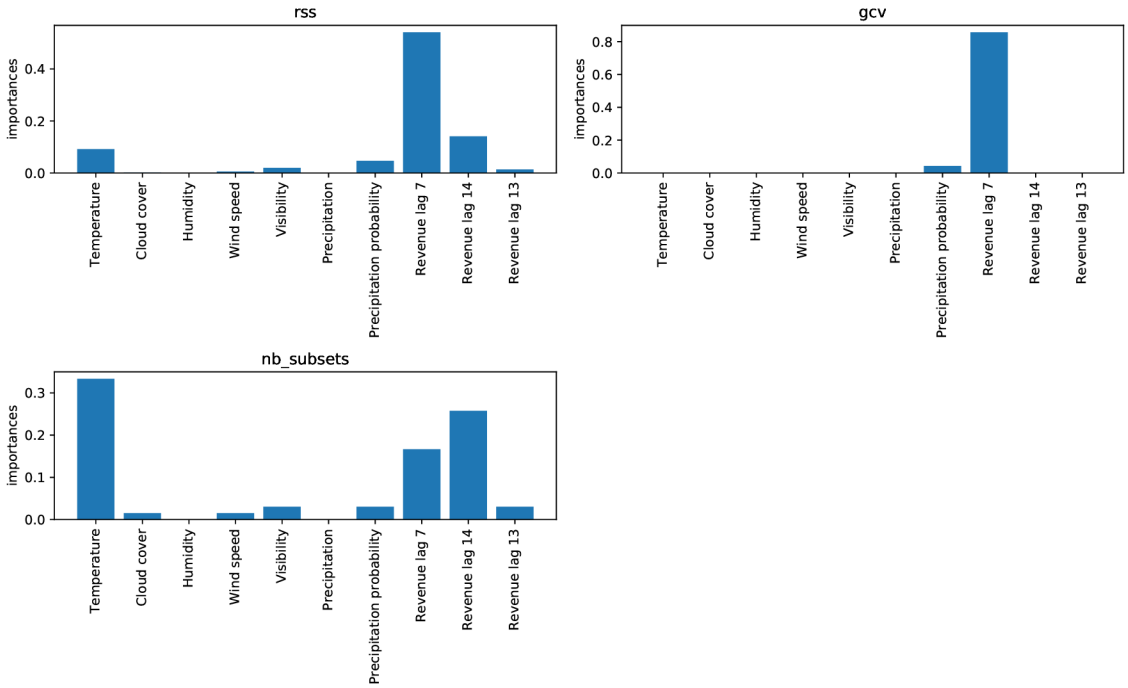


Figure 8.14: Nail Salon: Variable importance

8.11 Evaluation

Table 8.15 shows aggregated results for the three evaluated models. Decomposition model was omitted, since it was included mainly for comparison and exhibits poor results. Models have approximately similar results, although SARIMA is slightly superior. It is interesting, since other two models have greater capacity to model complex processes.

Dataset	ANN	SARIMA	MARS
Nail Salon	16.39%	16.72%	16.43%
Café	19.70%	18.50%	19.72%
Gelato	35.30%	42.23%	40.91%
Deli	18.17%	15.04%	14.57%
Sushi Restaurant	36.68%	31.18%	32.92%
Dry Cleaners	45.25%	37.80%	35.90%
Thai Restaurant	33.86%	18.11%	32.31%
Mean	29.34%	25.65%	27.54%

(a) MAPE

Dataset	ANN	SARIMA	MARS
Nail Salon	16.90%	16.44%	16.84%
Café	20.34%	19.16%	20.49%
Gelato	36.11%	35.51%	41.61%
Deli	16.32%	14.30%	13.37%
Sushi Restaurant	27.63%	24.36%	25.31%
Dry Cleaners	35.71%	32.40%	36.24%
Thai Restaurant	28.18%	19.37%	25.54%
Mean	25.88%	23.08%	25.63%

(b) SMAPE

Dataset	ANN	SARIMA	MARS
Nail Salon	0.59	0.75	0.58
Café	0.81	0.75	0.81
Gelato	0.56	0.76	0.59
Deli	0.78	0.71	0.64
Sushi Rest.	1.00	0.83	0.96
Dry Cleaners	0.82	0.77	0.79
Thai Rest.	0.95	0.74	0.85
Mean	0.79	0.76	0.75

(c) MASE

Dataset	ANN	SARIMA	MARS
ANN	SARIMA	MARS	
Nail Salon	$228.30	$218.50	$223.90
Café	$140.40	$128.80	$142.00
Gelato	$139.80	$128.80	$145.60
Deli	$164.80	$145.30	$134.60
Sushi Restaurant	$490.90	$444.30	$472.10
Dry Cleaners	$195.10	$177.00	$186.90
Thai Restaurant	$224.90	$146.90	$200.50
Mean	$226.31	$198.51	$215.09

(d) MAE

Table 8.15: Comparison of evaluated models

Clearly, the primary information obtainable from the data is auto-correlation, which can be well utilized by SARIMA. The other two models were much harder to develop, much more effort to optimize parameters through genetic algorithm had to be made. Regarding neural networks, during the training, more problems occurred, namely the fact, that they work in stochastic way. During optimization by genetic algorithm, encoded parameters for the model resulted in different outcomes, when running the model. In some cases error was substantially superior, in some cases were drastically inferior. Neural networks can potentially improve performance, but more sophisticated framework for their use in this context has to be developed.

Chapter 9

Conclusion

This thesis had as an objective to evaluate techniques of data mining for use in small business. By examining data and consultations with domain experts, two approaches were chosen: association rules mining and time series forecasting.

Both approaches exhibit potential to future improvement and use in production application. Several challenges are present in use of the data mining in small business.

First, procedure has to be automated to some degree, since modeling is made as a part of software solution, which non-expert is used by non-experts.

Secondly, amount of data can be insufficient, since new clients are expecting outcomes in short time after start of use of a software and also amount of data in small business is substantially lower.

One of the solution could be to use other client data and by statistical comparing of the time series choose proper complementary datasets, which would be use as an input for model for the particular client.

Satisfiable results were obtained for some datasets by optimizing model, although providing weather data for use of non-linear model did not improve accuracy. More historical data as well as data from more businesses is needed. Suggested solution is to automatically evaluate error of given business and based on the result show or hide the feature, so only useful insights will be presented.

As consequence of this thesis, experimental feature was integrated into the POS. Program developed as part of this thesis is capable of running as a web services, which communicates with the main application through API and provides forecasting data. By author of this thesis was also developed frontend component displaying the acquired data.

Bibliography

[1] Hyndman, R. and Athanasopoulos, G. (2015). Forecasting principles and practice. USA: O texts.

[2] "Top 20 Most Popular POS Software Infographic." Capterra Blog 5 Ways to Improve Communication with Your Remote Teams Comments, blog.capterra.com/top-20-most-popular-pos-software/.

[3] "KDnuggets." KDnuggets Analytics Big Data Data Mining and Data Science, www.kdnuggets.com/2017/01/most-popular-language-machine-learning-data-science.html.

[4] "Forecast." Merriam-Webster, Merriam-Webster, www.merriam-webster.com/dictionary/forecast.

[5] Arndt, Holger. "Java Data Mining Package | Machine Learning and Big Data Analytics." Java Data Mining Package | Machine Learning and Big Data Analytics, jdmp.org/.

[6] "3.1 Introduction." 4.4 Evaluating the Regression Model | OTexts, www.otexts.org/fpp/3/1.

[7] Nazar, Jason. "16 Surprising Statistics About Small Businesses." Forbes, Forbes Magazine, 30 June 2014, www.forbes.com/sites/jasonnazar/2013/09/09/16-surprising-statistics-about-small-businesses/#248f86a45ec8.

[8] Nazar, Jason. "16 Surprising Statistics About Small Businesses." Forbes, Forbes Magazine, 30 June 2014, www.forbes.com/sites/jasonnazar/2013/09/09/16-surprising-statistics-about-small-businesses/#248f86a45ec8.

[9] Tan, Pang-Ning, et al. Introduction to Data Mining. Pearson Education, 2005.

[10] Han, J., Kamber, M., & Pei, J. (2012). Data mining: concepts and techniques. Amsterdam: Morgan Kaufmann, an imprint of Elsevier.

[11] "Can I Use WEKA in Commercial Applications?" Weka, weka.wikispaces.com/Can I use WEKA in commercial applications?

[12] Ulanoff, Lance. "Amazon Knows What You Want Before You Buy It." Predictive Analysis Times, 27 Jan. 2014, www.predictiveanalyticsworld.com/patimes/amazon-knows-what-you-want-before-you-buy-it/3185/.

[13] Apache Mahout, mahout.apache.org.

[14] "Stack Overflow Developer Survey 2016 Results." Stack Overflow, insights.stackoverflow.com/survey/2016#technology.

[15] "TenMinuteTutor." Numpy Efficiency, 10 May 2017, www.tenminutetutor.com/python/numpy/numpy-efficiency/.

[16] „François Chollet on Twitter". Retrieved 2016-09-18.

[17] Quora. "This Is What Makes Keras Different, According To Its Author." Forbes, Forbes Magazine, 25 Aug. 2016, www.forbes.com/sites/quora/2016/08/25/this-is-what-makes-keras-different-according-to-its-author/#781ca2df66cf.

[18] Time Series Analysis: Forecasting and Control, by George E. P. Box et al., John Wiley,

2008.
[19] Brockwell, Peter J., and Richard A. Davis. Time Series: Theory and Methods. Springer-Verlag, 1991.
[20] "X-13ARIMA-SEATS Seasonal Adjustment." US Census Bureau, 22 Jan. 2013, www.census.gov/srd/www/x13as/.
[21] Willmott, Cj, and K Matsuura. "Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance." Climate Research, vol. 30, 2005, pp. 79–82., doi:10.3354/cr030079.
[22] Tofallis, Chris. "A Better Measure of Relative Prediction Accuracy for Model Selection and Model Estimation." Journal of the Operational Research Society, vol. 66, no. 8, 2015, pp. 1352–1362., doi:10.1057/jors.2014.103.
[23] Hyndman, R. J. and Koehler A. B. (2006). „Another look at measures of forecast accuracy." International Journal of Forecasting volume 22 issue 4, pages 679-688. doi:10.1016/j.ijforecast.2006.03.001
[24] "Jrmontag/STLDecompose." GitHub, github.com/jrmontag/STLDecompose.
[25] "Stl - Seasonal Decomposition Of Time Series By Loess." Function | R Documentation, www.rdocumentation.org/packages/stats/versions/3.4.3/topics/stl.
[26] Zhang, G.Peter. „Time Series Forecasting Using A Hybrid ARIMA And Neural Network Model". Neurocomputing, vol 50, 2003, pp. 159-175. Elsevier BV, doi:10.1016/s0925-2312(01)00702-0.
[27] Kimani Karuiru, Elias. „Sarima Versus Time Lagged Feedforward Neural Networks In Forecasting Precipitation". American Journal Of Theoretical And Applied Statistics, vol 5, no. 6, 2016, p. 359. Science Publishing Group, doi:10.11648/j.ajtas.20160506.15.
[28] Adebiyi, Ayodele Ariyo et al. „Comparison Of ARIMA And Artificial Neural Networks Models For Stock Price Prediction". Journal Of Applied Mathematics, vol 2014, 2014, pp. 1-7. Hindawi Limited, doi:10.1155/2014/614342.
[29] Abudu, Shalamu et al. „Comparison of performance of statistical models in forecasting monthly streamflow of Kizil River, China". Water Science and Engineering, vol 5, no. 6, 2010, pp. 269-281. doi:10.3882/j.issn.1674-2370.2010.03.003.
[30] Wikipedia contributors. „Multivariate adaptive regression splines." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 18 Mar. 2018. Web. 19 May. 2018.
[31] Contributors. "Scikit-Learn-Contrib/Py-Earth". Github, https://github.com/scikit-learn-contrib/py-earth. Accessed 19 May 2018.
[32] Brownlee, Jason. „On The Suitability Of Long Short-Term Memory Networks For Time Series Forecasting". Machine Learning Mastery, 2018, https://machinelearningmastery.com/suitability-long-short-term-memory-networks-time-series-forecasting/. Accessed 20 May 2018.
[33] Aggarwal, Charu et al. Frequent Pattern Mining. Springer International, 2016.
[34] Agrawal, Rakesh and Srikant, Ramakrishnan. „Fast algorithms for mining association rules". Proceedings of the 20th International Conference on Very Large Data Bases, 1994, pp. 487-499.
[35] Modi1, Anitha and Krishnan, Radhika. „Mining Frequent Itemsets in Transactional Database". International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 2, February 2013, pp. 170-174.
[36] Kirsch, Adam et al. „An Efficient Rigorous Approach For Identifying Statistically Significant Frequent Itemsets". Journal Of The ACM, vol 59, no. 3, 2012, pp. 1-22. Association For Computing Machinery (ACM), doi:10.1145/2220357.2220359.

[37] „What Is The Genetic Algorithm?- MATLAB & Simulink“. Mathworks.Com, 2018, https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html. Accessed 25 May 2018.

Appendix A

Database structure

Table A.1: **account**: presenting account of particular business

Column Name	Data Type	Description
id	character varying(36)	
name	character varying(100)	Display account name

Table A.2: **order**: unit of purchasing action, containing one or more items

Column Name	Data Type	Description
id	character varying(36)	
customer_id	character varying(36)	Reference to customer
account_id	character varying(36)	Reference to account withing which order was created
location_id	character varying(36)	Reference to branch where order is placed
assignment_id	character varying(36)	Reference to assignment where customer was served
assignee_id	character varying(36)	Reference to employee serving the customer
created_timestamp	timestamptz	When order was put into the system
total_price	bigint	Total price paid by the customer

Table A.3: **order_item**: particular item representing purchasable object

Column Name	Data Type	Description
id	character varying(36)	
category_id	character varying(36)	Reference to defined product in category entity
name	character varying(60)	Display item name (redundancy from category entity)
order_id	character varying(36)	Reference to associated order
quantity	bigint	Number of purchased items
option_id	character varying(36)	Reference to option in category entity, defining variation of product
option_name	character varying(60)	Display variation name of item
total_price	bigint	Total price of items with given count
special_pricings_total	bigint	Total sum with modification (discounts etc.)
type	character varying(45)	Type of pricing - onetime, normal or custom-priced

Table A.4: **category**: entity representing all defined category, option and product

Column Name	Data Type	Description
id	character varying(36)	
name	character varying(60)	Display name of option / product / category
type	character varying(45)	One of option, product or category
parent_id	character varying(36)	Reference for parent assignment allowing nesting

Table A.5: **customer**: customer making an order

Column Name	Data Type	Description
id	character varying(36)	
name	character varying(100)	Display name
deleted	boolean	Soft delete flag

Table A.6: **user_account**: account used for customers to log into systems

Column Name	Data Type	Description
id	character varying(36)	
name	character varying(100)	Display name
deleted	boolean	Soft delete flag

Table A.7: **transaction**: customer making an order

Column Name	Data Type	Description
id	character varying(36)	
account_id	character varying(36)	Reference to account withing which associated order belongs
order_id	character varying(36)	Reference to associated order
accounting_type	character varying(15)	debit - negative amount or credit - positive amount
transaction_type	character varying(15)	type of transaction - one of refund, nosale, pay_out, tip, pay_in, change, payment or void
state	character varying(25)	state of transaction - one of pending, processed or voided
payment_type	character varying(15)	method transfered amount - external, card or cash
amount	bigint	amount of transfer
amount_refunded	bigint	amount of refunded money
amount_tip	bigint	tip amount
amount_tip_refunded	bigint	amount if refunded tip
origin_timestamp	timestamptz	timestamp of creation

Table A.8: **location**: every account can have several locations, representing e.g. particular branches of chain

Column Name	Data Type	Description
id	character varying(36)	
account_id	character varying(36)	Reference to account
name	character varying(100)	Display location name
deleted	boolean	Soft delete flag

Table A.9: **assignment**: more granular category than location, represents physical place within branch

Column Name	Data Type	Description
id	character varying(36)	
name	character varying(100)	Display name
account_id	character varying(36)	Reference to account
parent_id	character varying(36)	Reference for parent assignment allowing nesting
deleted	boolean	Soft delete flag

Appendix B

Association rules

Table B.1: Nail Salon

items	p-value	lift	confidence	rel. supp	supp
Massage, 10 min	1.17643e-13	3.02609	66.3%	16.4%	57
Massage, Regular, 10 min	1.857e-15	3.61167	35.6%	14.9%	52
Regular, 10 min	0.168824	1.19351	32.2%	14.9%	52
Manicure, Massage, 10 min	1.74016e-12	3.11028	32.0%	14.4%	50
Manicure, Massage, Regular, 10 min	2.65392e-14	3.74356	25.0%	13.2%	46
Manicure, Regular, 10 min	0.1393	1.2371	22.3%	13.2%	46
Massage, 20 min	4.31987e-08	3.02609	43.5%	9.2%	32
Massage, Pedicure, Regular	0.496484	1.11128	19.4%	9.2%	32
Manicure, Vinylux	0.423876	1.17172	15.0%	6.9%	24
Massage, 30 min	5.92507e-06	3.02609	32.1%	6.3%	22
Massage, Pedicure, 10 min	1.63472e-06	3.4033	22.0%	6.0%	21

Table B.2: Thai Restaurant

items	p-value	lift	confidence	rel. supp	supp
Chicken, Pad Thai	0.0185374	1.4542	45.3%	14.7%	39
Chicken, Original Thai Fried Rice	0.140315	1.34234	32.2%	9.1%	24
Soup, Tom Kha	5.24375e-12	8.83333	75.0%	6.8%	18
Shrimp, Original Thai Fried Rice	0.000867194	2.49119	38.6%	6.0%	16
Shrimp, Pad Thai	0.0967944	1.55699	28.3%	5.7%	15
Ice Tea, Thai Beverage	1.29859e-11	12.619	80.0%	5.3%	14
Soda, Pad Thai	0.0460993	1.73446	28.0%	5.3%	14
Pad Thai, Original Thai Fried Rice	0.551863	1.16887	24.3%	5.3%	14
Coke, Soda	7.24398e-09	8.54839	59.1%	4.9%	13
Chicken, Pineapple Fried Rice	0.0661768	1.67233	21.1%	4.9%	13
Chicken, Drunk Man Noodle	0.333799	1.28641	20.2%	4.9%	13

Table B.3: Sushi Restaurant

items	p-value	lift	confidence	rel. supp	supp
Spicy Tuna Roll, California Roll	0.227848	1.23288	24.8%	5.1%	32
Spicy Tuna Roll, Miso Soup	0.383348	1.15068	24.1%	5.1%	32
Yama Special Salad, E. Jackpot Johnnie Roll	5.73467e-05	2.31405	31.8%	4.4%	28
Spicy Tuna Roll, Yama Special Salad	0.285968	1.22042	22.9%	4.4%	28
Tempura, California Roll	0.0157601	1.625	25.7%	4.1%	26
Tuna (Maguro),Salmon (Sake)	1.30599e-11	5.56144	46.7%	4.0%	25
Spicy Tuna Roll, Tempura	0.578122	1.10274	19.5%	3.7%	23
Spicy Tuna Roll, E. Jackpot Johnnie Roll	0.401535	1.17684	18.8%	3.3%	21
Yama Special Salad, Miso Soup	0.559298	1.11364	19.2%	3.3%	21
Spicy Tuna Roll, Salmon (Sake)	0.098132	1.46274	19.5%	3.2%	20
Spicy Tuna Roll, Yama Special Roll	0.135902	1.41478	19.3%	3.2%	20

Table B.4: Deli

items	p-value	lift	confidence	rel. supp	supp
Bagel, Plain	1.81042e-32	5.37692	55.6%	6.0%	79
02. Cream Cheese, Bagel	2.83385e-39	7.55491	59.3%	5.6%	73
Coffee, Large Iced	3.86899e-07	1.92206	18.9%	5.4%	71
01. Butter, Bagel	4.70906e-34	7.55491	53.4%	4.8%	63
Default, Egg And Cheese	3.23912e-49	29.0444	100.0%	3.4%	45
Sausage-Pork, McBasic	8.67906e-42	27.2292	89.7%	3.0%	39
Coffee, Bagel, Plain	2.9578e-15	4.97115	11.8%	2.9%	38
Coffee, Small Iced	0.000231745	1.92206	10.6%	2.9%	38
01. Butter, Bagel, Coffee	2.39783e-21	8.29771	11.8%	2.8%	36
Coffee, Muffin	0.156306	1.25807	9.8%	2.8%	36
02. Cream Cheese, Bagel, Plain	1.21731e-42	41.4322	28.6%	2.6%	34

Table B.5: Gelato

items	p-value	lift	confidence	rel. supp	supp
Per, Macaron	7.76647e-25	3.47273	88.9%	23.0%	88
Oreo Cookies & Cream, Rolling Gelato	5.20115e-13	3.97917	57.8%	10.2%	39
Gelato - Small, Cone	1.04294e-05	2.30879	38.4%	8.6%	33
Per, Gelato - Small, Macaron	9.30728e-07	3.01496	23.8%	6.8%	26
Box of 6, Macaron	7.08427e-06	3.47273	28.1%	4.7%	18
Gelato - Medium, Cone	0.0176685	1.8407	27.0%	4.5%	17
Rolling Gelato, 5. Chocolate Hazelnut	1.71759e-05	3.97917	25.5%	3.7%	14
Strawberry Cheesecake, Rolling Gelato	3.41344e-05	3.97917	23.9%	3.4%	13
Strawberry & Banana, Crepe	5.31265e-09	11.5758	50.0%	2.9%	11
Nutella & Banana, Crepe	5.31265e-09	11.5758	50.0%	2.9%	11
Rolling Gelato, Chocolate chip Cookie dough	0.000135315	3.97917	20.6%	2.9%	11

Table B.6: Cafe

items	p-value	lift	confidence	rel. supp	supp
Pastry, Coffee	0.347478	1.1672	25.3%	6.0%	35
Omelette, Veggie	1.20406e-21	10.0107	71.9%	5.5%	32
Chicken & Waffles, Waffles	5.58368e-22	10.8333	72.9%	5.3%	31
Eggs Special, in HELL	9.34824e-25	15.3947	86.6%	5.0%	29
Coffee, Large - Hot	0.0484897	1.45251	23.1%	4.8%	28
Bubble Tea, Taro	6.04786e-17	9.75	58.8%	4.3%	25
Latte, Large - Hot	0.00063676	2.09138	28.7%	4.3%	25
Omelette, Latte	0.000485955	2.20393	27.7%	3.9%	23
w/ Seasonal Fruit, Waffles	2.48905e-15	10.8333	56.0%	3.6%	21
Tea, Large - Hot	8.01269e-08	4.14894	38.2%	3.6%	21
Pastry, Cream Cheese	1.17905e-09	5.96939	32.5%	3.2%	19

Table B.7: Nails Salon 2

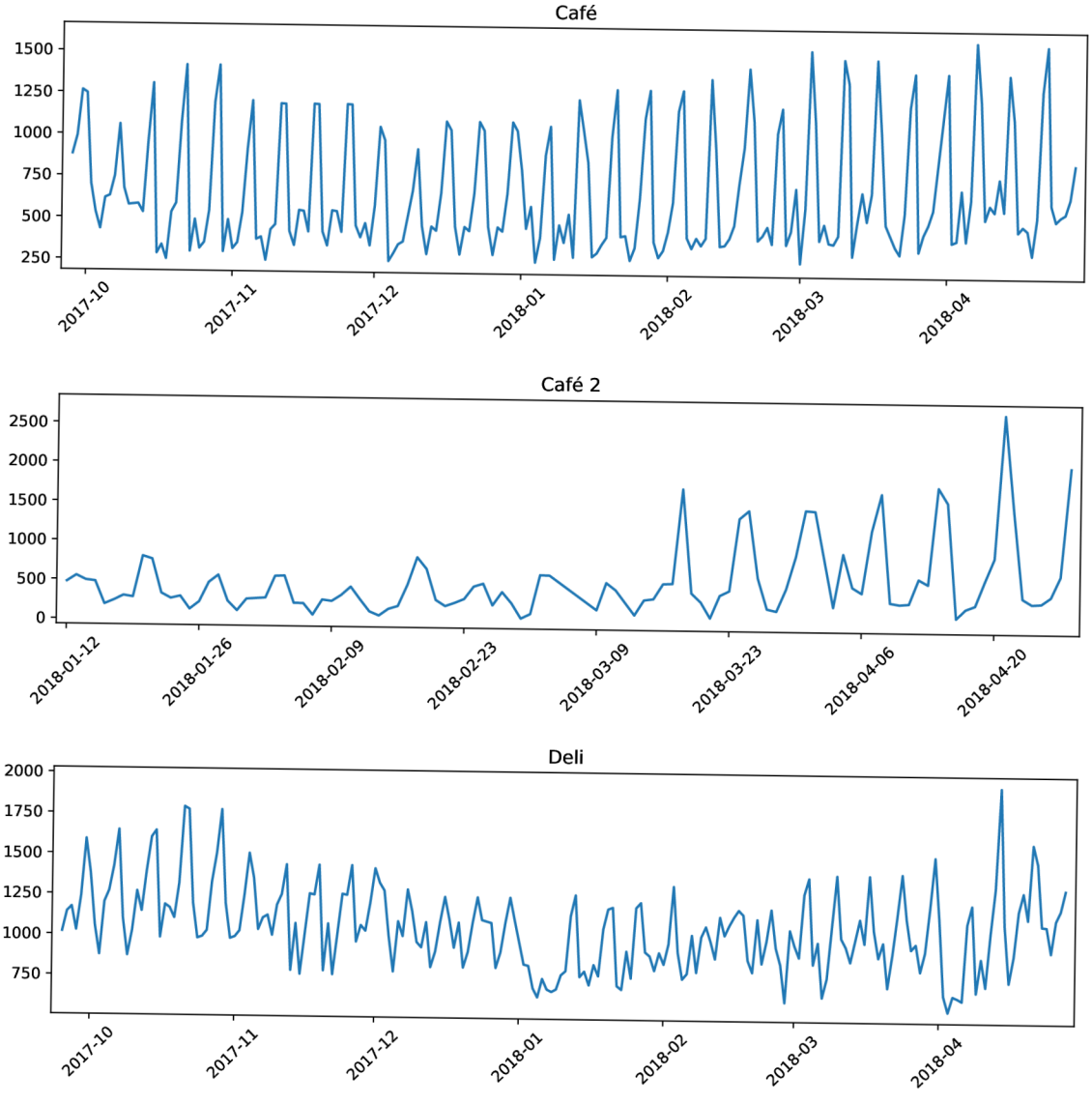
items	p-value	lift	confidence	rel. supp	supp
Chair Massage, 10 Minutes	5.28263e-16	41.1667	100.0%	2.4%	12
Pedicure, Callus Remover	0.140643	1.55672	10.0%	2.4%	12
Eyebrow, Lip	8.54847e-07	9.24324	36.0%	1.8%	9
Pedicure, Chair Massage, 10 Minutes	1.94456e-12	60.5248	9.7%	1.6%	8
Pedicure, 10 Minutes	0.27313	1.47024	6.8%	1.6%	8
Pedicure, Chair Massage	0.27313	1.47024	6.8%	1.6%	8
Eyebrow, Pedicure, Lip	9.53477e-05	11.3248	5.5%	1.0%	5
Pedicure, Chair Massage 10min	0.255722	1.57526	4.3%	1.0%	5

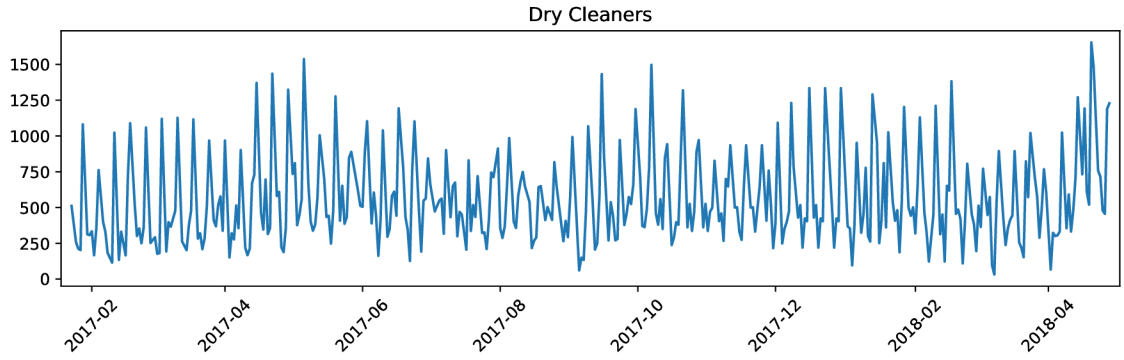
Table B.8: Cafe 2

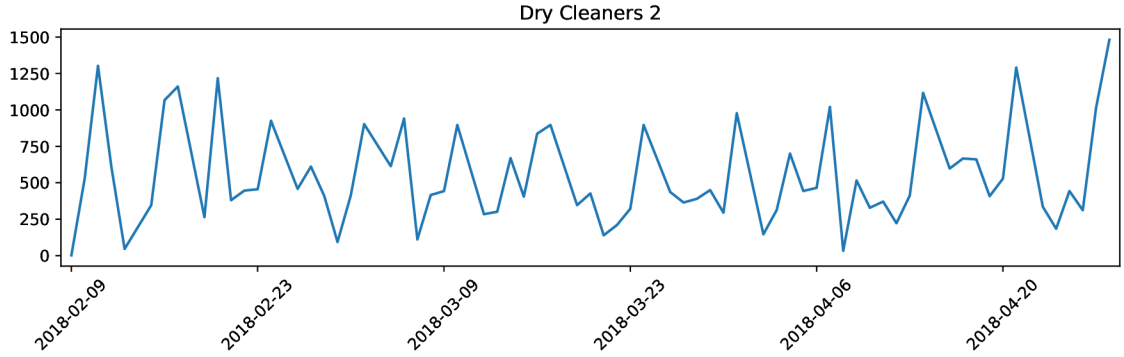
items	p-value	lift	confidence	rel. supp	supp
Rice - Brown, Sweet Soy Singer Chicken	6.1147e-08	2.40915	56.6%	13.6%	45
Rice - Brown, Sunny Egg	3.85977e-06	2.10435	52.7%	13.3%	44
Sunny Egg, 1/2 Avocado	6.8775e-09	2.66479	58.9%	13.0%	43
Rice - Brown, 1/2 Avocado	1.21691e-05	2.07134	50.3%	12.4%	41
Sunny Egg, Sweet Soy Singer Chicken	3.56561e-08	2.62687	56.3%	12.1%	40
Sunny Egg, Rice - White	1.36465e-07	2.57231	54.3%	11.5%	38
Sunny Egg, Chili Honey Pork	6.1473e-07	2.57931	51.1%	10.3%	34
1/2 Avocado, Rice - White	2.86185e-06	2.4312	50.0%	10.3%	34
1/2 Avocado, Sweet Soy Singer Chicken	4.6674e-06	2.35863	49.3%	10.3%	34
Rice - Brown, Chili Honey Pork	0.000161611	2.04085	44.0%	10.0%	33
Rice - White, Chili Honey Pork	2.30608e-07	2.80106	52.0%	9.7%	32

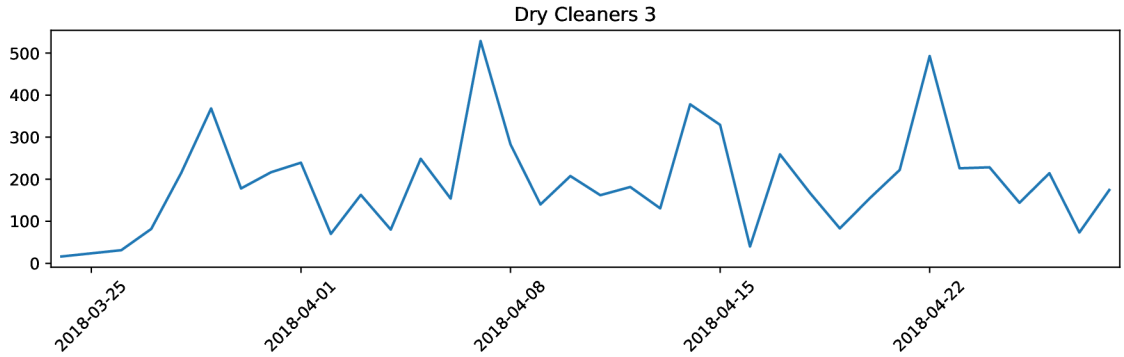
Appendix C

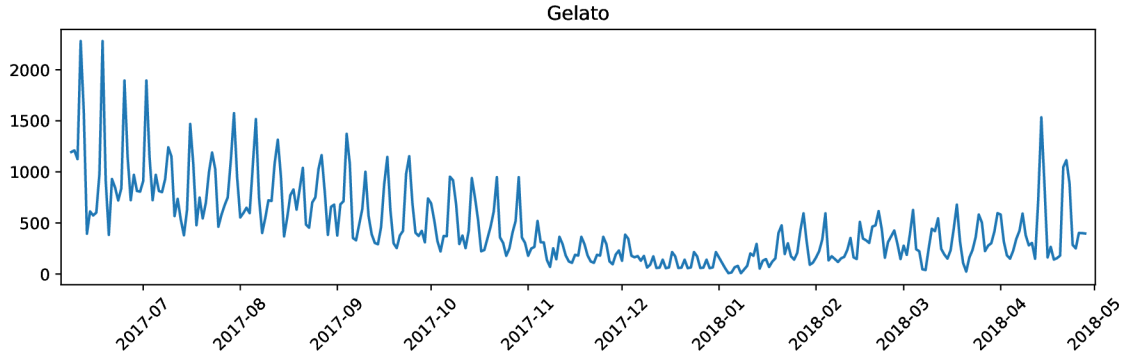
Preliminary

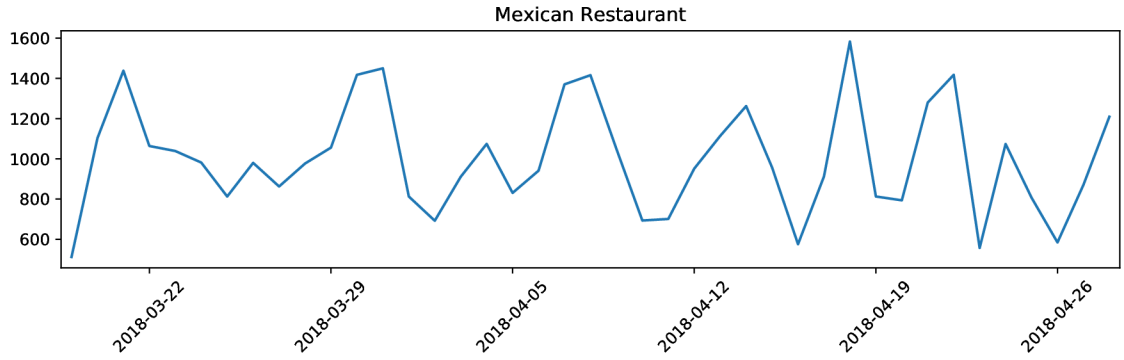


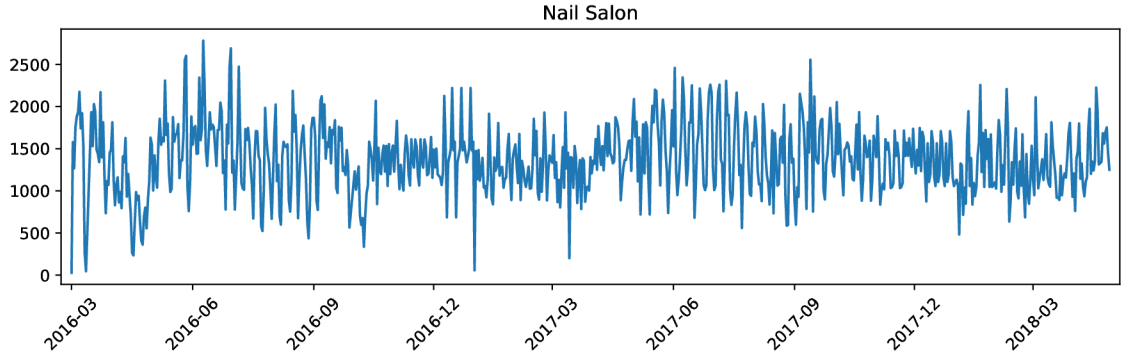

Dry Cleaners
1500
1250
1000
750
500
250
0
2017-02
2017-04
2017-06
2017-08
2017-10
2017-12
2018-02
2018-04

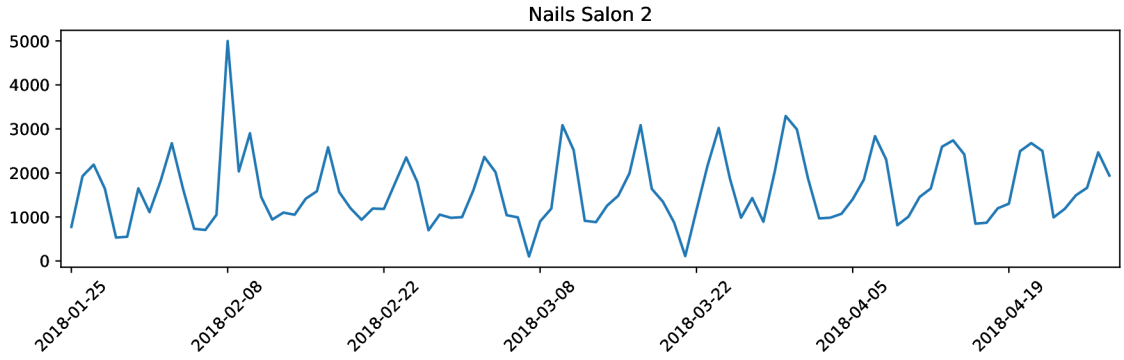

Dry Cleaners 2
1500
1250
1000
750
500
250
0
2018-02-09
2018-02-23
2018-03-09
2018-03-23
2018-04-06
2018-04-20

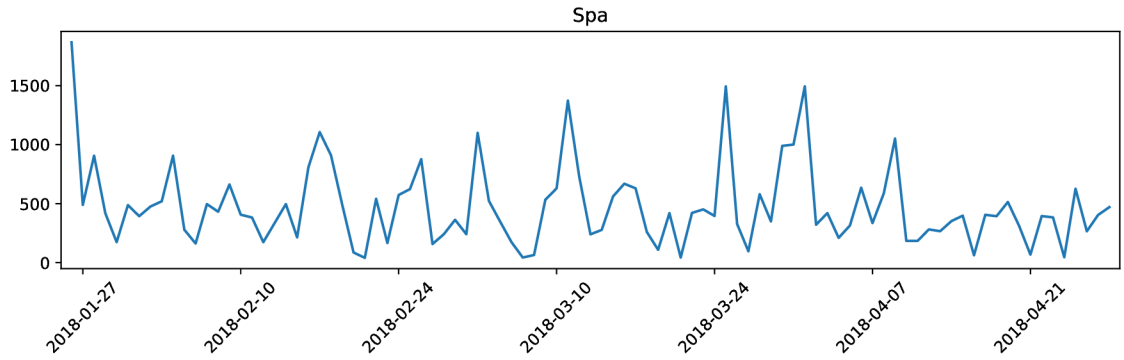

Dry Cleaners 3
500
400
300
200
100
0
2018-03-25
2018-04-01
2018-04-08
2018-04-15
2018-04-22

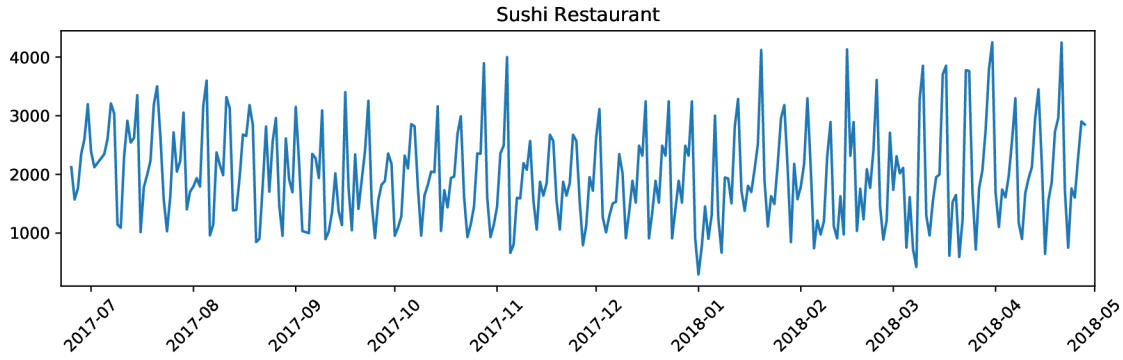

Gelato
2000
1500
1000
500
0
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05

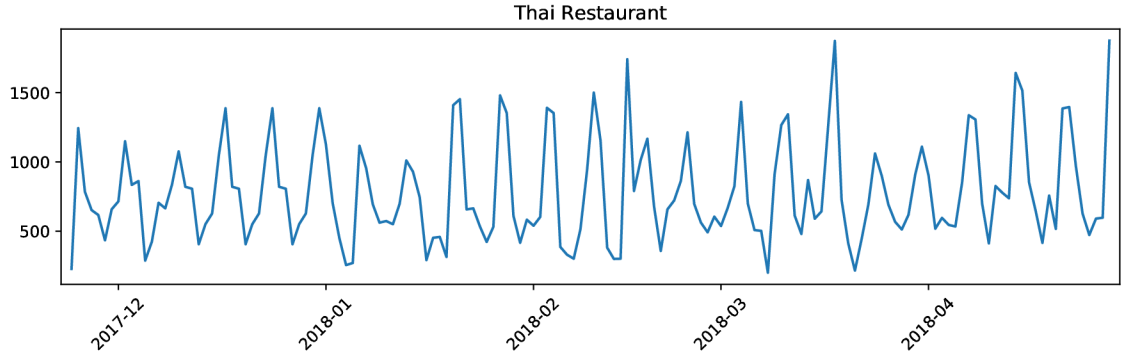

Mexican Restaurant
1600
1400
1200
1000
800
600
2018-03-22
2018-03-29
2018-04-05
2018-04-12
2018-04-19
2018-04-26

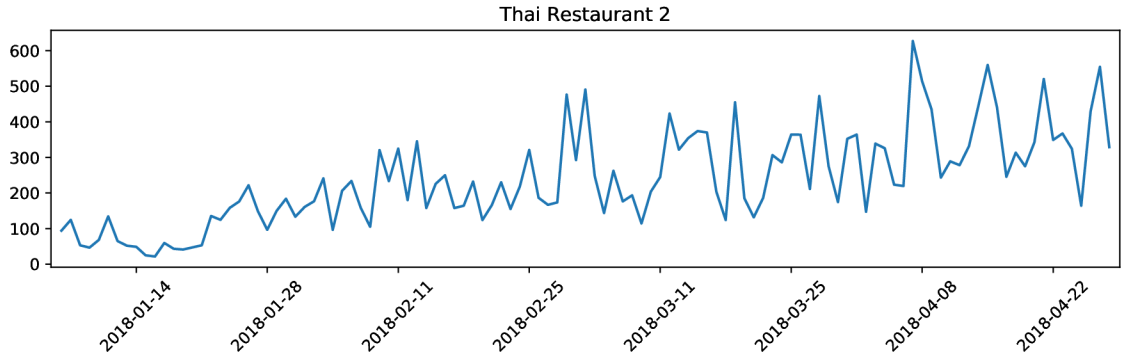

Nail Salon
2500
2000
1500
1000
500
0
2016-03
2016-06
2016-09
2016-12
2017-03
2017-06
2017-09
2017-12
2018-03

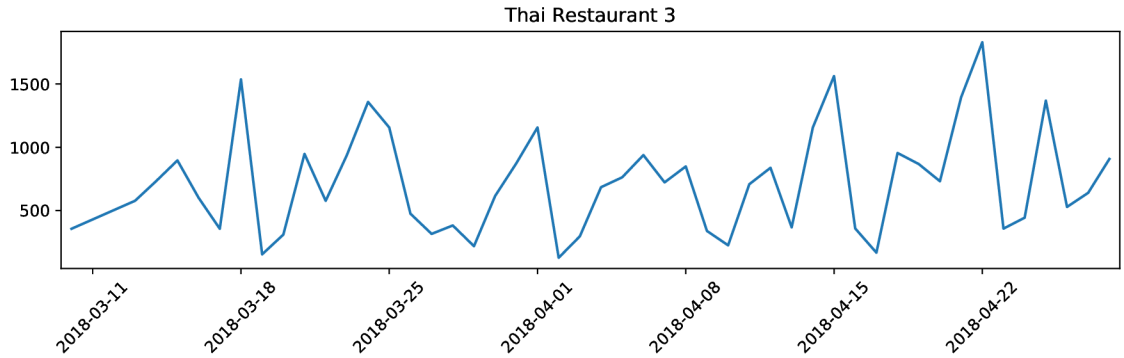

Nails Salon 2
5000
4000
3000
2000
1000
0
2018-01-25
2018-02-08
2018-02-22
2018-03-08
2018-03-22
2018-04-05
2018-04-19


Spa
1500
1000
500
0
2018-01-27
2018-02-10
2018-02-24
2018-03-10
2018-03-24
2018-04-07
2018-04-21


Sushi Restaurant
4000
3000
2000
1000
2017-07
2017-08
2017-09
2017-10
2017-11
2017-12
2018-01
2018-02
2018-03
2018-04
2018-05

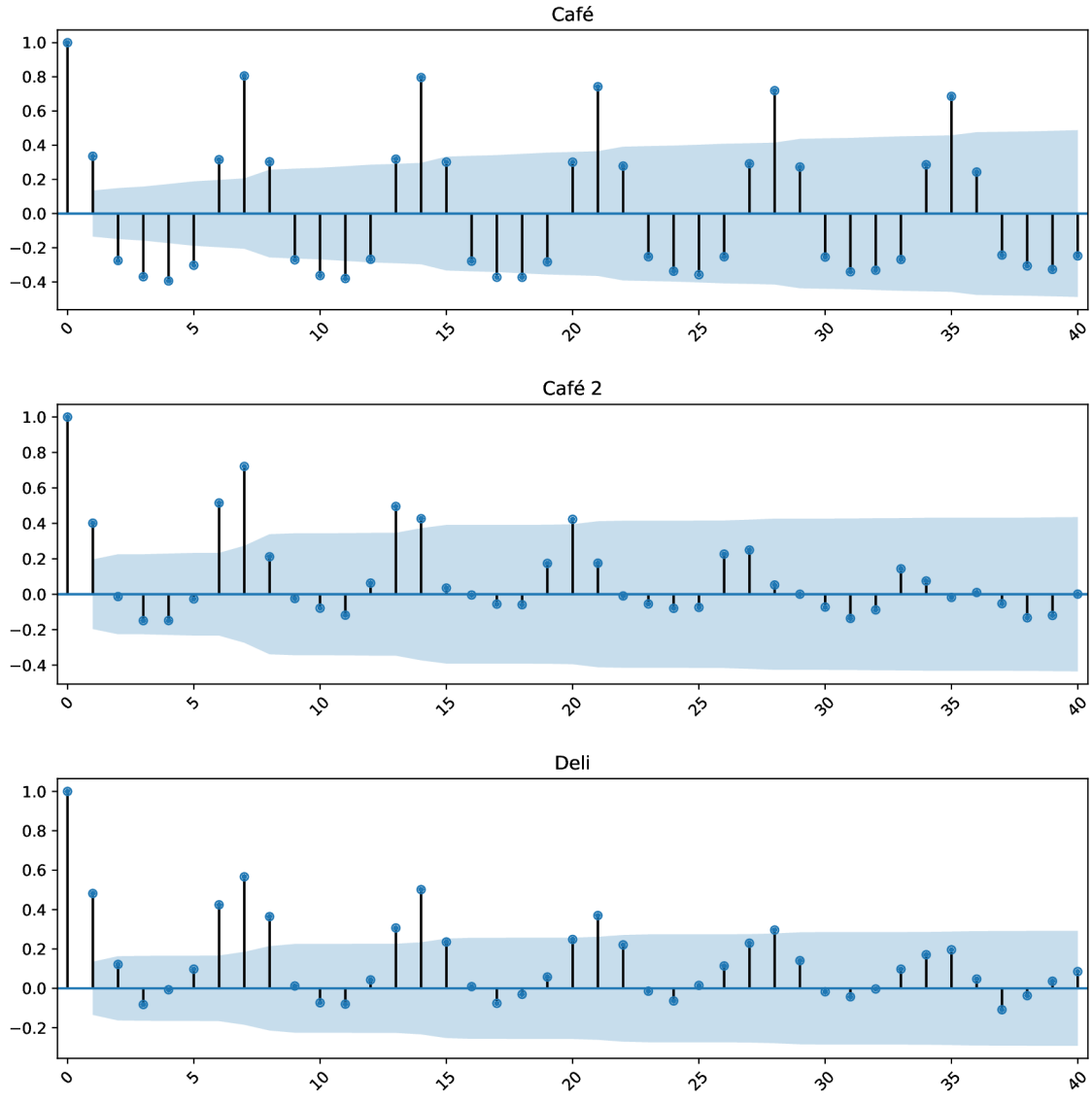

Thai Restaurant
1500
1000
500
2017-12
2018-01
2018-02
2018-03
2018-04

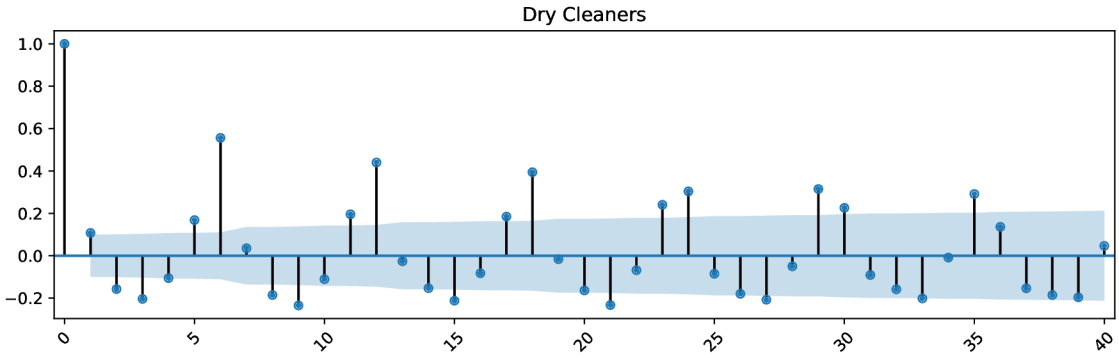

Thai Restaurant 2
600
500
400
300
200
100
0
2018-01-14
2018-01-28
2018-02-11
2018-02-25
2018-03-11
2018-03-25
2018-04-08
2018-04-22

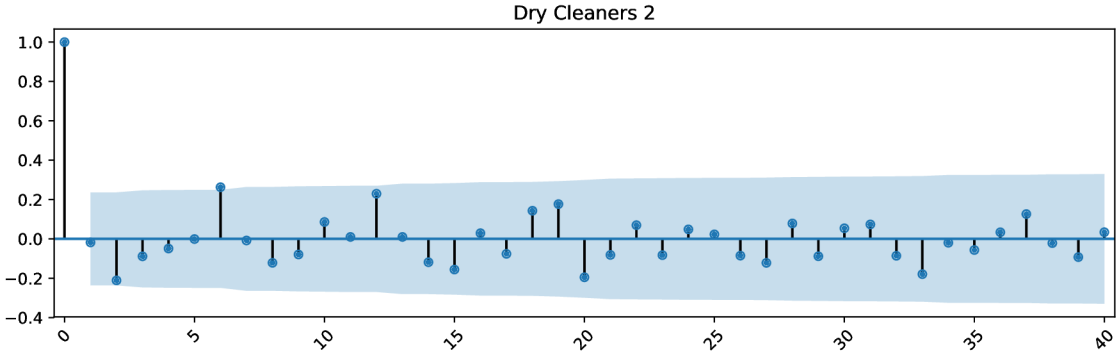

Thai Restaurant 3
1500
1000
500
2018-03-11
2018-03-18
2018-03-25
2018-04-01
2018-04-08
2018-04-15
2018-04-22

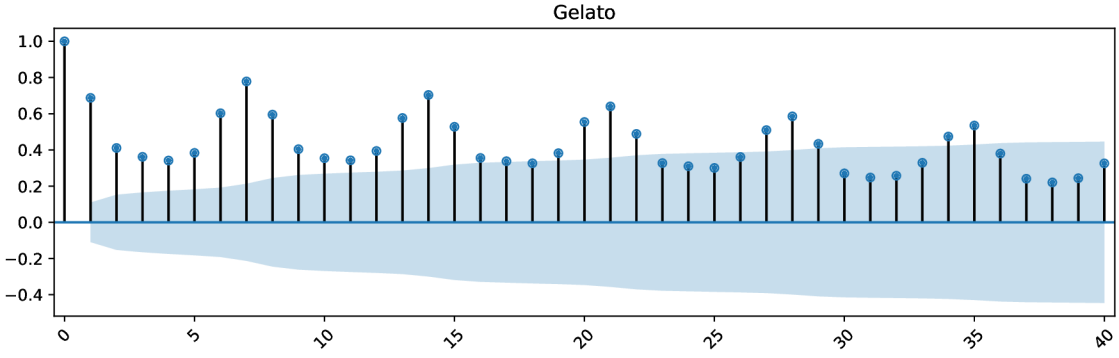
Appendix D

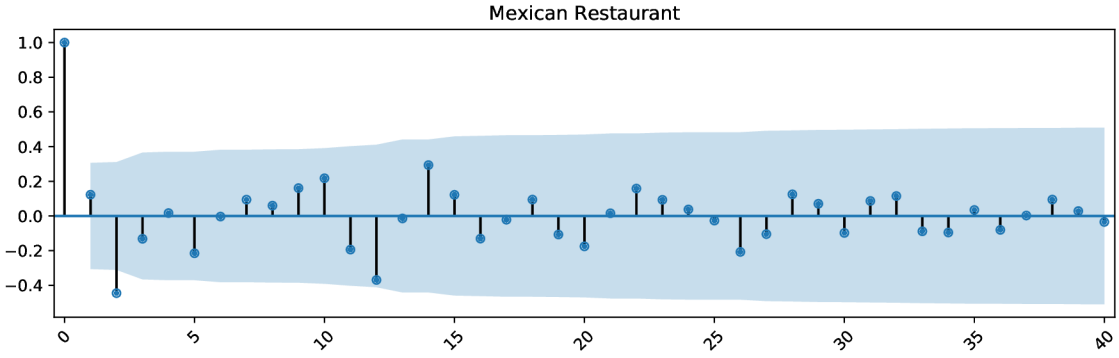
Autocorrelation plot

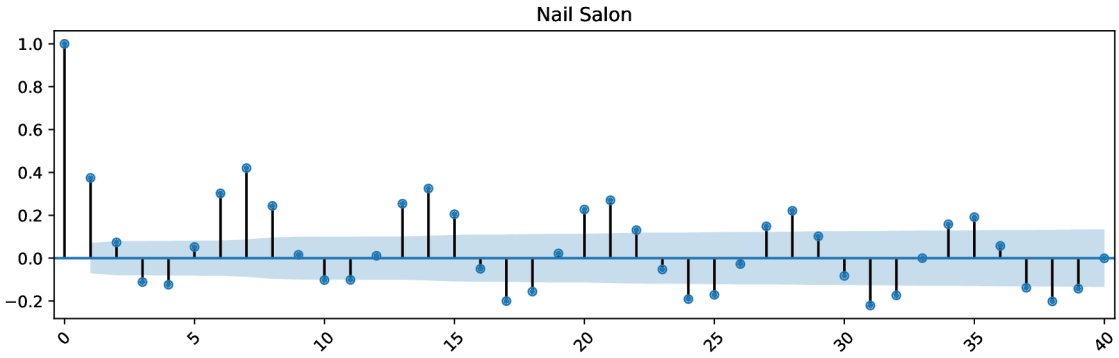


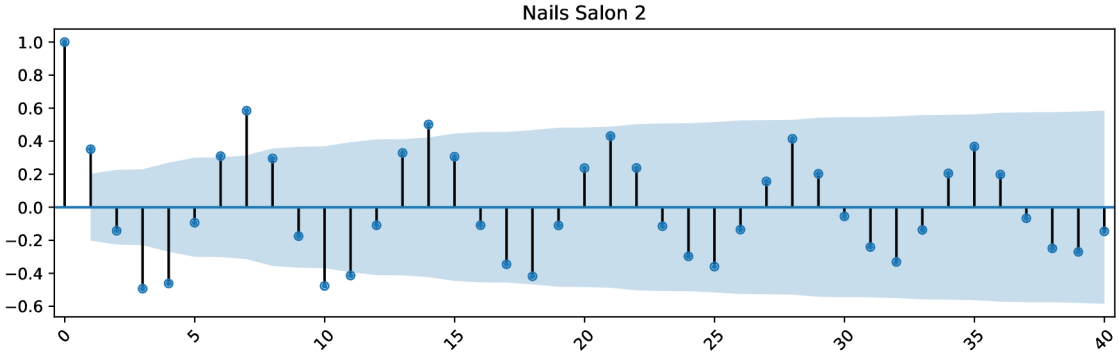

Dry Cleaners
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

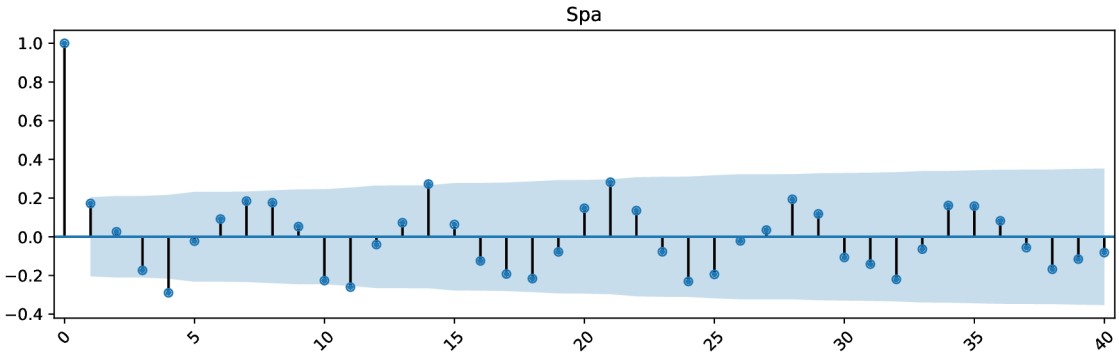

Dry Cleaners 2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

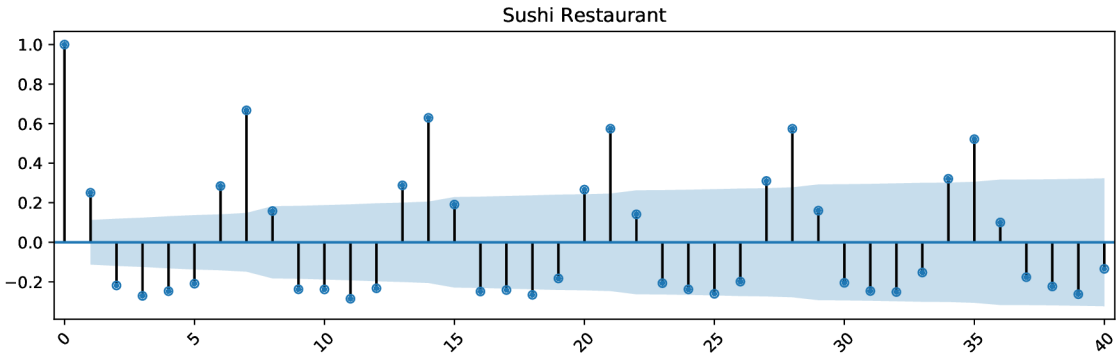

Gelato
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

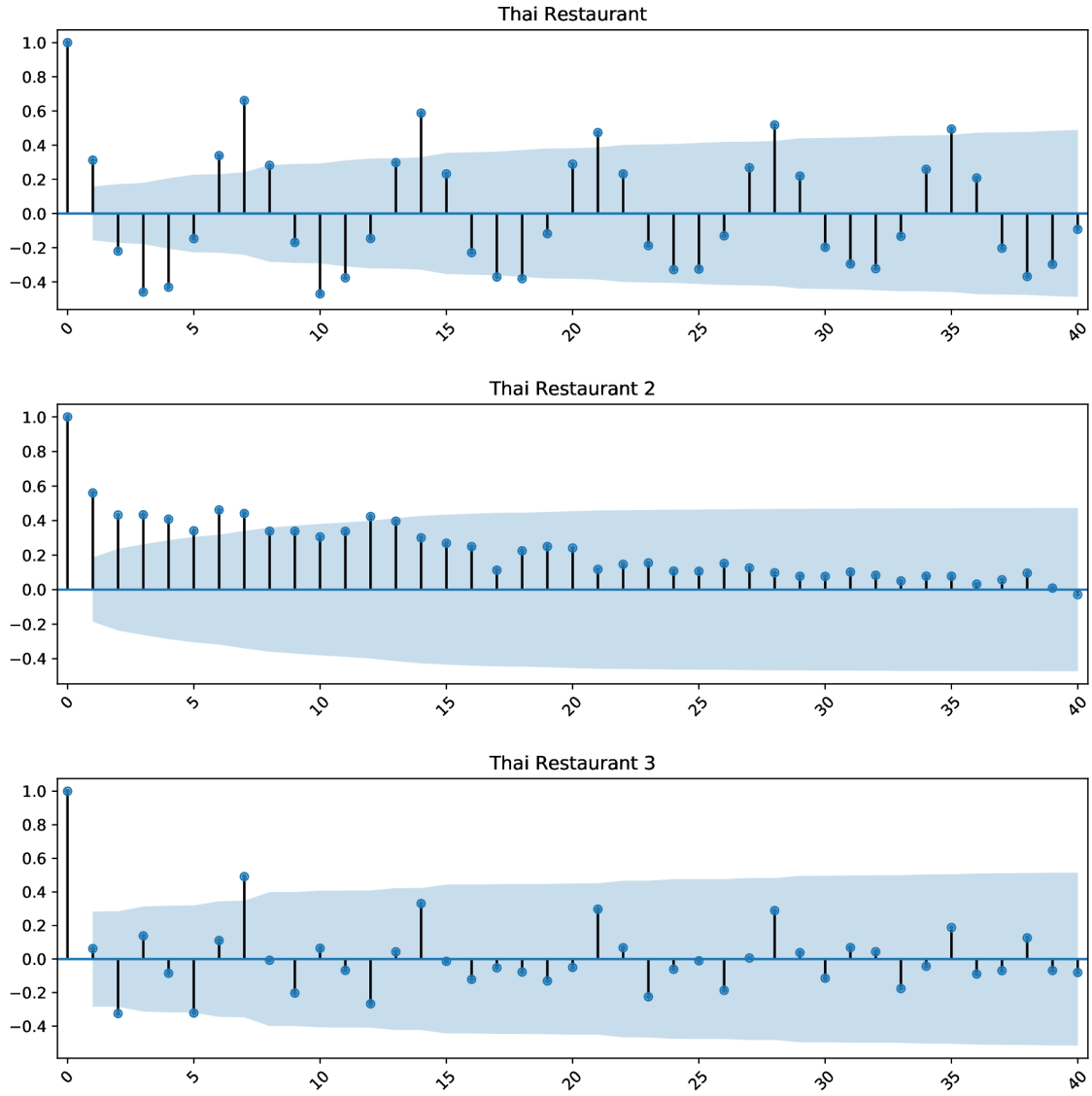

Mexican Restaurant
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40


Nail Salon
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40


Nails Salon 2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
0
5
10
15
20
25
30
35
40


Spa
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40


Sushi Restaurant
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

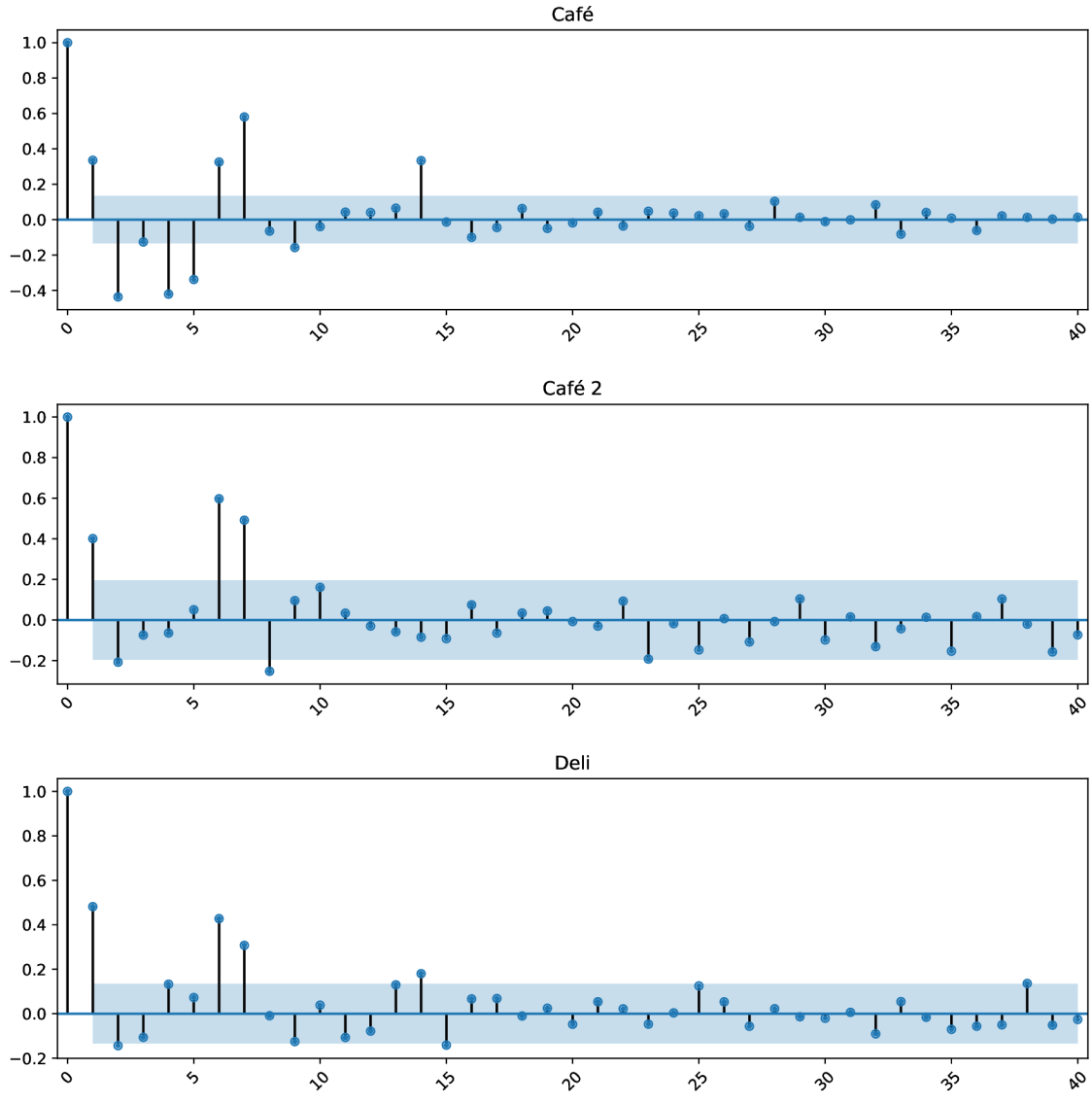

Thai Restaurant
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

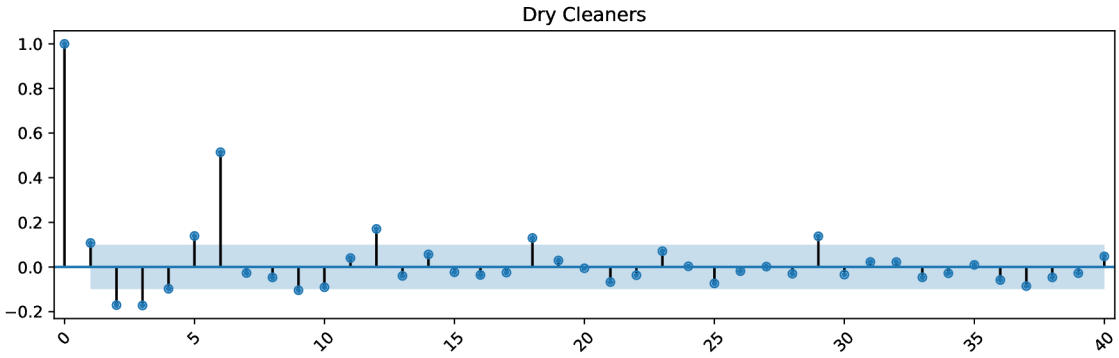
Thai Restaurant 2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

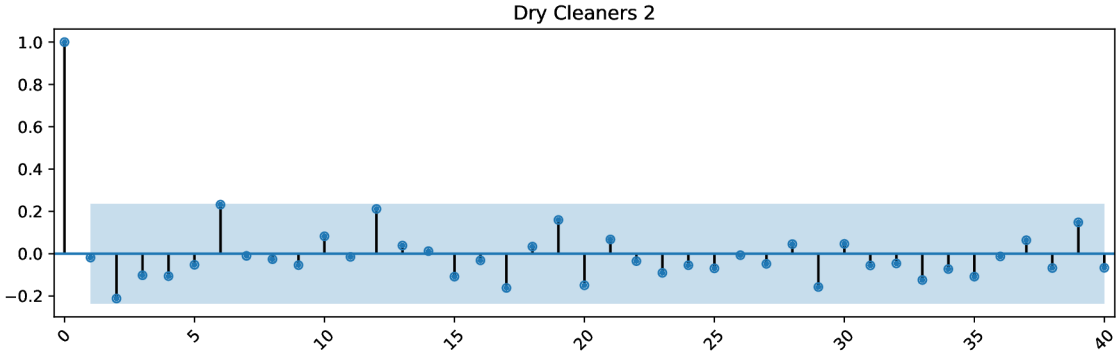
Thai Restaurant 3
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

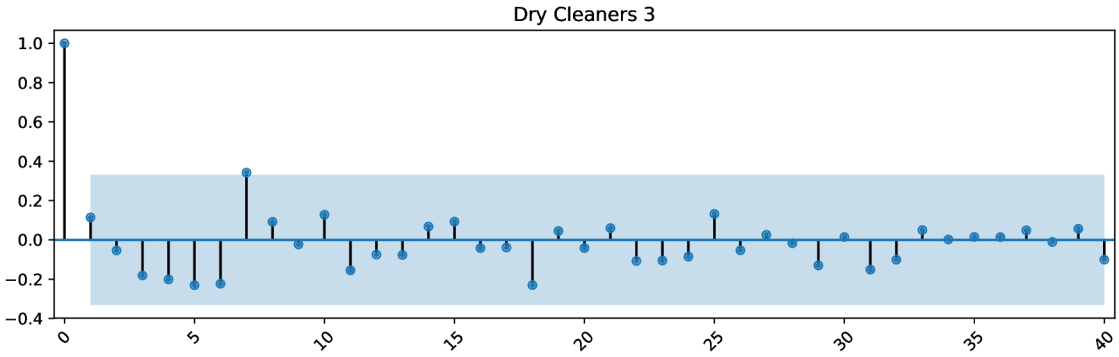
Appendix E

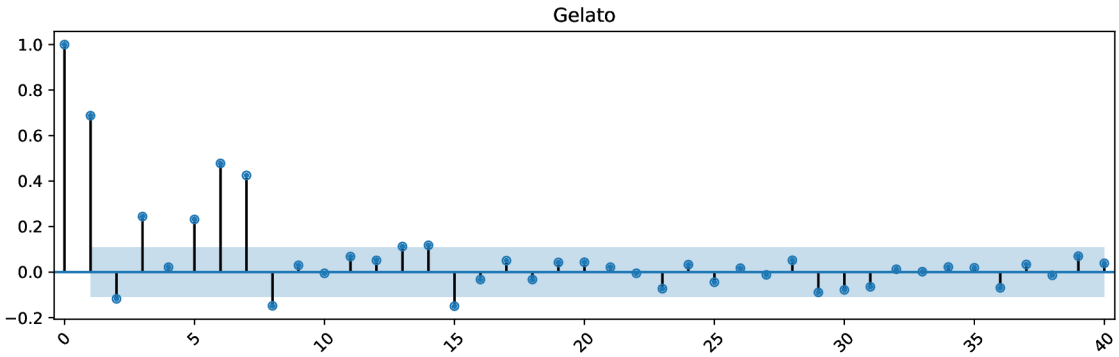
Partial autocorrelation plot

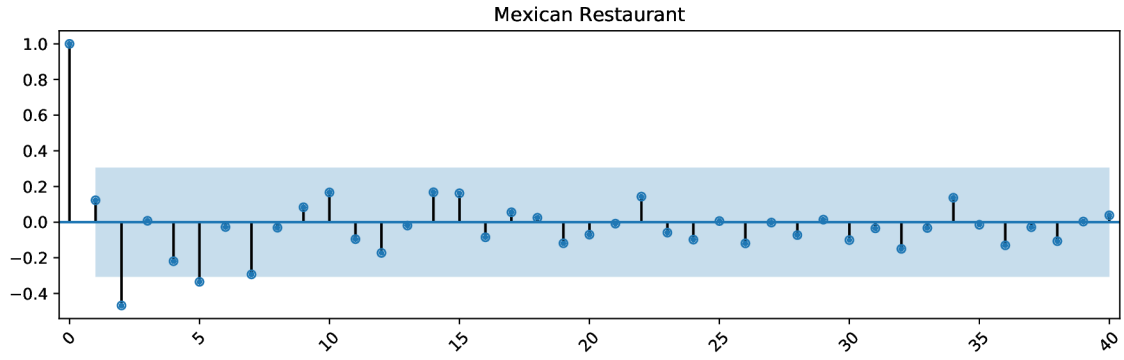


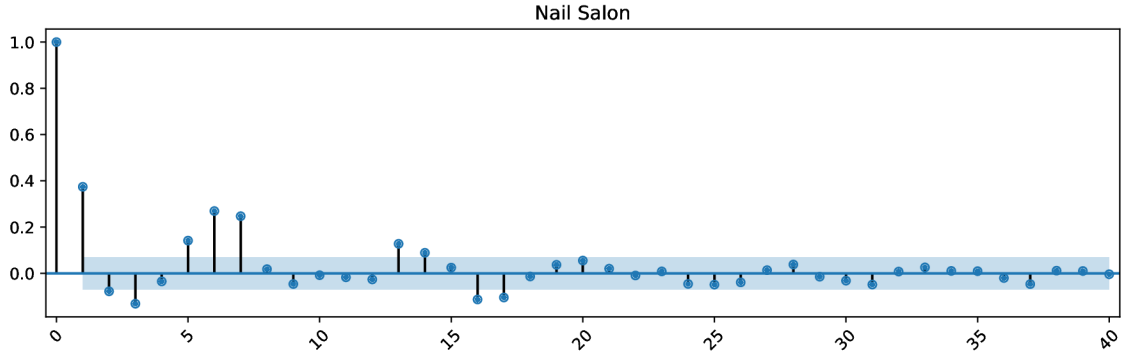

Dry Cleaners
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

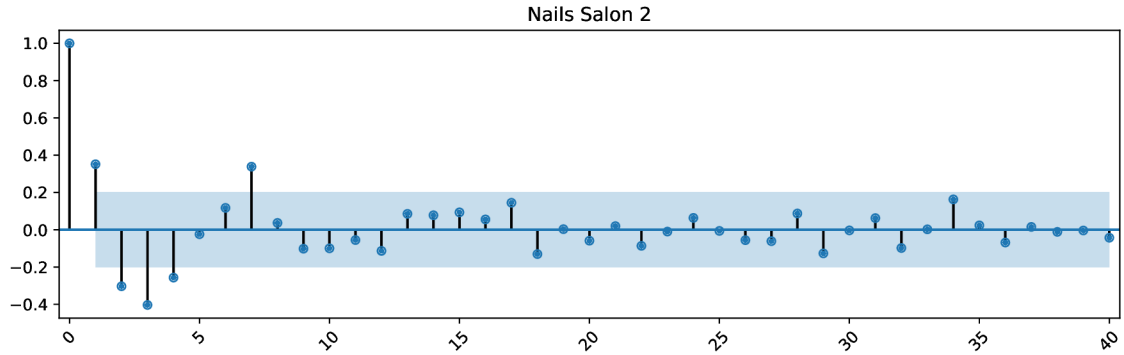

Dry Cleaners 2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

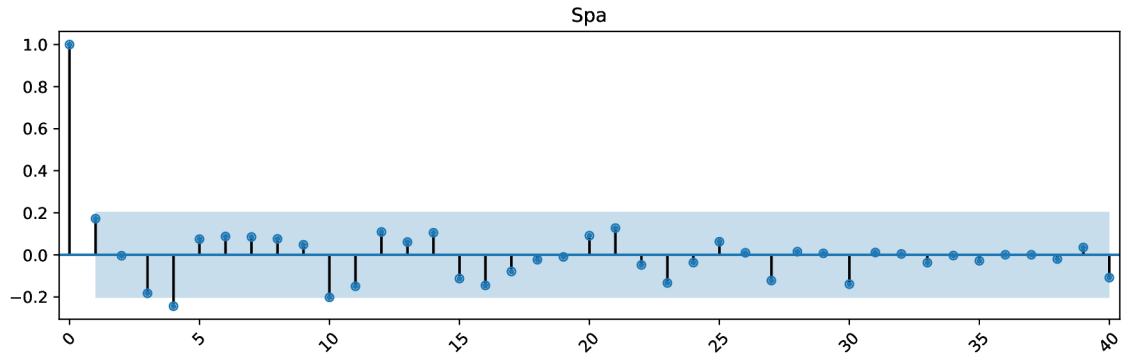

Dry Cleaners 3
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

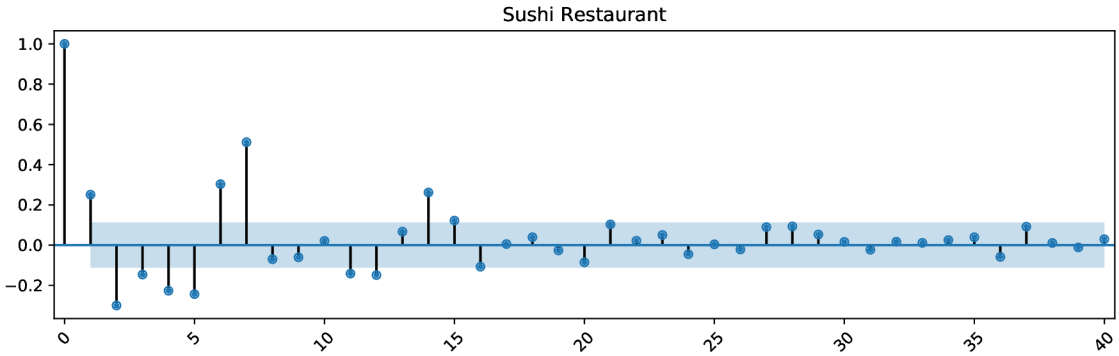

Gelato
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

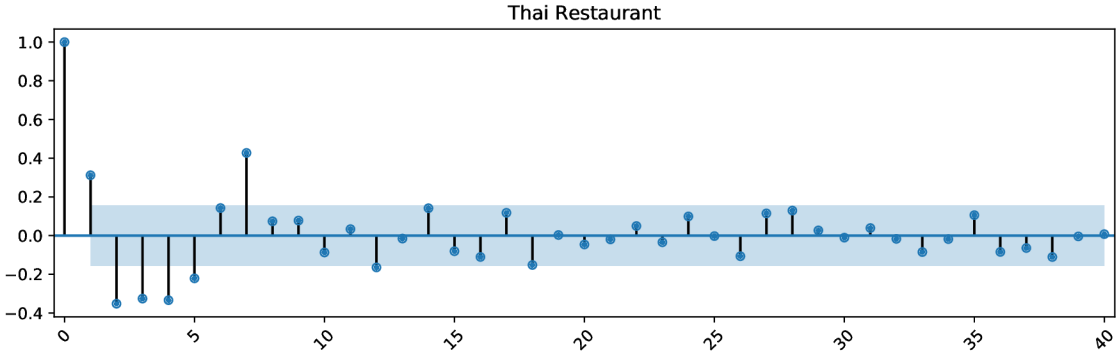

Mexican Restaurant
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

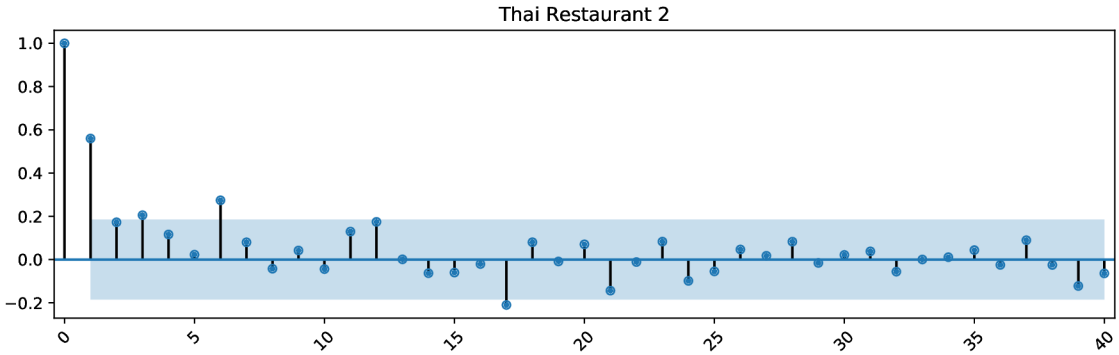

Nail Salon
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
35
40

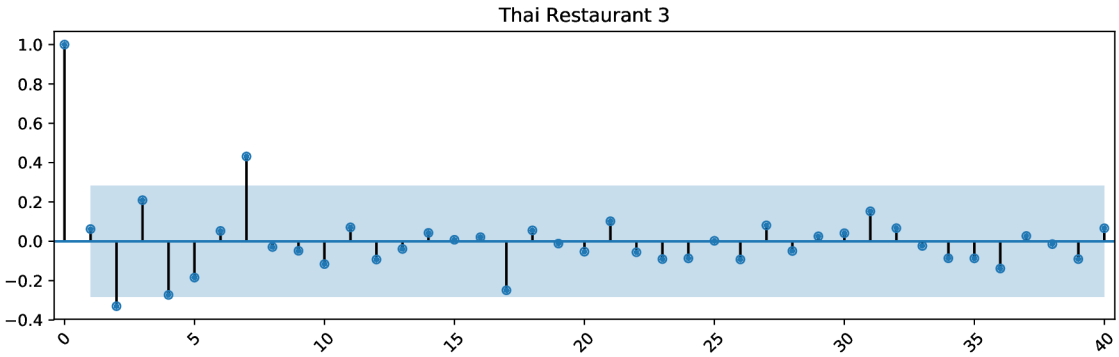

Nails Salon 2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40


Spa
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40


Sushi Restaurant
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

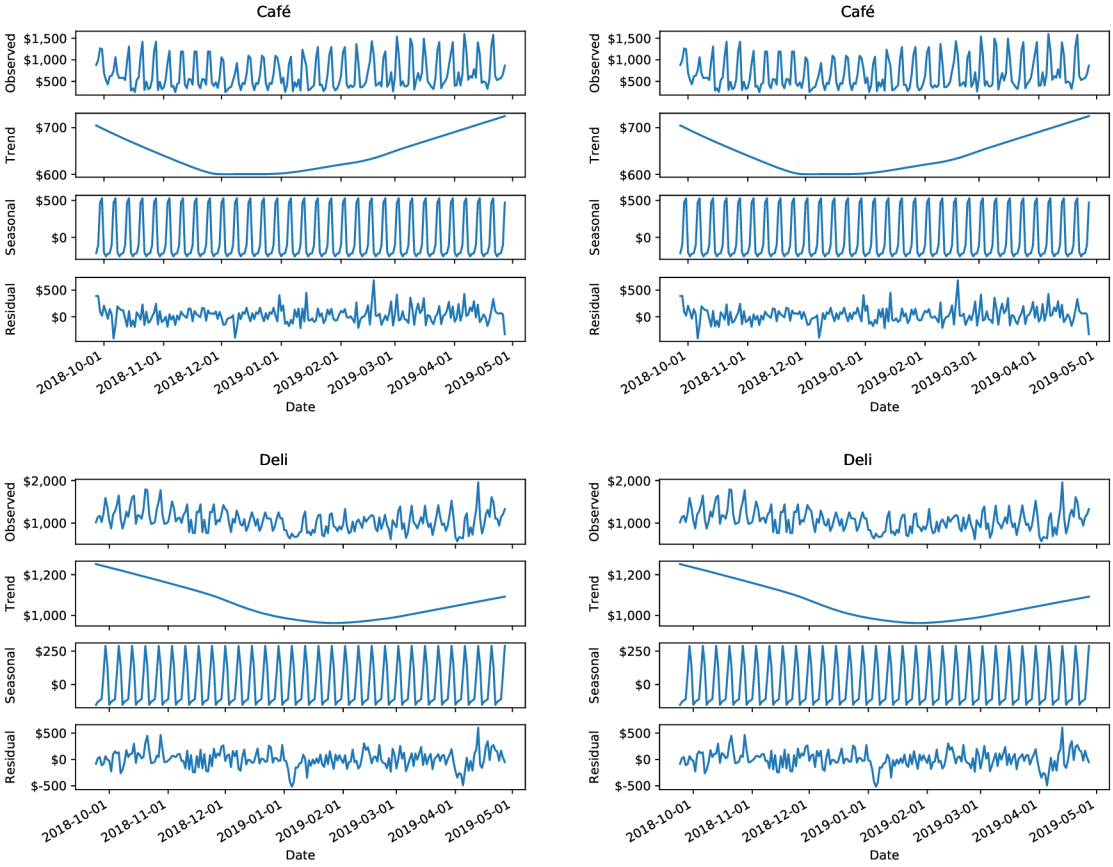

Thai Restaurant
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

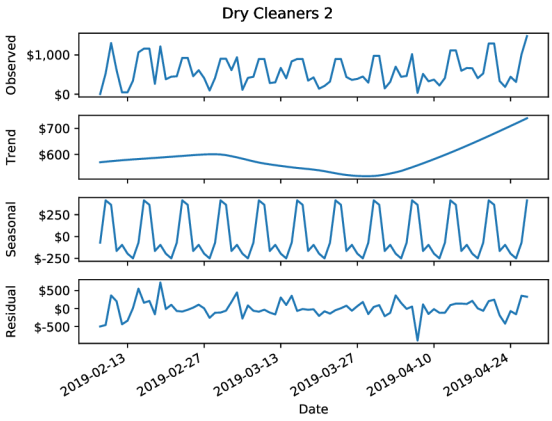

Thai Restaurant 2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
20
25
30
35
40

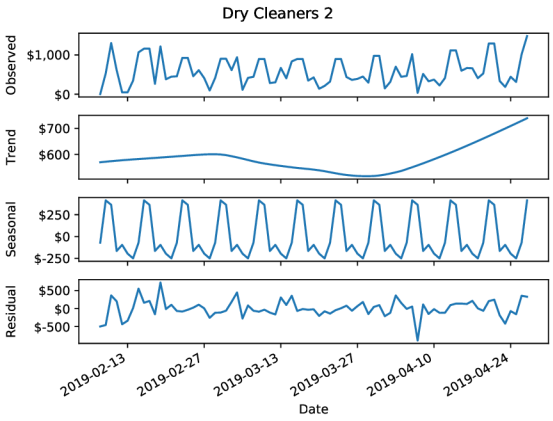

Thai Restaurant 3
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0
5
10
15
20
25
30
35
40

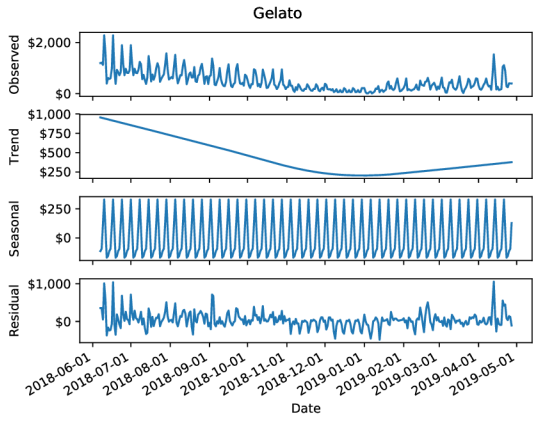
Appendix F

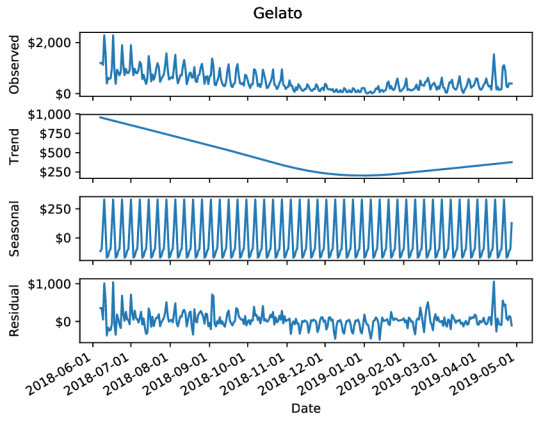
STL decomposition

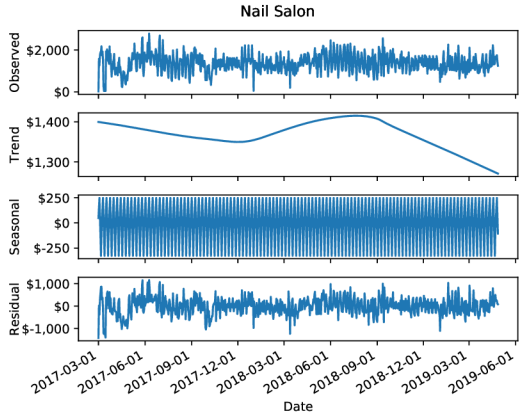


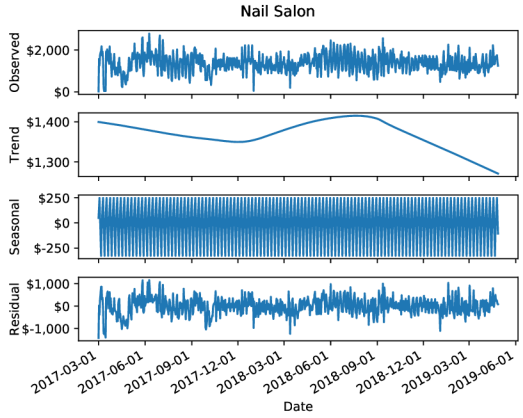

Dry Cleaners 2
Observed
$1,000
$0
Trend
$700
$600
Seasonal
$250
$0
$-250
Residual
$500
$0
$-500
2019-02-13
2019-02-27
2019-03-13
2019-03-27
2019-04-10
2019-04-24
Date

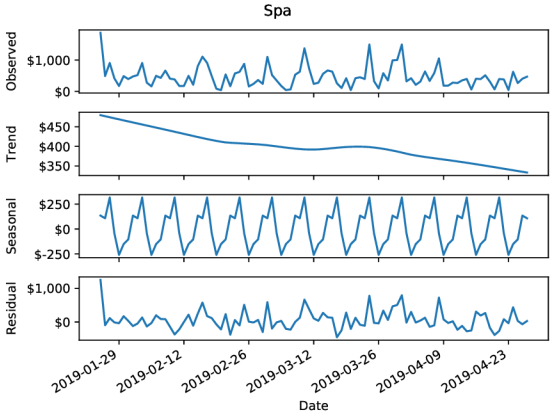

Dry Cleaners 2
Observed
$1,000
$0
Trend
$700
$600
Seasonal
$250
$0
$-250
Residual
$500
$0
$-500
2019-02-13
2019-02-27
2019-03-13
2019-03-27
2019-04-10
2019-04-24
Date

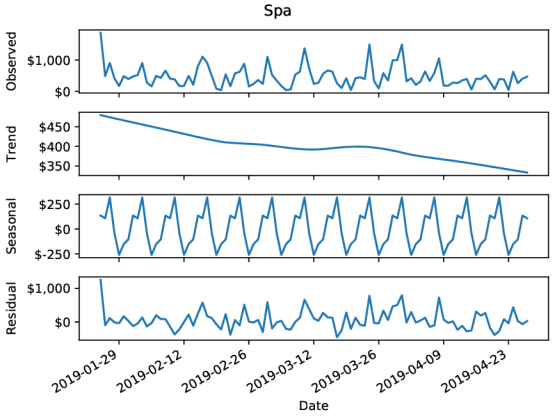

Gelato
Observed
$2,000
$0
Trend
$1,000
$750
$500
$250
Seasonal
$250
$0
Residual
$1,000
$0
2018-06-01
2018-07-01
2018-08-01
2018-09-01
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
Date

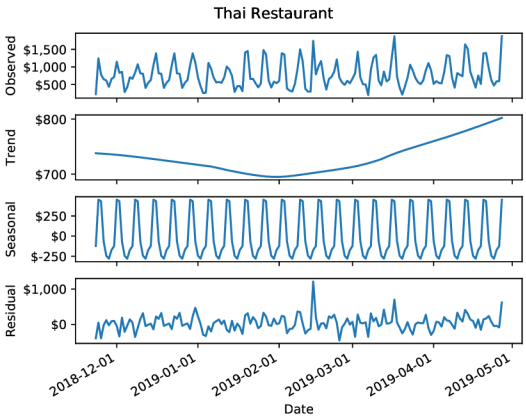

Gelato
Observed
$2,000
$0
Trend
$1,000
$750
$500
$250
Seasonal
$250
$0
Residual
$1,000
$0
2018-06-01
2018-07-01
2018-08-01
2018-09-01
2018-10-01
2018-11-01
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
Date

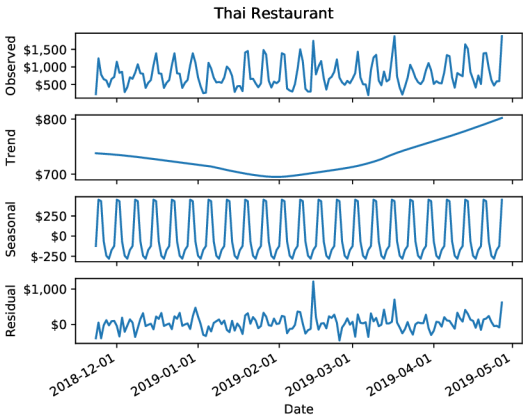

Nail Salon
Observed
$2,000
$0
Trend
$1,400
$1,300
Seasonal
$250
$0
$-250
Residual
$1,000
$0
$-1,000
2017-03-01
2017-06-01
2017-09-01
2017-12-01
2018-03-01
2018-06-01
2018-09-01
2018-12-01
2019-03-01
2019-06-01
Date


Nail Salon
Observed
$2,000
$0
Trend
$1,400
$1,300
Seasonal
$250
$0
$-250
Residual
$1,000
$0
$-1,000
2017-03-01
2017-06-01
2017-09-01
2017-12-01
2018-03-01
2018-06-01
2018-09-01
2018-12-01
2019-03-01
2019-06-01
Date


Spa
Observed
$1,000
$0
Trend
$450
$400
$350
Seasonal
$250
$0
$-250
Residual
$1,000
$0
2019-01-29
2019-02-12
2019-02-26
2019-03-12
2019-03-26
2019-04-09
2019-04-23
Date


Spa
Observed
$1,000
$0
Trend
$450
$400
$350
Seasonal
$250
$0
$-250
Residual
$1,000
$0
2019-01-29
2019-02-12
2019-02-26
2019-03-12
2019-03-26
2019-04-09
2019-04-23
Date


Thai Restaurant
Observed
$1,500
$1,000
$500
Trend
$800
$700
Seasonal
$250
$0
$-250
Residual
$1,000
$0
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
Date


Thai Restaurant
Observed
$1,500
$1,000
$500
Trend
$800
$700
Seasonal
$250
$0
$-250
Residual
$1,000
$0
2018-12-01
2019-01-01
2019-02-01
2019-03-01
2019-04-01
2019-05-01
Date

Appendix G

Program interface

Program, which was developed as a part of this thesis, uses Python 3.6 and associated Python libraries. It is standalone program is suitable for runnin in command line. Application is accesible by entry point `run_evaluation.py` with these parameters:

```
usage: run_evaluation.py [-h] [--graph] [--data-local]
                         [--run-function RUN_FUNCTION] [--args ARGS] [--api]

optional arguments:
  -h, --help            show this help message and exit
  --graph               Display graph in HTML format
  --data-local          Load data from local cached source
  --run-function RUN_FUNCTION
                        Execute given function
  --args ARGS           Argument for the executed function
  --api                 Load data through API
```

Parameter `-run-function` can be provided with desired model to be tested, options are:

1. `run decompostion`
2. `run_sarimax`
3. `run_ff`
4. `run_mars`

Program also contains configuration file `config.json`. File `config.local.json` has same format, but overrides parameters from `config.json` for environment-specific purposes.