Evaluating Predictions on Bike Rental Frequency

Phanindra Kumar Kannaji, Venkata Rami Reddy Bujunuru
Graduate Student
UMBC - CSEE
{pkanna1, bo26494}@umbc.edu

***Abstract*—The purpose of this project is to develop a model which will predict the approximate number of bikes rented on a particular day or hour based on the Capital Bike share data. The predictions are made by combining the bike share data with daily weather data and holiday data of Washington DC. The models that are used for prediction is evaluated based on the accuracy of the predictions. Various models are evaluated such as Random Forest model, Bootstrap Aggregation, Conditional Decision Trees(C-trees), K-Nearest Neighbors (KNN), Support Vector Machines (SVM) for classification. From our observations, the best model for this bike share data is Random Decision Forest model on decision trees which is one of the ensemble learning methods for classification. The accuracy of this model is approximately 80%.**

I. Introduction

Bicycle sharing system started to grow in the year 1996 at Wisconsin, North America where red painted bikes are used for public service primarily students. In 2010, capital bike share started the bike share service in the area of Washington DC which is open to public. Heavy traffic in the capital areas and the demand for environment-friendly transport promoted the use of the bike share more frequently. The number of bikes that are rented are gradually increasing year by year. The usage of bikes depends primarily on weather and whether its a holiday or not.

The bike rental frequency is the number of bikes that are rented for a given hour. The elements/conditions of the weather that can be considered in deciding whether the bikes will be rented or not are :

- rain
- snow
- wind speed
- inclement weather conditions
- temperature
- humidity
- cloud cover etc..

The predictions related to any statistical models can be done by regression, classification, clustering etc,. which depends on the data. The bike share data is more suitable for classification type of models since the prediction is not an exact value but a class or a range of values. There are various machine learning models which can be applied to the bike share data to predict the busyness of the bike rentals during an hour. The accuracy of prediction is calculated for all these models and the best model with highest accuracy will be used to estimate the bike rental frequency in the future.

II. Motivation

There is a significant impact of the method of transportation an individual choose on the global climate. The heavy traffic and the fueled vehicles are one of the main reasons for the environmental pollution. A bike share program is the best approach for this problem as the bikes are rented at the starting point A and are left at a point B and within that paid hours, any other bike can be used to go to any other point C. The price that the user have to pay is based on the amount of time the bike is taken out of the slot and put back again. There is a demand and supply gap at some particular hours for example, when it is a holiday and sunny outside, the bikes rented will reach the saturation and leads to unavailability of the bikes. To resolve this problem, we created a model to predict the number of bikes rented on hourly-basis so that the demand and supply gap can be minimized and bikes will be available all the time which makes this a reliable mode of transportation. The idea is taken from Kaggle[4], a machine learning competition program.

III. Data Formation

The primary datasets involved in the data that have to be provided to the machine learning model are:

- Capital Bike Share data
- Weather data
- Holidays data

A. Bike Share Data

The bike share data[1] consists of the information regarding the bikes rented per bike rent instance from January, 2011 to June, 2016. Following are the features/variables of the dataset:

- Duration - Duration of trip
- **Start date** - Includes start date and time
- End date - Includes end date and time
- Start station - Includes starting station name and number
- End station - Includes ending station name and number
- Bike Number - Includes ID number of bike used for the trip
- Member Type - Lists whether user was a Registered or Casual member.

  Since the data contains all the bike rental instances, we performed a few operations on the data to calculate the hourly count and the time is split into year, month, day, day of the week, hour and the count for that hour is calculated by means of a GROUP-BY kind of functionality.

The season is also added to the data based on the month as mentioned in Table I.

Season	Months
1	Jan - Mar
2	Apr - Jun
3	Jul - Sep
4	Oct - Dec

TABLE I: Seasons based on months

B. Weather Data

The weather data[2] is provided by Forecastio through an API offered by Darksky.net. The API will give the hourly data for a given latitude, longitude and day. Some important details provided by the Darksky.net are:

- **Apparent Temperature** - Feels like temperature during that hour
- **cloudCover** - The percentage of sky occluded by clouds, between 0 and 1, inclusive
- dewPoint - The dew point in degrees Fahrenheit
- **humidity** - The relative humidity, between 0 and 1, inclusive
- icon - A machine-readable text summary of this data point
- precipIntensity - The intensity (in inches of liquid water per hour) of precipitation occurring at the given time
- **summary** - A human-readable text summary of this data point
- **temperature** - The air temperature in degrees Fahrenheit
- **windSpeed** - The wind speed in miles per hour

The attributes provided by the weather data are stored from Jan, 2011 to June, 2016 to merge with the bike share data. The features considered in weather data are time, summary, wind speed, temperature, cloud cover and humidity. Summary provided in the dataset is a textual representation of the weather in the area and it is categorized into following:

- **1** - Clear, Breezy, Partly cloudy, hum,id, dry
- **2** - Foggy, overcast, Mostly cloudy, Drizzle, Windy
- **3** - Light snow, Snow, Light rain, Flurries, Rain
- **4** - Heavy rain, Heavy Snow and other inclement weather conditions

The summary is transformed into the above 4 classes and the possible values are a million as the text can be anything related to weather conditions. The combinations of the text is also possible. For example, a text can be Light snow and windy. In such cases the class with highest number will be picked as under fitting is much friendlier than over fitting in the weather because the combination of higher class and lower class will be dominated by the impact of the higher class on the number of bikes practically.

C. Holiday Data

The holiday data[3] is entered manually as the number of data points or samples that are effected by the holiday data is very less and is less than 12 per year. This entering of holiday data is done after merging both the weather and bike share data. The holiday data is added to increase the performance as the bike rental frequency increased almost to peaks during the holidays. Since the weekends are also considered as holidays, the weekends are also marked as holidays in the combined data set.

D. Combining the data

The common feature in both the weather and bike share data sets is the time. Since both the data sets are hourly based a join operation could be performed on both the data sets to merge them. First, the time in the weather data is also split into year, month, day, hour as done in bike share data. Then, the data sets are Left-outer-joined with left side of the join being the bike share data. The entries in the bike share data now have the weather data present. To this, the holiday data is added as another feature (0 or 1).

E. Classifying labels

The count of number of bikes rented per hour will be the final output of the other features in the dataset. Thus, for a given hour and weather conditions, the number of bikes rented have to be counted. With this approach, the error function have to be maintained as the exact prediction of the number of bikes is almost not possible. Thus we followed a classification kind of approach instead of regression by classifying the count of number of bikes into categories. We plotted the bike rental frequency and the number of times those frequencies occurred which is mentioned in Figure 1. Thus, the bike rental frequency is classified into 7 classes for accurate prediction as mentioned in Table II.

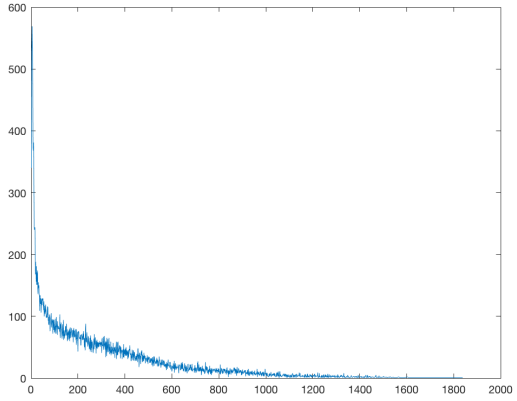

Figure 1: Bike Rental Frequency vs Count

F. Train and Test data

The data is split as training and test data based on the algorithm and the format of data the algorithm is expecting. For example, the input data for Bootstrap Aggregation, SVM and KNN will be in train and test data model, whereas the

Value	Frequency	Class
1	1 - 40	Can't see any bikes
2	40 - 200	Very few bikes
3	200 - 400	Less busy
4	400 - 600	Normal
5	600 - 800	Busy
6	800 - 1100	Very Busy
7	1100 - 2000	All bikes around

TABLE II: Classification of labels

input for Classifier app is a single data file. The total number of data samples that are present are 46936 as the number of hours between January 2011 to June 2016 is 48192 and not all hours will the bikes be rented.

For the models which require a split of data, we used random sampling and split the data as 40000 train samples and 6936 test samples which is equivalent to 85% train data and 15% test data.

IV. Related Work

There is a significant amount of related work in Kaggle but for the subset of the data i.e., the data of the years 2011-2012 and for the different weather parameters. The models that are applied are in vast number and have no correlation between them. The models and kernels that are developed for this data doesn't address the reason behind applying that model and why the accuracy varies with the applied method and the parameters that drastically effect the models in detail.

Our method of approach is different because the entire data set is constructed and the features are considered based on actual parameters that effect the bike rental frequency. As far as our knowledge is concerned, the best work done on this data so far is using Poisson's Regression[5]. Another paper[6] describes the models and the approaches that can be used in predicting the bike rental frequency but in a theoretical way. It compares the performance of Generalized Linear Models with Elastic Net Regularization (GLMNet), Generalized Boosted Models (GBM), Principal Component Regression (PCR), Support Vector Regression (SVR) and a kind of ensemble learning method called Stacking.

V. Methods Used

Considering all the models that are suitable for classification, we used a number of methods that can be trained for classification. The best algorithm is evaluated by getting the accuracy and implemented from scratch in Matlab to observe the importance of the parameters and if anyone of the default assumptions can be overridden for a better performance. The Classifier app of Matlab and Math & Stats model module in Matlab are used to calculate the accuracy of the predictions for all the above mentioned models. The methods varies from basic methods for classification to benchmark models. Following are some of such important methods:

- K-Nearest Neighbors (KNN)
- Support Vector Machines (SVM)
- Support Vector Regression
- Discriminant (Linear and quadratic)
- Classification Tree
- Ensemble Boosted Trees
- Ensemble Bagged Trees (Bootstrap Aggregation)
- Random Forest on Ensemble bagging trees
- Other models which are variants of the above mentioned models

The algorithms that performed better than all other methods are *Ensemble Bagged Trees (Bootstrap Aggregation)* and it's variation *Random Forest on Ensemble bagging trees* with an average accuracy of 80%.

A. K-Nearest Neighbors (KNN)

The K-Nearest neighbors is best suited for classification mostly when dataset is too large, number of features is small and if the distance metric is good. KNN is considered inefficient in some cases as it consumes a lot of space sometimes reaching $O(n^d)$. Variations of the KNN are verified and evaluated including Fine KNN, Medium KNN, Coarse KNN, Cosine KNN, Weighted KNN.

B. Support Vector Machines (SVM)

Support Vector Machine models for classification are best suited for binary classification. The SVM provided by the Math & Statistical module in Matlab doesn't contain multi-class classification. Thus we used libsvm module which supports multi-class classification. In case of SVM models, the hyperparameters and regularizer have to be tuned to find the best match for any data set which is very challenging. The core functionality of SVM depends upon what kernel is being used and it is also challenging to pick the right kernel method. Since it also takes a lot of time to train the data and requires small amount of data samples and large amount of features to achieve better performance, SVM is almost unfit for this data set.

C. Support Vector Regression

Support Vector Machines used for Regression are efficient only when the input data follows a pattern and thus establishes high correlation between the data samples. But SVR is highly insensitive to errors which makes this considerable, but the factors effecting the performance are the same as of SVM.

D. Discriminant (Linear and quadratic)

Linear and Quadratic discriminant analysis are best suitable only if the number of features are very large when compared to number of data samples. LDA requires an assumption of equal variance and co-variance matrices of the classes. The quadratic discriminant analysis comes somewhere in between the linear discriminant analysis and 1-NN (KNN with k=1) and can be considered as a variant of LDA where heterogeneity of classes' co-variance matrices are accepted.

E. Classification Tree

The classification trees or decision trees can be used to predict the class by selecting the most feature and evaluating the score for all the next features and choosing the next best optimal feature to make the split. The classification trees may overfit the data can also get struck in the local minima. Thus, ensemble methods are used which are described in the below sections.

F. Ensemble Boosted Trees

Boosting is a method of decision also called as weak learning where the decision trees are also called as shallow trees since the level of trees is very less and the trees are often called as decision stumps. Boosting reduces bias of a large number of small models with low variance. However, Boosted trees are very sensitive to noise and outliers, and are very prone to overfitting.

G. Ensemble Bagged Trees (Bootstrap Aggregation)

Bootstrap Aggregation or Bagging is a method of reducing the variance for high variance algorithms such as decision trees, Classification and Regression trees(CART). Bagging is a method where multiple models are created from the existing dataset by created sub-sample space. The models are trained and the average/ mode of all the predicted values is considered as the final predicted value. Bagging does not give importance to the individual underlying algorithms overfitting the training data. This is one of the best examples of Ensemble learning method. An ensemble learning method is defined as a learning model which collects and combines predictions from multiple machine learning models and combines the output to make more accurate predictions than the individual models.

The individual decision trees are generally grown deep without pruning and will have high variance and low bias. The only parameters for Bagging decision trees are the number of sample and hence the number of trees. The number of trees are gradually increased run-by-run until accuracy became stable. The number of models can be very large and will not overfit the model.

H. Random Forest on Ensemble bagging trees

In Bagging decision trees, the predictions are highly correlated since the algorithms are greedy. Bagging works better if the predictions are uncorrelated or weakly correlated. To address this problem, Random Forest method is used. In a decision tree or CART, when choosing a split point, the learning algorithm looks at all options and choose one optimal split. The random forest model limits the number of choices to a random subset of features to search. Thus, an extra parameter (m) is introduced here to specify how many features have to be searched at split point among all number of features(p). Generally, the m is defaulted to be square-root of p in case of classification and $p/3$ in case of regression.

Once the data samples are selected in multiple sub-spaces, some samples will be left without taking into any of the sub sample space. These are called Out-Of-Bag samples (OOB) and are used to calculate the validation accuracy. An error function is calculated at each split point to validate the performance of the learning model. For classification, Gini score is the most used error function. Thus, random forests can be used to efficiently construct the learning model even for higher dimensional space and for large data samples.

VI. Results

Evaluating all the models (mentioned in Section V) in Matlab and also in Classifier App gave the individual accuracy of the models as in Table III.

Model	Accuracy (%)
Fine KNN	50.2
Medium KNN	53.3
Coarse KNN	52.8
Cosine KNN	52.5
Weighted KNN	53.7
Linear SVM	48.3
Multi-class SVM	63.0
Support Vector Regression	40.0
Linear Discriminant	47.0
Quadratic Discriminant	46.7
Simple Tree	48.7
Medium Tree	55.4
Complex Tree	61.2
Ensemble Subspace Displacement	46.1
RUSBoosted Trees	55.2
Ensemble Boosted Trees	57.5
Ensemble subspace KNN	60.8
Classification Trees (using fitctree)	77.6
Ensemble TreeBagger	79.8
Ensemble Bagged Trees	80.3

TABLE III: Accuracies of all models used to predict the bike share frequency

A. K-Nearest Neighbors (KNN)

The accuracy is highest at $k = 3$.

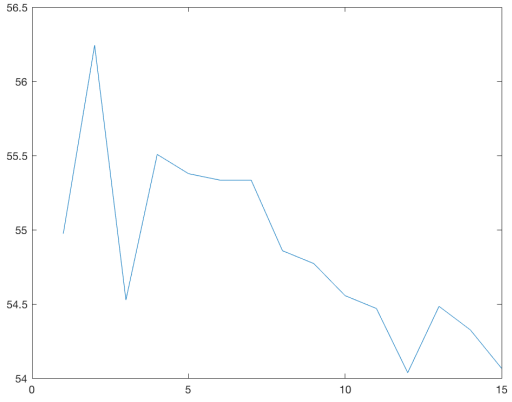

Figure 2: KNN - K vs Accuracy

B. TreeBagger - Random Forest

The TreeBagger is a class provided by the Math and Statistics module in Matlab to train the model using the ensemble bagging and also Random Forest method. The parameters that have to be changed here are the number of predictors that have to be changed for every split point (m) and the number of decision trees that have to grown (N_t). Both the parameters are changed and the best optimal parameters are considered where accuracy is no longer improved.

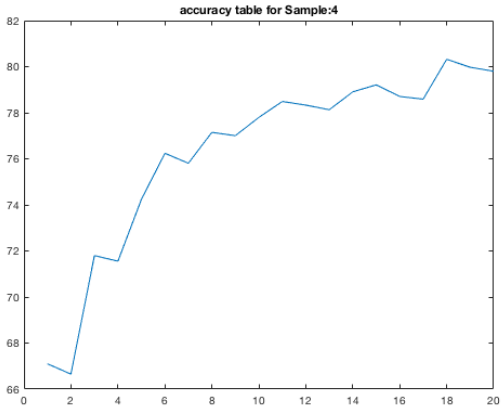

Figure 3: TreeBagger Random Forest - Number of Trees vs Accuracy by fixing the m = 4.

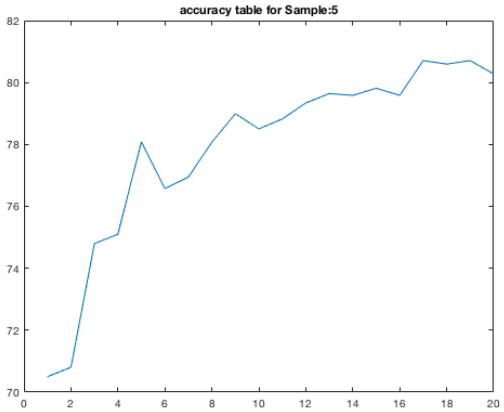

Figure 4: TreeBagger Random Forest - Number of Trees vs Accuracy by fixing the m = 5.

VII. Conclusion

We are fucked up as of now

References

[1] Capital Bike Share Data-set
http://www.capitalbikeshare.com/system-data
[2] Weather Data - ForecastIO - Washington DC
https://darksky.net/dev/docs/response
[3] Holiday Schedule Data
http://dchr.dc.gov/page/holiday-schedules

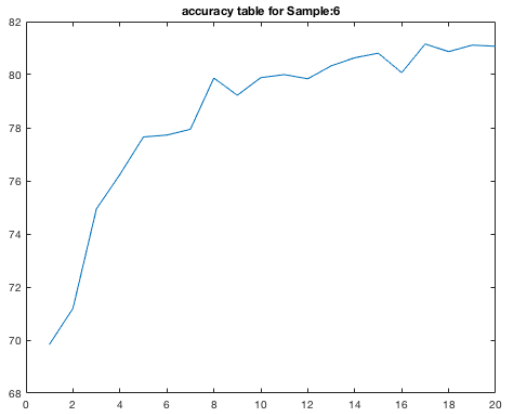

Figure 5: TreeBagger Random Forest - Number of Trees vs Accuracy by fixing the m = 6.

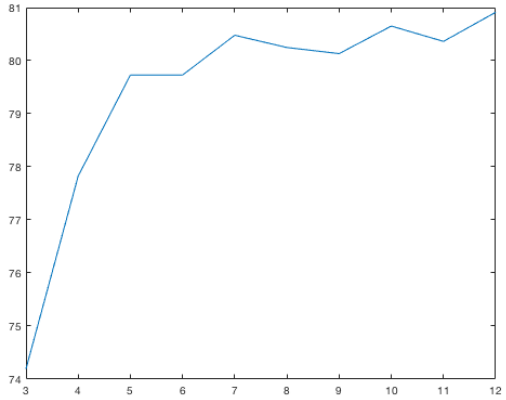

Figure 6: TreeBagger Random Forest - Number of Predictors vs Accuracy by fixing the N_t = 10. The accuracy is optimal at m = 7

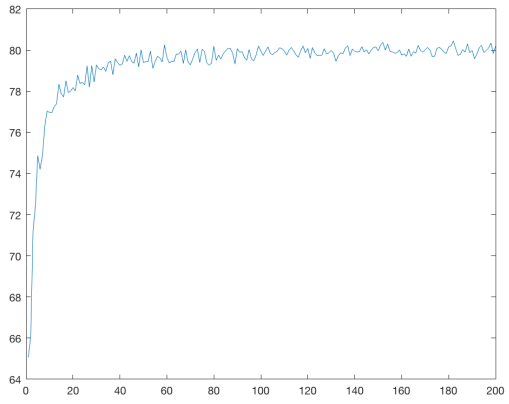

Figure 7: TreeBagger Random Forest - Number of Trees vs Accuracy by fixing the m = 3 (default - square-root of p = *12*).

[4] Kaggle

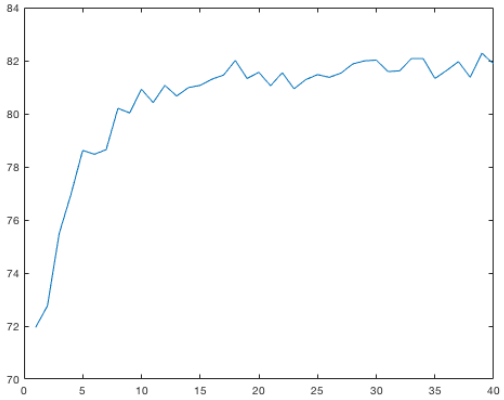

Figure 8: TreeBagger Random Forest - Number of Decision trees vs Accuracy by fixing the $m = 7$. The accuracy is optimal at $N_t = 10$

https://www.kaggle.com/c/bike-sharing-demand

[5] Predictive bikeshare rebalancing
https://github.com/dssg/bikeshare

[6] Jimmy Du, Rolland He, Zhivko Zhechev - *Forecasting Bike Rental Demand*
http://cs229.stanford.edu/proj2014/Jimmy\%20Du,\%20Rolland\%20He,\%20Zhivko\%20Zhechev,\%20Forecasting\%20Bike\%20Rental\%20Demand.pdf