# CS 188 Project 3 Report


| | |
|---|---|
| Mingchao Lian | 005348062 |
| Zijian Zhao | 005355458 |
| Qing Shi | 805140765 |
| Qingyang Li | 705392481 |

# 1. Executive Summary

In this project, we aim to develop a classification model to better determine whether or not a driver will be a high-performing one for NEXT. based on the 83414 driver information from 1/26/2015 to 2/17/2021. We will analyze the dataset and see which features are helpful for our model construction and which features contribute/correlate more with the label to predict.

We first process our data to make it easier to input into the model. We checked the correlation matrix to decide which features are correlated with the label and other labels. We split the data we have into 80% training and 20% testing. Four models were made to do the classification and test their accuracy In this project, we always use F1 score as our metric. We did a bagging classifier, a neural network model, a support vector machine, and a decision tree. For Kaggle submission, we picked the SVM and predicted the label of 1000 drivers based on 1000 different driver information and we achieved 0.954 on the mean F1 score.

# 2. Background/Introduction

The trucking industry serves the American economy by transporting large quantities of goods from manufacturing plants to retail distribution centers. However, truck driver shortage has always been a problem, and therefore driver costs are the most significant challenge faced by the industry.

NEXT offers an end-to-end logistics solution that combines a freight marketplace with company drivers and strategic partnerships to provide a powerful suite of services for Drayage (shipping goods a short distance via ground freight), OTR (over the road, referring to truck drivers hauling freight over long distances), and transloading (the process of transferring a shipment from one mode of transportation to another.).

To help the company achieve their goal and reduce freight costs, we are interested in building a predictive model capable of determining overall network capacity to handle incoming shipping requests as well as more effectively recruiting high-performing drivers to enhance their network. This report illustrates how we developed a classification model to determine whether or not a driver will be a high-performing one.

# 3. Methodology

## 3.1 Label Generating

We first label the drivers in the 75th percentile of ‘total_loads’ and the 75th percentile of

‘Most_recent_load_date’ as 1, means one is a high-performance driver and the rest of them are labeled as 0. We then found that we have 73021 drivers with label 0 and only 10393 label 1. To balance the dataset, we use random undersampling. Since we have enough samples, we just randomly dropped 40000 data out of 73021 so we will approximately achieve a 1:3 positive/negative ratio.

## 3.2 Data Strategy

From the result of correlation analysis we did before, we noticed that ‘year’ and ‘id_driver’ do not correlate much with the rest of features, and thus we dropped them. Also, we dropped ‘date’, ‘weekday’, ‘id_carrier_number’, ‘dim_carrier_company_name’, ‘ts_signup’, ‘ts_first_approved’, ‘first_load_date’, ‘load_day’, ‘days_signup_to_approval’ as they are features related to date and time and do not relate to our analysis. We dropped ‘dim_preferred_lanes’ because most of the drivers do not have a preference. We finally dropped ‘home_base_city’ for the reason that there are too many cities, and it is difficult to one-hot encode and interpret all of them. We use ‘home_base_state’ as a reference instead.

As a result, our final data selection for categorical features is: ‘dim_carrier_type’, ‘driver_with_twic’, ‘home_base_state’, ‘carrier_trucks’, ‘interested_in_drayage’, ‘port_qualified’, ‘signup_source’. Out final data selection for numerical features is: ‘num_trucks’, ‘loads', 'marketplace_loads_otr’, ‘marketplace_loads_atlas’, ‘marketplace_loads’, ‘brokerage_loads_otr’, ‘brokerage_loads_atlas’, ‘brokerage_loads', 'loads_per_truck’.

For categorical features, we impute an arbitrary state for home_base_state. Then we applied one-hot encoding for them.

For numerical features, we impute a constant 1 for the number of trucks since a driver will have at least one truck. Then we applied a standard scaler on them.

We also augment a new cross-feature loads_per_truck into the dataset, as `df["loads_per_truck"] = df["loads"]/df["num_trucks"]`.

## 3.3 Kaggle

For the kaggle submission, we performed several validations to find the best model along with their best parameters. The candidate models we selected were Supported Vector Machine, decision tree classifier, and the MLPClassifier in the neural network. We coded a function that streamlines the data preprocessing. Within the preprocessing period, except for dropping the unwanted features as mentioned above, we also decide to drop the “home base state” and “carrier type” features because the unique count of these two categorical features are too large compared

with the provided “score_V3.csv” file. Then we trained the same training over the three different classification models:
`sklearn.svm.SVC`, `sklearn.tree.DecisionTreeClassifier` and `sklearn.neural_network.MLPClassifier`. Next, we test the models with sklearn.metrics.f1_score using the same validation set, it turns out that the support vector machine model gives the best result, with a score of around 0.93, while the decision tree has f1_score around 0.9 and the neural network model scores around 0.89. For a better result, we also use cross-validation technique to find the best penalty term for the SVM model, which gives us that the best parameter to use is between C=1.0 and C=1.2. Finally, we train the model again using all training data and use the test cases in “score_V3.csv” and generate predicted labels. Our best submission result has a mean f1_score of 0.954.

# 4. Results

## 4.1 Basic Statistics on Variables

For problem 3, we ran the correlation analysis and displayed a summary on our data using describe function.

From the correlation analysis, we are able to find that there is a strong negative correlation between days_signup_to_approval and year and there are positive correlation between marketplace_loads and marketplace_loads_atlas, brokerage_loads and brokerage_loads_otr. This might be due to the reason that predictors related to loads are correlated. What’s more, we find that marketplace_loads and marketplace_loads_atlas are most correlated with the label.

From the describe function, we are able to find the count, mean, standard deviation, min value, max value, and 25%, 50%, 75% percentile of each predictor. For example, for the num_trucks, we observe that each carrier has approximate 20 trucks on average, and the minimum and maximum trucks that a carrier could have is 1 truck and 195 trucks. More than 25% of carriers only have 1 truck, more than 50% carriers have fewer than 2 trucks, and the 75% percentile for num_truck is 11 trucks. This indicates that we are having outliers of carriers with large numbers of trucks.

## 4.2 Neural Net classifier

For the neural network, we decide to use the architecture of 3 hidden layers each has 5 perceptrons and ReLU as the activation function. We fit multiple models with different hyperparameters and compare the F1 score from them, then we select the best combination of

these parameters for our model. We found that SGD solver with adaptive learning rate starting with 0.04 and L2 penalty 0.001 has the highest F1 score 0.979 on 80/20 split data among all our neural network models. We also apply the same method on our data set after PCA. The best model is constant learning rate 0.01 and L2 penalty 0.0001. However, the PCA data doesn't give us a good result as the normal ones did. We got an F1 score of 0.735.

### 4.3 Cross Validation

For cross validation, we employed K-Fold cross-validation to our training regimen for both ensemble and NN classifiers. We first use the KFold cross validation method with the default 10 folds and 100 number of trees. That is, we divide our training data into 10 groups and regard 1 group as test data and the other groups as training data to fit our models. We also use the F1 score as our scorer. After employing cross-validation to our models, we are able to get the result score of 0.997958 accuracy with a standard deviation of 0.001061 for ensemble classifier, and 0.961455 accuracy with a standard deviation of 0.011447 for neural network classifier. This indicates that our ensemble model has a better ability at labeling new data than the NN model.

We also employ a stratifiedshufflesplit along with KFold cross validation. With stratified shuffle split, we are able to get the result score of 0.997567 accuracy with a standard deviation of 0.000597 for ensemble classifier, and 0.965697 accuracy with a standard deviation of 0.001347. This indicates that with stratified shuffle split, our ensemble classifier still has better performance on labeling new data than the NN model. In addition, the stratified shuffle split ensures equitable distribution along a key parameter and makes each dataset an approximate representation of original data, indicating the result score of 0.997567 accuracy and 0.965697 accuracy for ensemble and NN respectively may be a better evaluation of the models' abilities.

## 5. Discussion

From the correlation analysis, we found that marketplace_loads_atlas and marketplace_loads are most correlated with the label and contribute the most for being a high-performance driver. And we found marketplace_loads is the sum of marketplace_otr and marketplace_atlas, which means drivers with higher ATLAS load tend to have higher performance. Thus, the number of ATLAS (a drayage type of the loads) covered by drivers from the marketplace (the company's app) is a feature worthy of paying attention.

In future analysis or predictions, we can try to acquire more data. Having more data is always helpful for training a model. More data usually means more room to make mistakes since we can just pick from a big range. Likewise, we can experiment with more features along with data to train the classification model. Features and their cross transition will make the number of

features increase exponentially so we need to try different ways to decrease dimensionality. In addition, we can treat missing or outlier values more carefully such as surveying the suitable imputation value for each feature with missing values.

## 6. Conclusion

We first generate labels for our dataset where drivers in the 75th percentile of ‘total_loads’ and ‘most_recent_load_date’ as 1, and other drivers as 0. We use the correlation analysis and find marketplace_loads and marketplace_loads_atlas are most correlated with the label. We imputed the feature ‘home_base_state’ as an arbitrary state, and numerical data as 0. We also augment a new feature loads_per_truck. For categorical variables, we applied one-hot-encoding and for numerical data, we applied standard scaler. We implemented a linear regression and a logistic regression for feature importance and PCA to reduce dimensionality. We develop ensemble and neural network classifiers on the datasets, and cross-validate these two models. From the result of KFold cross validation, we observed the result score of 0.997567 F1 score and 0.965697 F1 score for ensemble and NN respectively.