Application of Machine Learning Algorithms to Predict Bus Arrival Delays

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Jicheng Li

Spring, 2022

Technical Project Team Members

Zhijun Cao

Yancheng Zhou

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

Application of Machine Learning Algorithms to Predict Bus Arrival Delays

CS4991 Capstone Report, 2023

Jicheng Li
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
jl9dbb@virginia.edu

ABSTRACT

The Charlottesville Area Transit (CAT) runs twelve bus lines that carry UVA students and local citizens every day. Although the SPOT app provides real-time update for the location of buses, the app is often inaccurate and students still need to predict the arrival time based on the location of the bus. We, a team of three students in CS 4774, use machine learning algorithms with publicly available data from 2020 to predict arrival time. We preprocess the data by extracting more features, formulate a regression problem, and apply Random Forest, Artificial Neural Networks and Support Vector Regression to train predictors. The Random Forest predictor achieves a low RMSE of 0.88 minutes, where the original delay time ranges from -50 to 50 minutes. To continue the project, real-time data, such as location of the buses, can be used in training to provide even better prediction. Another line of future work would be to incorporate the machine learning algorithms into the SPOT app so students can easily access the prediction.

1. INTRODUCTION

The Charlottesville Area Transit (CAT) is the publicly-owned company that runs twelve bus lines in Charlottesville. In particular, line 7 and line T pass through UVA, carrying students from the universities to apartments and shopping centers. Although the buses run on strict schedules, they can arrive before or after the scheduled time due to different traffic patterns. CAT provides an official SPOT app that shows the current location of the buses, but students still need to predict the arrival time based on the location.

If students mistaken the arrival time, they could be waiting for more than 10 minutes or even longer if the buses have already arrived and departed. To address this problem, the team used machine learning algorithms and publicly-available data to train models that can more precisely predict arrival times.

2. RELATED WORKS

Machine learning algorithms have been proposed to solve problems using large datasets. In his seminal textbook, Bishop (2006) explained the paradigm of machine learning and how it can be applied to many real-world applications. He posited that, given enough data, various machine learning algorithms could train powerful predictors in regression or classification problems. Bishop also used mathematical reasoning to show that these methods indeed converge to a workable solution.

To facilitate applications, software packages had been developed to allow machine learning algorithms to be applied like a black box. Pedregosa, et. al. (2011) developed

Scikit Learn, a Python package that implements various popular machine learning algorithms. The package provided our team easy and clean implementation of the algorithms, which significantly reduced our workload.

The idea of this project was inspired by a similar study done by Kuhn and Jamadagni (2017) at Stanford's CS229 course. Kuhn and Jamadagni applied machine learning algorithms to predict flight arrival delays. Different from our formulation, they only predict whether the flight was delayed or not, which is a classification problem instead of a regression problem. Although this may work in their case, we believe it oversimplifies our problem.

3. PROCESS DESIGN

The main work of this project consisted of fetching and processing data, visualizing data, and applying machine learning algorithms. The entire code was written in a Google Colab environment.

3.1 Data Processing

We first downloaded the publicly available data from this site:
https://opendata.charlottesville.org/datasets/charlottesville::transit-2020/explore

By filtering the line, we were able to extract 50,904 entries of arrival time of the line 7 bus over the time period from 2020/01/02 to 2020/03/01. Each entry has data of the stop, route, datetime, latitude and longitude information.

We loaded the data into a pandas object in our python file. Then, we extracted new features. We first split the datetime field into individual fields including month, day, hour and minute. We then extracted the day of the week based on the month and day information and determined whether the day is a school day or not.

Then we needed to get the delay time instead of the arrival time. We downloaded the expected bus arrival time from the CAT website and expressed the arrival time of each stop in a single number. After subtracting this number from the arrival time to get the delay time, we separated the delay time as the label column and dropped the delay time in the original dataset to use as the features. Finally, we applied simple imputer using the median value of each field and the standard scaler to each field.

After the data cleaning and processing, we acquired a dataset of features including stop, latitude, longitude, month, day, hour, minute, day of week, and is school day, and a column of labels of delay time. We were then ready to apply the machine learning models on this training dataset.

3.2 Visualization

Visualizing the data using various plots, we first plotted different fields as a histogram. It turned out that the delay time was roughly distributed as a normal distribution around mean zero, which fits with our prediction. Figure 1 shows the histogram of delay time.

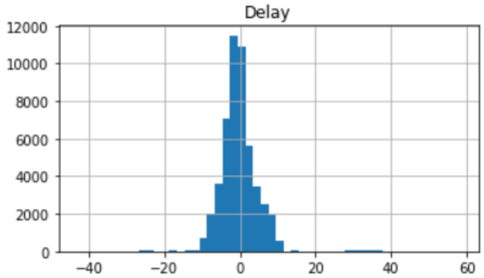


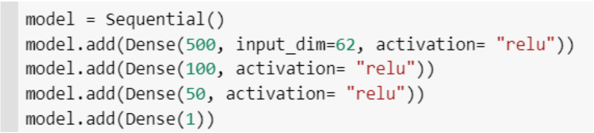
Figure 1: Frequency of delay time as histogram.

Then we calculated the correlation matrix of different features with the label and found that longitude has the highest linear correlation of -0.24, which is still small. We deduced that this is because each individual feature has relatively low effect on the label.

Finally, we plotted the correlation matrix scatter grams using the label and five features of hour, longitude, latitude, minute and day of week. None of the features presented a strong linear correlation with the delay time.

3.3 Models

We separated the dataset into the training set and test set using a test size of 23%. We imported the scikit learn package and directly applied models of linear regression, random forest regressor, SVR with rbf kernel and polynomial kernel of degree 3, and finally a ANN with three dense layer and an output layer. Figure 2 presents our ANN's simple architecture.



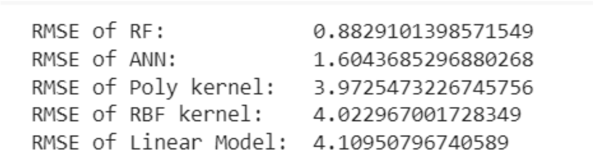
```
model = Sequential()
model.add(Dense(500, input_dim=62, activation= "relu"))
model.add(Dense(100, activation= "relu"))
model.add(Dense(50, activation= "relu"))
model.add(Dense(1))
```

Figure 2: ANN architecture.

Then we trained the models using our dataset and tested the performance using the test dataset.

4. RESULTS

We used the RMSE, or root mean square error, as the performance metric. We were able to achieve RMSE for different models as shown in Figure 3.



```
RMSE of RF:            0.8829101398571549
RMSE of ANN:           1.6043685296880268
RMSE of Poly kernel:   3.9725473226745756
RMSE of RBF kernel:    4.022967001728349
RMSE of Linear Model:  4.10950796740589
```

Figure 3: Root mean squared error of different models.

As shown above, the random forest predictor is able to achieve a low RMSE of 0.88. Given that the original data has a high value of 55 and low value of -45, and the majority of the values between -2 and 3, we concluded that the RMSE is relatively low given the data. On average, users only suffer from an inaccuracy of less than 1 minute from the model.

5. CONCLUSION

We extracted and cleaned data on the arrival time of the line 7 bus, formulated a regression problem, and applied machine learning algorithms to predict the delay time. We were able to achieve low RMSE in our models. This project showed that machine learning algorithms can be successfully applied to meaningfully improve lives in Charlottesville.

Personally, this project exposed me to applications of machine learning algorithms and familiarized me with various Python packages. This project also demonstrated my knowledge of machine learning and coding abilities.

6. FUTURE WORK

One area of future work would be to incorporate the models into user friendly applications such as the SPOT app, which is already widely used to check bus locations. The app could feed the machine learning algorithms with real-time data, and the accuracy of delay time prediction would be greatly improved.

Another line of future work could be to further improve upon our prediction models with more data and more complex models. We only used about two months of data from year 2020, and if more data are used, the accuracy of the models would surely increase.

7. UVA EVALUATION

The underlying course for this project, CS 4774 Machine Learning, succesfully prepared me with the knowledge and techniques to complete the project. However, according to opinions from my peers, I believe the curriculum could be enhanced by separating the machine learning course into two courses: one focused on applications; and another on theory. I have friends who claimed that the current course differs in approach when different professors teach it. In particular, some students feel less prepared in the theory side and struggle in later courses such as reinforcement learning that build on the theory.

8. ACKNOWLEDGMENTS

I would like to thank professor Rich Nguyen for teaching the course. I would also like to acknowledge my teammates in this project, Yancheng Zhou and Zhijun Cao.

REFERENCES

[1] Christopher Bishop. 2006. Pattern Recognition and Machine Learning. Springer, New York, NY.

[2] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 85 (2011), 2825–2830.

[3] Nathalie Kuhn and Navaneeth Jamadagni. 2017. Application of Machine Learning Algorithms to Predict Flight Arrival Delays. Stanford University, Stanford, CA.