Prediction of parking plot in SF: An Ensemble Method

Team TensorFlow BOYS[1]

Introduction

Due to high density of population in downtown area, finding an available parking spot in San Francisco has been a big trouble for drivers. However, even in rush hour, not every parking spot is occupied. There might be a suitable parking spot for their car only one block away, but people might fail to acquire that knowledge.

In this experiment, we developed a fast, reliable model based on historical parking data in certain areas in San Francisco. With this model, we can predict whether there will be a parking spot available according to information like latitude, longitude, current time, and day of week.

We evaluated the performance of this model from several aspects. First, we achieved a F0.5 score of 0.578 on kaggle public board. F0.5 score is an evaluation metric measuring accuracy using the statistics precision p and recall r. Compared with F1 metric that weights recall and precision equally, F0.5 measure weighs recall lower than precision. The reason we adopted F0.5 instead of F1 is that making sure the user have a valid parking spot is more important than finding a parking spot a little bit further (which would happen when we wrongly predicted False for a street that actually have a parking spot). Second, testing on validation set shows a precision rate of 61.0% and recall rate of 68.9%. In this study, we splitted the available data into 73% training set (1000) and 27% validation set (300). Models were trained on training set and evaluated on validation set. This situation where $p > r$ also reveals the fact that precision was given more weight than recall during training process.

The remainder of this report is organized as follows. In section 2, we discussed about features of the data and how we did feature engineering. We introduced the models we employed in section 3. Results were in section 4. This is followed by a general discussion regarding our approach with thoughts on future work in section 5. We conclude in section 6.

[1] Yiqiang Zhao, kaggle id: YiQ-Zhao
Fang Wang, kaggle id: fwang18

Exploratory Data Analysis & Feature Engineering

The training data has 1,100 observations, in which there are 96 unique streets (combinations of Street, From, and To). However, only 70 streets can be identified since many of them have same street names in Street & From or Street & To. In the period of Jan 18, 2014 - Mar 9, 2014, the observer recorded 401 times (36.5%) of streets having available parking spots against 699 times (63.5%) of no parking spots found. Most of the time, the observations happened in the day time, and it is almost equally distributed in weekdays and weekends.

As regard to the additional datasets, we can match 51766 payment records from the parking meter data to the identified streets in the training data but only 10 streets were found overlapping with blocks in sensor data. Overall, the parking meter data contains all streets in the training data, while the sensor data only gives us 10.4% (10/96) streets additional information.

With the knowledge we acquired from EDA, we conducted feature engineering for parking, sensor and training datasets in various methods.

For parking meter data, we employed some clever but simple math knowledge to identify which street that record occurred. Since a parking spot (P) usually located on a street where we have two nearby crossroads on two sides (let's say A, B), P is on the line AB. It's easy to prove that AP+BP=AB only in this case, but AP+BP>AB if P is not on the line AB but forms a triangle with point A and B. With the latitude and longitude of point A, B, and P, we can easily join the parking meter data into our training data. Here we set the threshold of distance difference as 10 meters.

For parking sensor data, we utilized a similar method with expanded threshold as 300 meters. As we stated before, only a small part of parking sensor data overlapped with training data. To take full advantage of that data, we assigned parking sensors' data to the most nearby street. Even in this way, only 23 (23.9%) streets in training data matched with sensor data.

Apart from geocoding for parking meter data and parking sensor data, we also did mean encoding, label encoding and missing data imputation to training data.

Model Selection

We did experiments with two models: logistic regression and random forest.

Logistic regression is a classic model widely used in binary classification problems. Instead of predicting a numeric variable based on a set of input data, logistic regression gives a probability that this point belongs to a class. Usually, we would take the class with higher probability as the prediction output. However, as we introduced in why we select F0.5 score part, precision (making sure the user can find a valid parking spot) is more important than recall (the driver might have to driver further to find the parking spot). Thus, we also did some experiment on the threshold we could use here.

Random forest is a popular ensemble learning method for regression and classification. With bagging idea and random selection of features, random forest tends to have a strong but unstable performance on prediction results. For this problem, we used grid search to find the optimal selection of parameter set on F0.5 score.

We also did some experiments on XGBoost models. Like random forest models, the validation score is not very stable.

To avoid overfitting, we also ensemble the models by voting. There are several options to ensemble the model: 1. a weighted probability based on score; 2. Voting based on different models' prediction. Since what we trained is classifier rather than regressor, we simply used the second method. The reason we didn't assign weight to different models' classification is that 1. Models we use have similar score and result (90% predictions are same), so weight won't change the result a lot; 2. Since we don't know the data in public leaderboard, we cannot use back-propagation to update this weight.

Results

We used approximately 73% of training data for training, and left the other 27% as validation set. Here is a list of our best models' performance. As we did not try ensemble on validation set, we cannot report validation precision and validation recall for ensemble.

Method	Publicboard Score	Validation Precision	Validation Recall
Random Forest	0.556	0.682	0.532
Logistic Regression	0.578	0.610	0.688
XGBoost	0.547	0.670	0.504
Ensemble	0.578	/	/

Conclusion and discussion

In this experiment, we managed to build a model to predict available parking spot. Although this model has 70% accuracy on validation set, it does not achieve a satisfying score on public leaderboard. As regards to true positive and false negative, Random Forest and XGBoost have better performance on validation recall, while logistic regression outperforms on validation precision. In this section, we want to discuss about current problems and possible future work.

First, overfitting is a serious problem. Due to the fact that we have only ~ 1000 available data points for training and validation, it's very easy for our model to overfit. I believe this problem also exists in other group's work. Second, also because of this reason, we have a too small validation set to indicate how good (or how bad) our model is.

To overcome those problems, we thought about including external data like local business density, local residence density, and local business density. Also, we could cluster the streets based on similarity, and do mean encoding for every cluster. With limited time and intensive homeworks, we haven't been able to practice those good, creative thoughts. We believe our model will have a better performance with all those features included, and will provide San Francisco citizens with help if put into production!

Team Responsibilities

Yiqiang's work mainly focused on EDA and feature engineering, and Fang contributions were centered around modeling. We finished the writing report together.