Bike sharing demand forecast

Domain Background

Bicycle sharing services are platforms for renting and returning bicycles as needed basis. The services were used to be accessible through a network of kiosk locations throughout a city, in which registered users have to rent and return at some particular locations. The rise of bike rental companies like oBike means that these services can now be accessed with a mobile phone application, allowing users to locate the nearest available bicycle and rent it with a couple of tap on mobile phone application.

These accumulated rental records provide information such as the duration of travel, departure location, arrival location, and time elapsed. These allow monitoring the condition of bicycles(if wheels wear out), studying mobility in a city, predicting the demand of bicycle during the time of a day.[1,2] Therefore, the data can be utilized to help improving company's services towards users, such as meeting the demand of usage by supplying optimal amount of bicycles, reducing possible accident due to the faulty brake system or worn out wheels, and setting up new service locations to cater the services to region where it was historical unavailable. Above problems can be addressed via getting insightful information by analyzing the existing data. To develop a prototype model, I took the bike sharing data from a Kaggle competition(Bike Sharing Demand, https://www.kaggle.com/c/bike-sharing-demand) for this work. This project will provide insights on the factors that affect the demand of rental bike in Washington, D.C.

Problem Statement

The goal of the project is to utilize historical usage patterns with weather data to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C. The inputs consist categorical (e.g. weather, season, etc) and numerical data (e.g. temperature, windspeed, etc). The output, count of rental bike, is a numerical output, therefore the project is a regression problem.

Datasets and Inputs

The data used in the projects can be obtained from Kaggle website (https://www.kaggle.com/c/bike-sharing-demand/data). The original data was obtained from Capital Bikeshare(https://www.capitalbikeshare.com/system-data) and it was used as one of practice competitions on Kaggle.

Hourly rental data spanning two years are provided. For the training set, the data comprises the first 19 days of each month(total data point is ~ 10886), while the test set is the 20th to the end of the month. This project is tasked to predict the total count of bikes rented during each hour by using only information available prior to the rental period.

To predict the outcome, 10 input features (date, time, season, holiday, working day, weather, temperature, "feels like" temperature, humidity, windspeed, casual and registered users) are provided. Count, which is the number of total rentals, is to be predicted based on weather information and the day and time. As this is a time series data, the validation and test dataset must be split chronologically to avoid the "leakage" of future data into the prediction.

Solution Statement

For this project, I will explore boosting method as a solution to the problems. Boosting method is one of the ensemble model that uses boosting to improve the performance of the model. It trains and improves the subsequent learner based on the previous learner. In this case, we can use a decision tree model as the base model for training. The decision tree works by answering a series of questions that enables better purity in the leaves. The model makes predictions by calculating the weighted average of all the predictions from all learners that were trained. This reduces the biases and variances from each individual learner, thus giving better and more generalized predictions.

Benchmark Model

According to competition, the result should be benchmarked with Random Forrest Model. Random Forrest Model is an ensemble method that combining several weak learner(i.e. decision tree) to enhance and improve the overall prediction accuracy. Compare to most

ensemble model, Random Forrest offers relatively high performance with shorter training time; the model can also handle missing values; it can be modeled with categorical inputs; given enough trees, the model tends not to overfit.

Evaluation Metrics

The prediction outcomes are evaluated with the Root Mean Squared Logarithmic Error (RMSLE). The RMSLE is calculated as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i+1)-\log(a_i+1))^2}$$

Where:

- n is the number of hours in the test set
- p_i is your predicted count
- a_i is the actual count
- $log(x)$ is the natural logarithm

Project Design

Before starting to use apply any machine learning program to the data, we should start by performing data exploratory. First we perform feature engineering to the data. One of feature that we should include will be the day of the week (Monday etc) as a feature to dataset. Then, we can perform missing value analysis to see if there are any missing values. This is to understand if there are needs to engineer new features to account for the missing values. After that, outlier and correlation analysis should be performed. Depending on the result of correlation analysis, some of the outlier may have to be removed to obtain a more generalized model. To help visualizing the data, we can also plot the distribution of data. Before we proceed to run model fitting on the data, we should identify all categorical features and performing feature engineering (one-hot encoding etc) to the categorical data so that the data can be used for any models.

Once the feature engineering and data exploratory are done, we can start fitting the data to machine learning models. Before performing the model fitting, we should use some of the

dataset as validation set. If the size of dataset is small, cross validation method will be used to improve the accuracy of the model. We can start learning the data with the benchmark model, which is Random Forrest (RF) model. To achieve better performance than the benchmark model, I will be using the boosting method (including Adaboost and Gradient Boosting). To improve the model, we can fine-tune the model parameters (e.g. no. of estimators, learning rate, etc). Furthermore, we can obtain the feature importance from the model. This will help us understanding the key features of the problems and we can get a simplified model by fitting with the key features.

Reference

1. Ashqar, Huthaifa I., et al. "Modeling bike availability in a bike-sharing system using machine learning." Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2017 5th IEEE International Conference on. IEEE, 2017.

2. Datta, Arnab Kumar. Predicting bike-share usage patterns with machine learning. MS thesis. 2014.