FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

# Data Science and Visualisation Techniques applied on Bus Search Requests and its Correlating Booking Data

Subtitle

**Bachelor Thesis**

Submitted in partial fulfillment of the requirements for the degree of

**Bachelor of Science in Engineering**

to the University of Applied Sciences FH Campus Wien
Bachelor Degree Program: Computer Science and Digital Communications


**Author:**

first name surname

**Student identification number:**

Number

**Supervisor:**

Title first name surname


**Date:**

dd.mm.yyyy

Declaration of authorship:

I declare that this Bachelor Thesis has been written by myself. I have not used any other than the listed sources, nor have I received any unauthorized help.

I hereby certify that I have not submitted this Bachelor Thesis in any form (to a reviewer for assessment) either in Austria or abroad.

Furthermore, I assure that the (printed and electronic) copies I have submitted are identical.

Date: Signature:

# Abstract

(E.g. “This thesis investigates...”)

# Kurzfassung

(Z.B. “Diese Arbeit untersucht...”)

# List of Abbreviations

| | |
|---|---|
| ARP | Address Resolution Protocol |
| GPRS | General Packet Radio Service |
| GSM | Global System for Mobile communication |
| WLAN | Wireless Local Area Network |

# Key Terms

GSM

Mobilfunk

Zugriffsverfahren

# Contents

# 1 Introduction

# 2 Info

### 2.0.1 Research Question

How to apply Data Science and Visualization Techniques on Bus Search Requests and its Correlating Booking Data to create an Analytical Dashboard.

### 2.0.2 General

Basic Idea/Content:

- Explain available dataset, data structure, how data is gathered
- Explain what techniques are used to clean the base dataset
- Interdisciplinary - explain why certain KPIs or models are choosen and are applied onto the dataset
- Prediction Model - explain the technologies, methods etc. used to create a ML based prediction model (To improve Yield Management). Maybe two models, Supervised Learning and unsupervised learning
- Data Clustering and other KPIs + applied statistical models (e.g. Clustering, LR), Heatmaps etc.
- Visualisation techniques used to display results of the applied statistical models
- Maybe? Short chapter about technical setup of the Dashboard

In general i would appreciate some general feedback what else could/should be descriped within this thesis. I think the Prediction Model will be the main aspect of this thesis (How it is created + which techniques are used, how is the performance when comparing prediction to actual booking data etc.)

### 2.0.3 Lit Research

[1] - provides also a lot of useful references to other papers that can be used
[2] - ML
[3], [4], [5], [6] - Tensorflow, ML etc.
[7], [8] - interdisciplinary to provide context which KPI's etc are chosen etc.

# 3 Available Dataset

This chapter focuses on explaining and analysing the available data. The data is analyzed for Business Intelligence (BI) purposes as well as on metrics that can be used to create predictions. Whereas BI [**?**] focuses on historical data and aims to support managers to make decisions traditional methods like predictive analytic asses potential future scenarios using advanced statistical methods [**?**].

## 3.1 Data Origin

The available dataset is gathered from a website that provides a service to find and book buses for individual journeys. This service is currently available in Austria, Germany, Switzerland and Lichtenstein. The buses itself are offered in real time by various different bus companies. Offers can vary in price which is based bus calculations which may vary from operator to operator. The data is stored in a relational database. Since the service also provides the possibility to directly book a bus, booking and corresponding user meta data is available as well.

## 3.2 Data Structure

The service launched in March 2017 therefore booking data is available back to this date. Tracking the search requests was introduced in October 2020. The request table itself keeps track of 40 attributes but not all of them host valuable information that could be analysed therefore only the ones which can be analysed are listed and explained below:

- `task_id` - PK (incremented value)
- `createdAt` - At which time the search request was made.
- `accountId` - Not empty when the user is currently logged in
- `amountSearchResults` - How many buses can be offered
- `containsTripCompany` - If the user wants to stop at a certain company during the trip
- `distanceInMeters` - Distance between departure and destination place
- `durationInSeconds` - Duration of the trip
- `pax` - Amount of passengers
- `taskFrom_address` - Departure address
- `taskFrom_lat and lng` - Latitude and Longitude of the departure
- `taskTo_address` - Destination address

- `taskTo_lat and lng` - Latitude and Longitude of the destination
- `taskFrom_Time` - Desired departure time
- `taskTo_Time` - Estimated arrival time
- `cheapestPrice_amount` - The cheapest price for a bus
- `bus_id` - The operator with the cheapest bus
- `city` - From which city the request was made
- `country` - From which country the request was made

Whenever a booking is made the correlating data is stored within a booking table. As the booking table contains sensitive data which is not scope of the analysis, so only three attributes are used:

- `createdAt` - At which time the booking was made.
- `company_id` - FK used to identify who received the booking
- `task_id` - FK used to link the booking to an search request

## 3.3 Data Cleansing

During this process the available data is investigated for irregularities that cause distortion when applying statistical models.

Search requests are tracked whenever a user opens the service and searches for a certain connection. Given that behaviour it may occur that a user searches for the same connection within a short time window. This behaviour results in the need of de-duplication to avoid bias. To filter out duplicates the attributes `ipHash`, `createdAt`, `taskFrom_address` and `taskTo_address`. A search request is considered as non duplicate whenever the timespan between equal entries is larger than one hour. To pre-process the data the following logic is applied once //todoChange:

```
    query = '''
    DELETE t1
    FROM search_requests_clean t1
    INNER JOIN search_requests_clean t2
        ON t1.taskFrom_address = t2.taskFrom_address
        AND t1.taskTo_address = t2.taskTo_address
        AND t1.ipHash = t2.ipHash
        AND t1.createdAt > t2.createdAt
        AND t1.createdAt - t2.createdAt <= %s
'''

    timespan = 3600  # 3600 seconds  - 1 hour
    cursor = connection.cursor()
    cursor.execute(query, (timeframe,))
    connection.commit()
```

//todo more explanation The logic above compares all entries based on the attributes mentioned above removes equal entries that are within a timespan of 1 hour.

Regarding validation and norming the available data present in both tables no actions are required due to fact that attributes that do not meet their defined data types are not stored in first place.

## 3.4 Data Augmentation

Starting from March 2020 countries like Austria, Germany, Switzerland and Lichtenstein had to put travel restrictions into effect due to the ongoing Covid19 pandemic
citeHere. This travel restrictions impacted the gathered booking data as those restrictions forbid travelling.

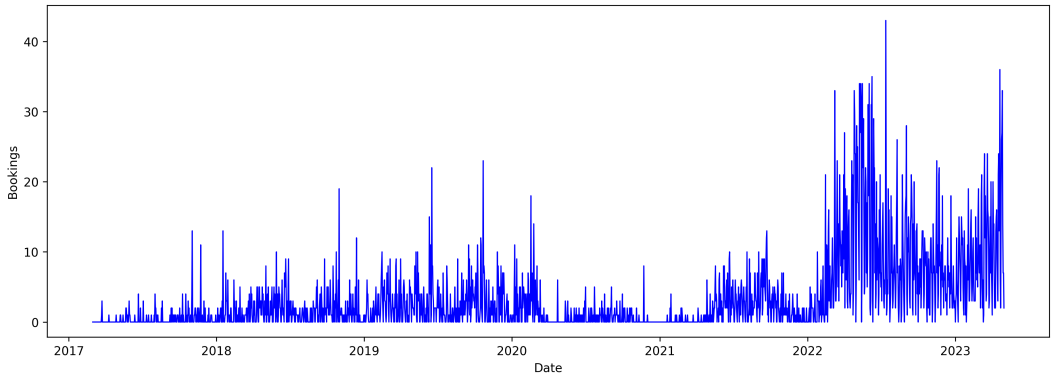


Figure 3.1: Drop in bookings - [source:author]

Figure 3.2 highlights the drop of bookings starting from March 2020 until June 2022. To achieve reliable results when utilizing this data for a time series forecasting ML model this time period needs augmentation. When analysing the chart 3.2 an continuous growth of bookings is visible until 2021. One way to augment the data
citehere is calculate the average growth during this time span. To substitute the distorted data the current data is replaced by the value of the previous year. This value is then multiplied by the average growth. Furthermore missing timestamps throughout the whole time series are added with a value of zero. The following logic is applied to the data frame:

```
df = db.get_booking_data()
average_growth = df['bookings'].pct_change().mean()
substitute_corona = pd.date_range(start='2020-03-01', end
   ='2022-05-01', freq='D')
df['date(createdAt)'] = pd.to_datetime(df['date(createdAt)'])
df = (df.set_index('date(createdAt)')
      .reindex(pd.date_range('2018-01-01', '2023-05-01', freq='D'))
      .rename_axis(['date(createdAt)'])
      .fillna(0)
      .reset_index())

df.set_index('date(createdAt)', inplace=True)

for date in substitute_corona:
```

```
year_ago = str(date - relativedelta(years=1)).split(" ")[0]
val = int(math.ceil(df.loc[year_ago]['bookings'] * (1+
    average_growth)))
df.loc[str(date).split(" ")[0]] = val
```

The average growth per anno is around 30%. After applying the logic the data set looks the following:

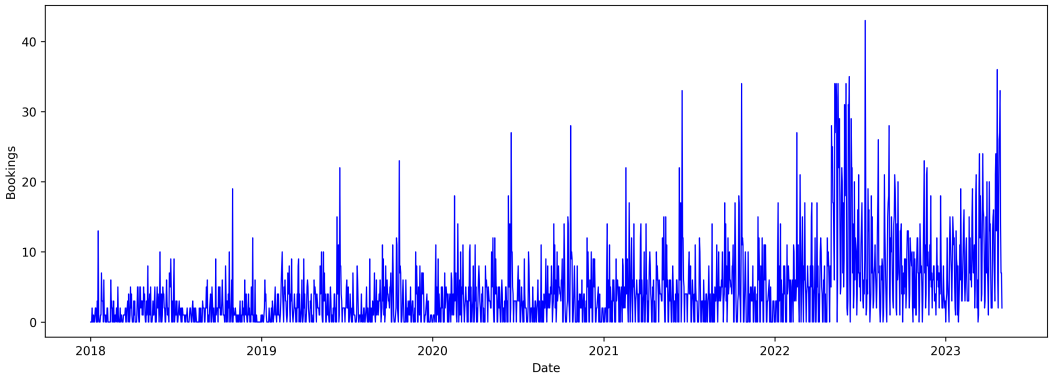


Figure 3.2: Augmented Data Set - [source:author]

The impact of this augmentation in terms of prediction accuracy is compared in chapter **??**.

## 3.5 Possible Analytical Strategies that can be applied

Explain what can be applied to the dataset, what procsses could be improved by analysing the data.

Improve Yield Managment (Prediction Model, ML)

# 4 Predicting Future Bookings

The knowledge of potential future bookings provide useful insights when it comes to yield management. Yield management in general describes controlling price and capacity control in a simultaneous ways [9]. Therefore those predictions can be used to support bus operators in their pricing strategy. This chapter focuses on creating two prediction models utilizing different techniques based on the data that is available. Both models are implemented using python and the following libraries.

- `matplotlib`[1] - used for plotting
- `pandas`[2] - used for data manipulation
- `tesnorflow`[3] - provides ML models
- `keras`[4] - Neural Network library

As there are various models available a literature review was conducted to figure out which models fit the purpose of time series forecasting. It turns out that the most promising NN that can be utilized for time series prediction are either Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) especially Long Short-Term Memory (LSTM)[10][11][12][13].

## 4.1 The Models

Both models CNN and RNN/LSTM can be used for time series forecasting. To create accurate prediction models a basic knowledge about models functionality is required. Therefore this section explains the components of each NN as well as the approaches those models follow.

### 4.1.1 LSTM

LSTM is an RNN and was invented by [14] in 1997. Until today this NN is widley used for time series forecasting and provides reliable results for short as well as long term predictions [15]. LSTM have so called memory cells which are responsible to store the state of data. Whenever information arrives at a memory cell its outcome is defined by refreshing the cell state with the newly arrived information. LSTM utilizes gates to control a cells state by either including or excluding information [16]. The gates are called:

- input gate - data selection and storage for upcoming state
- forget gate - data selection and storage which will not be used for the upcoming state
- output gate - sets information within the state that is send to the output

[1] https://matplotlib.org/
[2] https://pandas.pydata.org/
[3] https://www.tensorflow.org/
[4] https://keras.io/

Those gates are created by combining sigmoid functions. The results of this gates are values ranging from zero to one. A result of zero indicates the cell to not pass any infomration whereas values close to one indicates the cell to pass all information. The LSTM Module or Repeating module consists of four NN layers which interact together as shown in Figure 4.1:

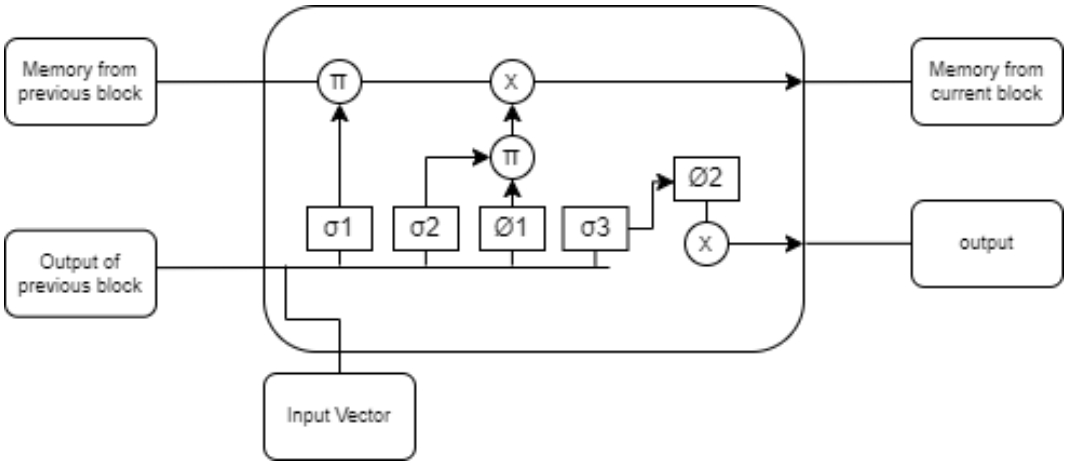


Figure 4.1: Repeating LSTM Module - [source:[17]]

In total the repeating model has 3 gate activation functions which are named $\sigma_1$, $\sigma_2$, $\sigma_3$ in figure 4.1. Furthermore $\sigma_1$ and $\sigma_2$ act as output activation functions too. The cell state is illustrated using a blue line which starts at St-1 which indicates the previous memory block to St representing the current memory block. The amount of information that is passed is regulated by layer $\sigma_1$ using the following function:

$$cf_t = \sigma_1(W_c f * [O_t - 1, x_t] + b_c f)[17] \tag{4.1}$$

Furthermore two network layers are used to store new information to the cell state. Therefore sigmoid layer $\sigma_2$ chooses the values which are updated by utilizing the following formula:

$$l_t = \sigma_2(W_1 * [O_t - 1, x_t] + b_l)[17] \tag{4.2}$$

Layer $\phi_1$ or `tanh` is created by using new candidate values. This layer outputs a vector by utilzing the following formular:

$$\widetilde{S}_t = tanh(W_s * [O_t - 1, X_t] + b_s)[17] \tag{4.3}$$

The last step includes combination of both states 4.2 and 4.3 which is added to the state. Also the state is reconditioned by applying: [17]

$$S_t = cf_t * S_t 1 + I_t * \widetilde{S}_t - 1[17] \tag{4.4}$$

The reason why a LSTM model is used for this purpose is that a standalone RNN is challenging to train due its characteristics. As Back propagation is used for RNN's problems like vanishing-gradient occur. The gradient in general can be understand as a computed value through all time setps which in the end used to update parameters of the RNN. The vanasihing-gratdient over time results in information decay.[18]

### 4.1.2 CNN

## 4.2 Implementation

## 4.3 Reliability Comparison - LSTM, CNN

## 4.4 Model accuracy

Having a look at the model performance accuracy (comparing predictions of the model with already available data) , explain potential twerks that have been applied to the model itself to achieve a higher level of accuracy.

# 5 Analytical Dashboard

## 5.1 Technical Setup

Explain the basic setup and used technologies for used for the analytical web based dashbaord

## 5.2 Applied Statistical Models

explain which attributes also provide additional information that can be gathered from the dataset, which models were applied (algorithms)

## 5.3 Visualisation techniques

which plots etc (and why) are used to display the gathered information

# Bibliography


[1] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham, and L. A. Gabralla, "A review on machine learning styles in computer vision—techniques and future directions," *IEEE Access*, vol. 10, pp. 107 293–107 329, 2022. 2

[2] K. Pahwa and N. Agarwal, "Stock market analysis using supervised machine learning," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 197–200. 2

[3] A. GÃ©ron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*, third edition. ed., 2022. 2

[4] A. Géron, "Praxiseinstieg machine learning mit scikit-learn und tensorflow : Konzepte, tools und techniken für intelligente systeme," 2018. 2

[5] P. R. Gulhane and T. S. Pradeep Kumar, "Tensorflow based website click through rate (ctr) prediction using heat maps," in *2018 International Conference on Recent Trends in Advance Computing (ICRTAC)*, 2018, pp. 97–102. 2

[6] A. Dehghan-Banadaki, T. Taufik, and A. Feliachi, "Big data analytics in a day-ahead electricity price forecasting using tensorflow in restructured power systems," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 1065–1069. 2

[7] A. Wannes and S. A. Ghannouchi, "Kpi-based approach for business process improvement," *Procedia Computer Science*, vol. 164, pp. 265–270, 2019, cENTERIS 2019 - International Conference on ENTERprise Information Systems / ProjMAN 2019 - International Conference on Project MANagement / HCist 2019 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919322215 2

[8] P. R. Martins de Andrade and S. Sadaoui, "Improving business decision making based on kpi management system," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 1280–1285. 2

[9] K. Donaghy, U. McMahon, and D. McDowell, "Yield management: an overview," *International Journal of Hospitality Management*, vol. 14, no. 2, pp. 139–150, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0278431995000133 7

[10] L. Wang, J. Chen, W. Wang, R. Song, Z. Zhang, and G. Yang, "Review of time series traffic forecasting methods," in *2022 4th International Conference on Control and Robotics (ICCR)*, 2022, pp. 1–5. 7

[11] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep learning-based stock price prediction using lstm and bi-directional lstm model," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2020, pp. 87–92. 7

[12] C. Hou, J. Wu, B. Cao, and J. Fan, "A deep-learning prediction model for imbalanced time series data forecasting," *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 266–278, 2021. 7

[13] J. Deng and P. Jirutitijaroen, "Short-term load forecasting using time series analysis: A case study for singapore," in *2010 IEEE Conference on Cybernetics and Intelligent Systems*, 2010, pp. 231–236. 7

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, p. 1735–1780, nov 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735 7

[15] K. Moharm, M. Eltahan, and E. Elsaadany, "Wind speed forecast using lstm and bi-lstm algorithms over gabal el-zayt wind farm," in *2020 International Conference on Smart Grids and Energy Systems (SGES)*, 2020, pp. 922–927. 7

[16] M. A. Istiake Sunny, M. M. S. Maswood, and A. G. Alharbi, "Deep learning-based stock price prediction using lstm and bi-directional lstm model," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, 2020, pp. 87–92. 7

[17] J. Kumar, R. Goomer, and A. Singh, "Long short term memory recurrent neural network (lstm-rnn) based workload forecasting model for cloud datacenters," *Procedia Computer Science*, vol. 125, pp. 676–682, 01 2018. 8, 13

[18] S. Bodapati, H. Bandarupally, and M. Trupthi, "Covid-19 time series forecasting of daily cases, deaths caused and recovered cases using long short term memory networks," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)*, 2020, pp. 525–530. 8

# List of Figures

# List of Tables

# Appendix

(Hier können Schaltpläne, Programme usw. eingefügt werden.)