NEXT Trucking Company

NEXT Step: Improving Network Efficiency

Predicting High-Performance Drivers

Belle Lerdworatawee

Joshua Lee

Natalia Luzuriaga

Rob Royce

Kyle Whitecross

March 18, 2021

Introduction

Generating more than $700 billion in revenue[1], the freight trucking industry plays a major role in the US economy. However, there still exist many issues such as the inefficient use of time and money, as well as safety concerns with hired drivers. NEXT Trucking aims to improve upon the current trucking system, serving as a platform to effectively match carriers with shippers via cutting-edge predictive technologies. One of the *next* steps for this company is to build a model to classify high-performing drivers, leading to an enhanced network of capable drivers.

Executive Summary

Before any models can be made, it is necessary to analyze the dataset to find correlations, bad data entries, or any other relationships between features. Due to the nature of the data, most of the rows could be collapsed to only include the latest load by each unique driver, although we did not ultimately take that approach. Instead, we found that using the original, non-aggregated, dataset resulted in more accurate predictions. We performed pipeline transformations on the original dataset, which ultimately resulted in a dataframe with 40 columns and approximately 80,000 rows.

Once the data is transformed, a series of classification models can be used to capture high-performing drivers. After fine tuning the hyperparameters to maximize accuracy, it then becomes clear which model can best categorize the drivers by their performance. We also used **Principal Component Analysis**, which was useful in reducing the complexity of the data, but did not result in an overall improvement to our models.

The first classifier we tested was **Logistic Regression**. A t-test was performed on bootstrapped data to indicate the significance of coefficients for each predictor. After finding the best coefficients for the model, it gave an accuracy of 96%. A decision tree classifier utilizing **AdaBoost** was developed as well. Obtained from this were important features that were key to the classification of trees. As a result, the model achieved an accuracy of 95.4%.

We implemented a **Neural Network** using the ReLU activation function and an ADAM optimizer. By using cross validation and fine tuning hyperparameters such as maximum iterations, activation functions, and hidden layer sizes, a score of over 99% was obtained. Finally, we created a custom model that utilized our knowledge of the dataset, rather than an ML model, to predict high-performance drivers.

[1] American Trucking Associations. "New Report Finds Trucking Industry Revenues Topped $700 Billion." Aug 20, 2018. From Cison website, https://www.prnewswire.com/news-releases/new-report-finds-trucking-industry-revenues-topped-700-billion-300699492.html, accessed March 17, 2021.

Methodology

In order to generate labels for the dataset, we used the rows which exceeded specific threshold values. These thresholds were designed to differentiate high-performing drivers from the rest, since a high-performing driver is any driver that is in the top 75th percentile in the “total_loads” columns *and* top 75th percentile in the “most_recent_load_date” column. Since the data has one row per trip, as opposed to one row per driver, we decided to check the top 75th percentile of both the original dataset and an aggregated dataset with a single row per driver. The graph below shows the distribution of “total_loads” and “most_recent_load_date” for both the original dataset and the aggregated dataset. After running several tests on the data to determine a proper methodology, we ended up using **325 as the “total_loads” threshold**, and **Feb 10, 2021 as the “most_recent_load_date” threshold**. This was generated using the original dataframe, without aggregating by driver ID, and without dropping any rows due to null values. This ultimately resulted in a higher average accuracy, presumably because the aggregated dataset inherently loses information during the aggregation process.

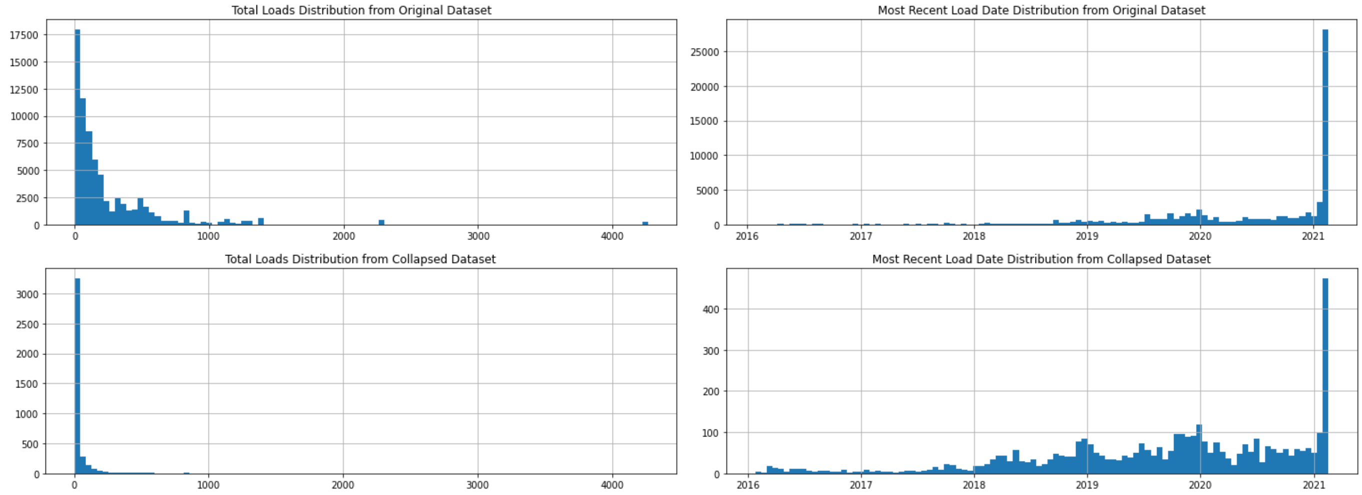


Figure 1: Distribution of High Performance Metrics - Original and Aggregated

After we decided on a proper threshold value for the performance features, we decided to augment the dataset as such:

- Convert text-valued boolean columns to numerical valued boolean columns:
    - interested_in_drayage
    - port_qualified
    - drvier_with_twic

  - dim_carrier_type (name changed to “self_owned”)
  - signup_source (name changed to “mobile_signup”)
  - dim_preferred_lanes (name changed to “has_route_preference”)
- One-hot encode certain categorical columns:
  - carrier_trucks (resulted in 16 boolean columns, one for each truck type)
  - weekday (resulted in 7 boolean columns, one for each day of the week)
- Standardized numerical values:
  - Used for the Neural Networks, PCA, Logistic Regression, and Ensemble models.

The dataset transformations and aggregations resulted in a total of 40 numerical columns. We have included a **correlation matrix** in the figure below.

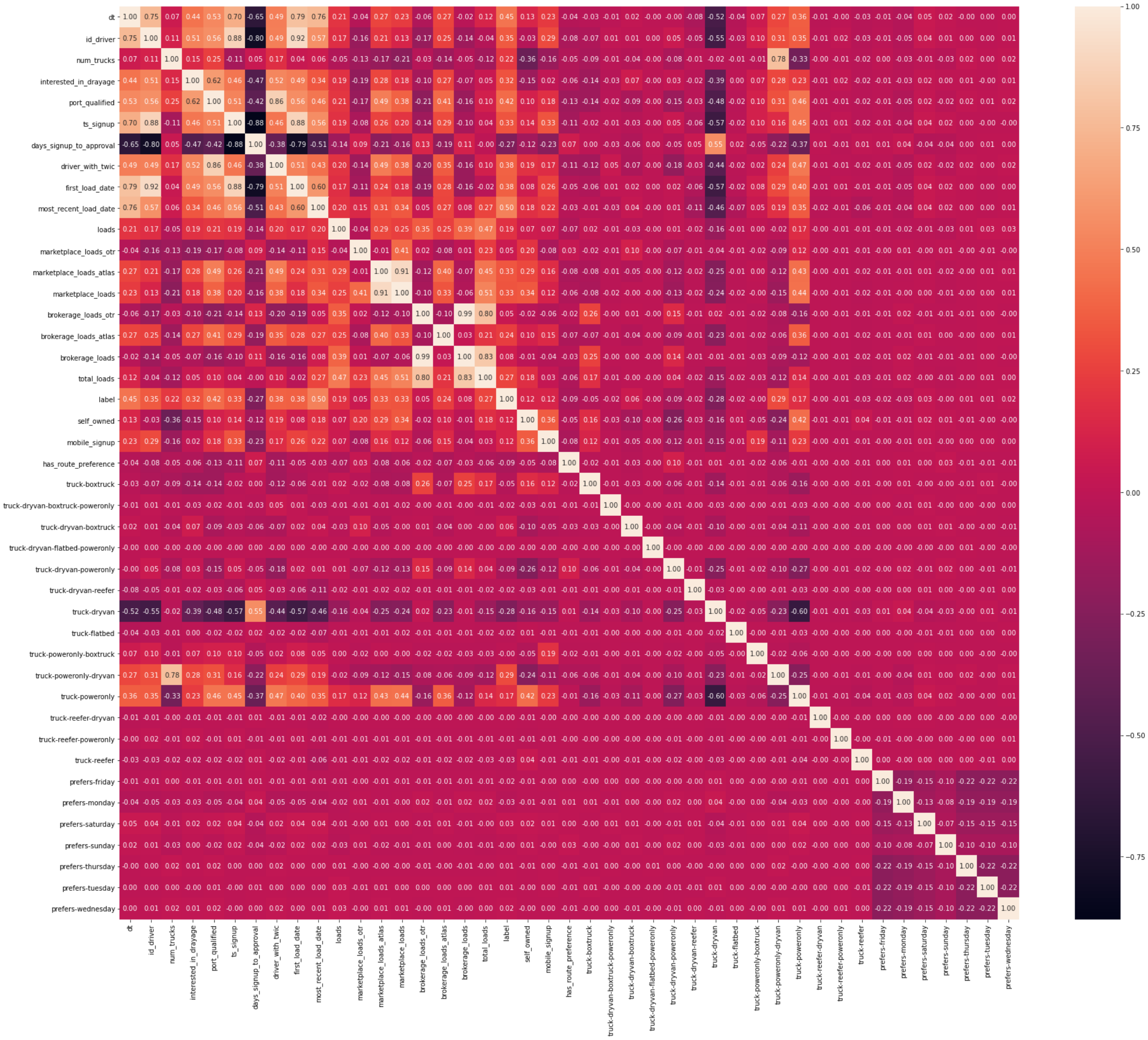


Figure 2: Correlation Matrix

From this matrix, we can see that labels were highly correlated with features involving dates and loads, which is expected from the way we generated the labels. We also see relatively high correlations between various truck-types and amongst the various weekday preferences.

The “dryvan” truck type had significant negative correlations with a lot of the different date features, which suggests that fewer dryvan’s have been employed over time, whereas the “poweronly” truck-type frequently has a positive correlation with the date features, suggesting more of this type has been employed over time. Notably, the truck-type “poweronly-dryvan” had the highest correlation with the actual labels out of all the truck types.

A Logistic Regression model was built to try to predict high-performing drivers from the dataset. The idea was to build a model using all the predictors, and subsequently use bootstrap to determine the significant ones. First the data was split into a training set and a testing set. A logistic model was trained on the former set to get a baseline of how well logistic regression would perform. Then a **bootstrap** with 150 samples was performed to obtain a distribution for each predictor’s coefficient. From the resulting estimators, the mean and standard deviation of each predictor was calculated and then used to perform a **t-test** on each predictor.

We decided to use an alpha value of 0.05 for the threshold, and eliminated all predictors whose p-value was greater than 0.05. Next, a **Variance Inflation Factor** (VIF) test was performed on the remaining predictors to determine which ones were highly correlated. After insignificant and highly correlated predictors were removed, a final logistic regression model was trained on the remaining columns.

Additionally, a model using AdaBoost was trained and developed to find significant predictors. The base classifier of the AdaBoost model was a DecisionTreeClassifier with a maximum depth of 1. After experimenting with multiple hyperparameters, the best-performing model that produced the highest accuracy score was the model with 100 n_estimators or trees and a learning rate of 0.01. **K-Fold Cross Validation** was performed using the RepeatedStratifiedKFold class with n_splits equal to 10, providing the mean accuracy and standard deviation.

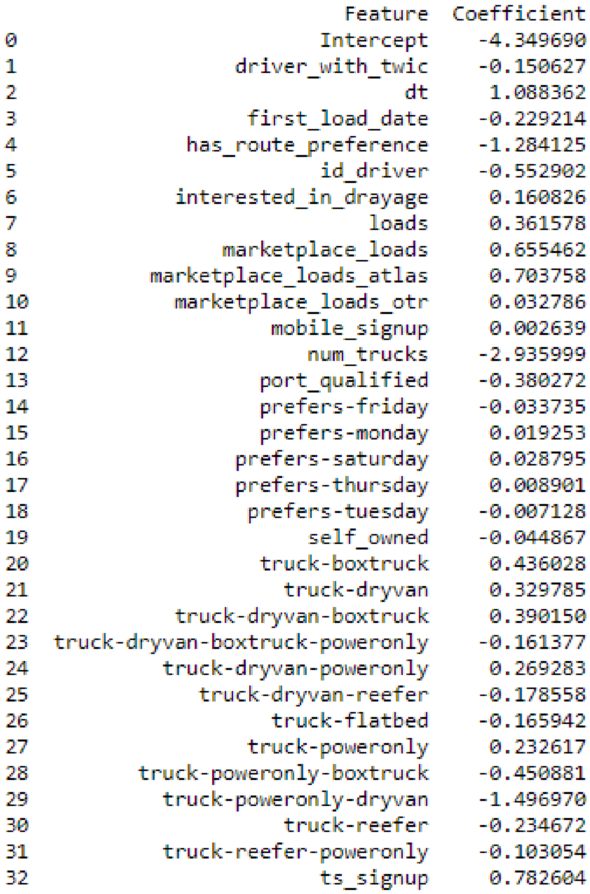
Multiple different neural networks were used to model the data. The first neural network had a single hidden layer with 100 neurons and Rectified Linear Unit (ReLU) activation functions. It was optimized using the Adam optimizer with a learning rate of 0.001, an L2 regularization coefficient of 0.0001, and the default adam beta 1 and beta 2 coefficients. The second neural network was an older fashioned model with 1 layer of 10 hidden neurons, each with a sigmoid activation function. It was optimized with stochastic gradient descent and momentum, with a learning rate of 0.001, an L2 regularization coefficient of 0.0001, and a momentum coefficient of 0.9. Both models were cross validated along 10 folds, and learning curves were plotted for the first model.

Finally, we created a custom model that used our knowledge of the dataset, rather than applying a direct ML approach. This model leveraged the fact that, while each instance-label pair in the data might be unique, the 2 quantities used to create the label were driver specific, meaning that each instance with the same driver had the same label. Furthermore, since there were roughly 15 distinct instances of each driver ID, the likelihood that a small random cross-validation sample contained all of a drivers instances and left none in the training set was exceedingly unlikely. In essence, the custom model could just look up a driver in the training data, and with high probability, find the driver and predict the correct label. If the model couldn’t find the driver, then the likelihood that the driver was in the top 25% of total loads, and all of the instances adding up to those total loads were in the small random sample, was so small that the model could just predict 0 for the label.

Results

Logistic Regression

Looking at the results from the logistic regression model, we can see that the predictors “dt”, “has_route_preference”, “num_trucks”, and “truck-poweronly-dryvan” have the largest coefficients. Thus, these are the more significant predictors. Respectively, the features have coefficients of 1.088, -1.28, -2.93, and -1.50. This indicates that for a unit increase in num-trucks, for instance, we should expect an increase of about -2.93 in the log of odds of that driver being a high performing one. Another finding based on this result is that only truckers with a *dryvan* are less likely to be a high-performing driver.



```
                              Feature  Coefficient
0                           Intercept    -4.349690
1                      driver_with_twic    -0.150627
2                                  dt     1.088362
3                     first_load_date    -0.229214
4                 has_route_preference    -1.284125
5                           id_driver    -0.552902
6                interested_in_drayage     0.160826
7                               loads     0.361578
8                   marketplace_loads     0.655462
9             marketplace_loads_atlas     0.703758
10              marketplace_loads_otr     0.032786
11                      mobile_signup     0.002639
12                         num_trucks    -2.935999
13                      port_qualified    -0.380272
14                      prefers-friday    -0.033735
15                      prefers-monday     0.019253
16                    prefers-saturday     0.028795
17                    prefers-thursday     0.008901
18                     prefers-tuesday    -0.007128
19                         self_owned    -0.044867
20                      truck-boxtruck     0.436028
21                        truck-dryvan     0.329785
22               truck-dryvan-boxtruck     0.390150
23     truck-dryvan-boxtruck-poweronly    -0.161377
24              truck-dryvan-poweronly     0.269283
25                 truck-dryvan-reefer    -0.178558
26                       truck-flatbed    -0.165942
27                     truck-poweronly     0.232617
28            truck-poweronly-boxtruck    -0.450881
29              truck-poweronly-dryvan    -1.496970
30                        truck-reefer    -0.234672
31              truck-reefer-poweronly    -0.103054
32                           ts_signup     0.782604
```

Figure 3: Feature Importance from Logistic Regression

PCA

The PCA results indicated a moderate degree of collinearity among the dataset. Notably, the first two components were able to explain 30% of the variance of the dataset, indicating several redundant features. Beyond these initial components, following components were not able to explain much more variance than they would without PCA. This meant that around 30 components could explain roughly 99% of the variance, which indicates around 10 features are highly collinear, and the data could be represented just as well with around 25% less features.

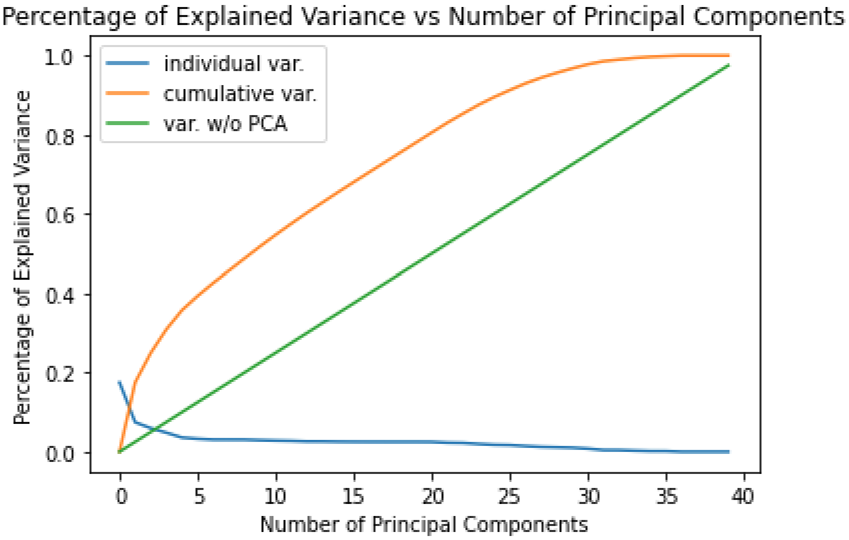


Figure 4: PCA Variance Graph

Adaboost

Analyzing the AdaBoosting Model’s feature importances revealed the three most significant predictors: “marketplace_loads”, “brokerage_loads_otr” and “borkerage_loads” with importance values of 0.26100, 0.21900, and 0.18300 respectively. After being cross validated, the model achieved a 0.953 accuracy score on the test set with a small standard deviation of 0.006.

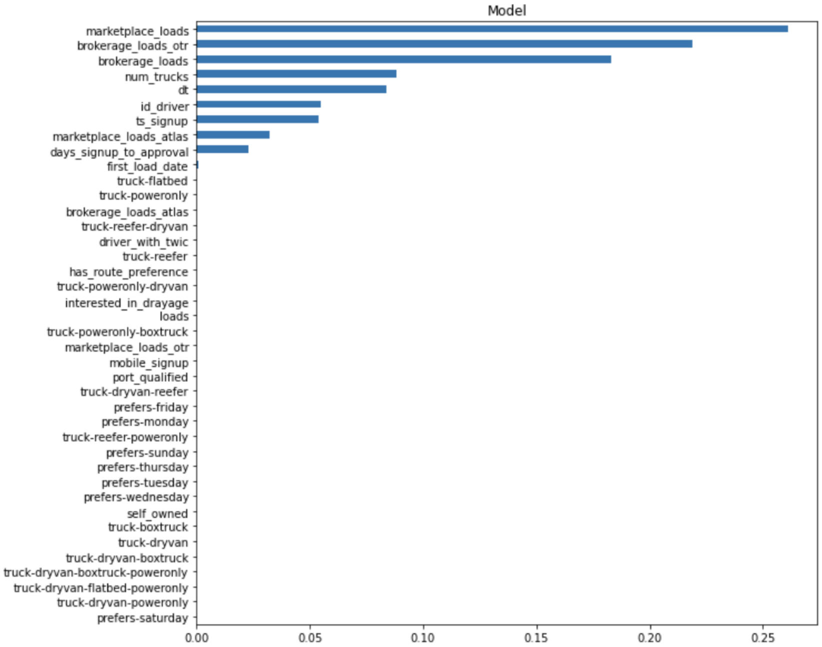


Figure 5: AdaBoost Model’s Feature Importances

Neural Network

Both neural networks performed exceptionally well. The first neural network achieved a mean cross validation accuracy of 0.9999, and the second achieved a mean cross validation accuracy of 0.9561. Both models took significantly longer to train than any other model in this report, and interestingly, the second model took much longer to train, despite having less neurons.

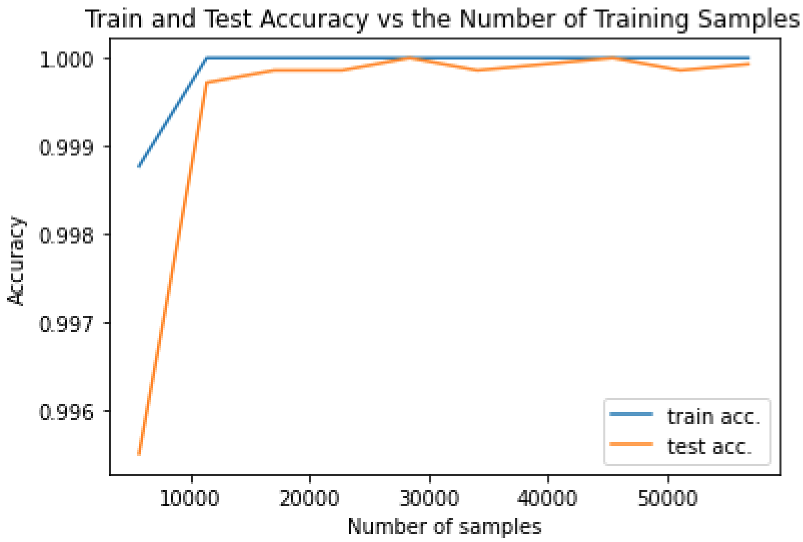


Figure 6: NN Accuracy

The learning curves for the first model indicate that 5,000 samples is not enough to converge, 10,000 is enough to converge but slightly overfits, and 15,000 and beyond is enough for the model to have a very high training and test accuracy. These results are consistent with the modern neural network research. The larger model, with ReLU activations instead of sigmoid activations, trained with a newer optimizer, Adam, dramatically outperformed the smaller model. The higher number of neurons in the hidden layer allows the network to learn more complex relationships. The ReLU activation prevents the "dying gradient" problem prevalent in sigmoid-based networks, and the Adam optimizer dynamically adapts the learning rate and weight updates intelligently to converge faster. While the first neural network performs very well, it's not perfect, and misses a few examples.

Custom Model

The custom model performed exceptionally well. On a shuffled 84-fold cross validation evaluation of all 84,000 training samples, the custom model had a perfect training and cross validation accuracy. It also achieved 100% accuracy on the test set provided by the instructors that contained no labels.

Recommendations

We were unable to tell with statistical significance which features resulted in more high-performing drivers. This seems to be a result of the methodology we used to develop the labels and real-world results are likely to be exceedingly different. However, from our AdaBoost results, it seems plausible that high-performing drivers may tend to take on more “marketplace” and “brokerage” loads. Similarly, from our Logistic Regression analysis, it seems that drivers with a larger number of trucks (most likely Fleet’s) appear to perform at a higher level than drivers with fewer trucks. This might imply that employing Fleet’s would result in higher performance overall.

One of the issues we ran into was that of generating labels. Since a high-performing driver is any driver that meets the threshold of both “total loads” and “most recent load date,” it is important to decide what “75th percentile” means. We first assumed that aggregating by driver ID would give us the most accurate results, because otherwise certain drivers would be counted more than once when looking at the distribution of dates and loads. However, this turned out to be an incorrect assumption when compared to the test data. Overall, we found the metric of what determines a high-performing driver to be a bit arbitrary.

Conclusion

In conclusion, after a thorough analysis of the data provided, several predictive methods were developed to describe the performance of a “high-performing” truck driver. We used Logistic Regression to predict high performance at an accuracy of 0.96, which led us to conclude that the most important coefficient was the intercept. Principal Component analysis was applied to the dataframe to discover that the data could be reduced in size by 25% with minimal variance loss. An ensemble method, AdaBoost, was applied to predict the data with an accuracy of 0.95, and further analyze the most important features. Two different neural network architectures were applied to the data, with one achieving a virtually perfect accuracy of 0.999, and demonstrating the power of modern machine learning techniques. Finally, a custom model was built to take advantage of the overlap between the provided training and testing data. We did not conclusively demonstrate which features are most significant to predicting a high-performance driver, but this seems to be a result of the method used to generate the labels, and not due to the models we developed.

References

American Trucking Associations. “New Report Finds Trucking Industry Revenues Topped $700 Billion.” Aug 20, 2018. https://www.prnewswire.com/news-releases/new-report-finds-trucking-industry-revenues-topped-700-billion-300699492.html, accessed March 17, 2021.