Bike Sharing Demand Prediction In Washington, D.C. Area

Haixiang Fang
Department of Mechanical Engineering
University of Maryland
College Park, USA
hfang@terpmail.umd.edu

***Abstract*—This project is a classic regression problem solved with supervised machine learning technique in python 3.7.3 via Jupyter Notebook. The dataset comprises of data related to bike usage over the year 2011 & 2012 in Capital Bikeshare system in Washington, DC with the corresponding weather and seasonal information. The training set includes hourly rental data for the first 19 days of the month. The goal is to predict the rest of the days for that month. This analysis attempts to generate several machine learning models based on regression to predict rental demand by the hour. The key challenge of this project is to ft models and evaluate accuracy through parameter tuning and feature engineering. Regressions and cross-validations utilize scikit-learning and all figures rely on Seaborn and Matplotlib from python library.**

Keywords—Bike sharing systems, Visualization, Regression

I. Background and Motivation

A bike sharing systems is a service in which bicycles are made available for shared use to individuals on a short term basis among different locations for an affordable price or free. People can rent a bike through membership (mostly regular users) or on demand basis (casual users). This process is controlled by a network of automated kiosk across the city. Few advantages of promoting bike sharing systems in urban areas are reduction of traffic congestion and air pollution, easier way to commute, and a healthier lifestyle. In 2014, University of California - Irvine (UCI) Machine Learning Repository publicized a dataset contains information partially from a two-year historical log from Capital Bikeshare system in Washington D.C. [1]

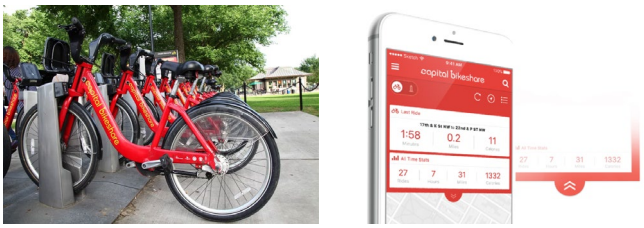

Fig. 1. Capital bikeshare kiosk and mobile application.

Source: Adapted from [2]

It is a fairly simple dataset suitable for implementing some concrete machine learning techniques like Regression and also for some advance ensemble models such as Random Forest. In order to forecast bike rental demand of the Capital Bikeshare program, this project aims to analyze historical usage patterns combined with weather and seasonal data.

II. Preliminaries

A. Data Summary

The complete data contains two sets. The training set is comprised of the first 19 days of each month, while the test set is the 20th to the end of the month. The data is saved in a comma separated file (CSV) with 12 attributes and has already been converted to numeric values. Before starting exploratory analysis, we should review the dataset size, get a glimpse of data by printing a few rows of it and find out what type of variables contribute to the data and their independence.

```
train.shape
```

```
(10886, 12)
```

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
datetime      10886 non-null object
season        10886 non-null int64
holiday       10886 non-null int64
workingday    10886 non-null int64
weather       10886 non-null int64
temp          10886 non-null float64
atemp         10886 non-null float64
humidity      10886 non-null int64
windspeed     10886 non-null float64
casual        10886 non-null int64
registered    10886 non-null int64
count         10886 non-null int64
dtypes: float64(3), int64(8), object(1)
memory usage: 1020.7+ KB
```

```
train.head()
```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered	count
0	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0	3	13	16
1	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0	8	32	40
2	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0	5	27	32

Fig. 2. Variables data type and categorical features.

B. Data Clean

The next step is to examine whether we have any missing values in the data. As the Fig.3. visualizes through the 'missingno' library, the data is consistent and there is no need to correct or delete any errors or corruptions.

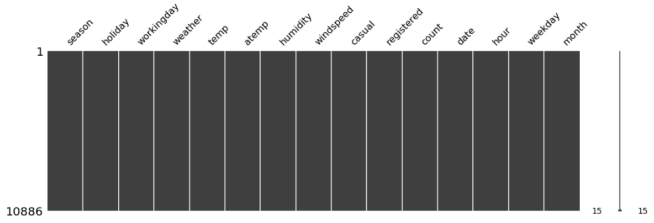

Fig. 3. Missing value analysis.

III. Technical Approach

A. Feature Engineering

- Transform the datatype of "season", "weather", "holiday" and "workingday" from integer to categorical data type.
- Create new columns "date", "hour", "weekday" and "month" extracted from datetime column and then drop the original datetime column.

B. Correlation Analysis

- One general way to understand how a dependent variable is influenced by numerical features is by plotting correlations between them. Therefore, I

calculate the Pearson correlation coefficient between every variable and the target using the ".corr" dataframe method. It gives us an idea of possible relationships within the data.

- Plot histograms as well as perform "barplot" and "pointplot" to depict the relationship between two features and to observe the data distribution.

C. *Model Building*

First, we start by predicting the outcome variable with its mean to set the baseline. Later, Linear Regression, K-Nearest Neighbors (KNN), Decision Tree and Random Forest models will be implemented as benchmark to improve upon the baseline.

After setting the baseline, we train the data with algorithm and checked overfitting through K-fold (k=5) cross-validation (CV) and learning curve method. Then calculate best parameters related to the model.

Finally, we need a model evaluation procedure that allows us to estimate how well a given model is likely to perform on out-of-sample data. One common way to evaluate the regression model is through calculating the cost function of error. In this case, I set up Root Mean Square Logarithmic Error (RMSLE) and R-Squared (R^2) as the metric.

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(Predicted+1)-\log(Actual+1))^3} \quad (1)$$

IV. Results

Examine the Distribution of the Target.

A. *Data visualization*

1) *Variables correlation map*

Fig.4. Haetmap of variables correlations.

Count is highly positively correlated with registered and casual with coefficients of 0.69 and 0.97, respectively. Because count = casual + registered, this positive correlation is shown as expected.

Temp and humidity features have positive and negative correlations with count respectively. In general, extreme low temperature or high humidity is not suitable for cycling.

Furthermore, windspeed may not be a really useful numerical feature since its influence with count is visibly tiny from the correlation value.

Atemp and temp has a strong correlation with each other. During model building, any one of the variables should be dropped since they will exhibit multicollinearity in the data.

2) *Weather data distribution*

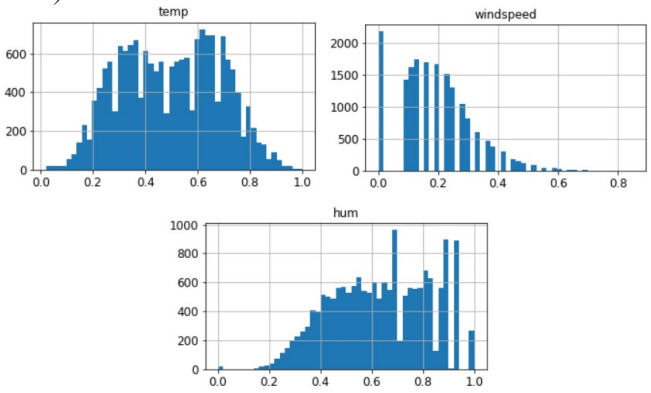

Fig. 5. Weather data distribution.

From the Fig.5., we can obtain a suitable range of weather conditions that customers tend to ride. Temp is ideally following normal distribution. Higher windspeed inhibits demand for bike. While an interesting fact is that higher humidity encourages more bikers to ride.

3) *Rental count regarding to season and weekday*

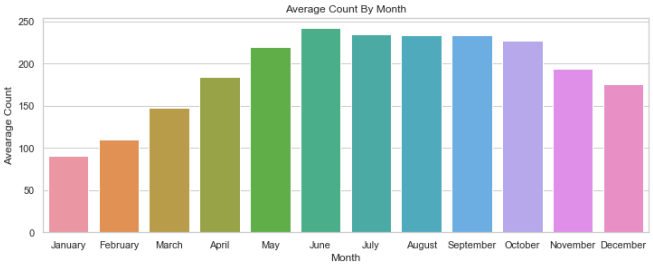

Fig. 6. Average count by month.

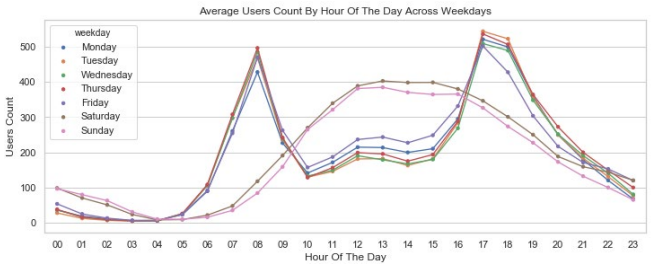

Fig. 7. Average count by hour of the day across weekdays.

It is obvious that people tend to rent a bike during summer and fall season from Fig.6. shown since the weather is more conducive. Therefore June, July, August and September have relatively higher demand for the bicycle.

The Fig.7. examines usage pattern on each day of the week. We clearly see that on weekdays, usage spikes around 8am and 5pm. This likely corresponds to people using the bike-sharing service as commute transportation. On weekends, the usage has a broad peak from 12pm to 4pm. People are likely using these bikes for leisure.

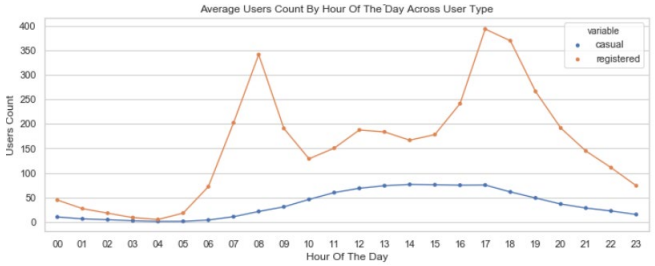

Fig. 8. Average users count across user type.

And Fig.8. shows that registered users are most likely the same group of people who use bike for commute. Since its curve follows a similar pattern as weekdays.

B. Regression model fitting

After dropping a few unnecessary variables and taking log-transform for count, we get the baseline score. After some fine tuning with 5-fold cross-validation with each model, the error is improving through each model and the results show in TABLE I.

TABLE I. ACCURACY EVALUATION

Model	Score		
	RMSLE	***R^2***	***CV***
Baseline	1.591	-0.072	-0.381
Linear Regression	1.326	0.613	0.585
KNN	0.919	0.642	0.635
Decision Tree	0.570	0.751	0.821
Random Forest	0.510	0.844	0.849

Because our features are mostly numeric and continuous, and we have many examples relative to features, a KNN model works well here. Due to its ability to model highly complicated boundaries, Decision Tree performs better than KNN regression. Random Forest combines the results of many decision trees to reduce variance. Therefore, we finally get the lowest error through this model. However, overfitting problems do exist in last two models since CV is higher than R^2 score.

We can keep tuning the hyperparameters to see if we can improve upon the Random Forest results through Grid Search technique. From the Fig.9. the error seems to stop getting lower after around n=20. In other words, 20 decision trees are enough to get most of the benefits from random forest in this case.

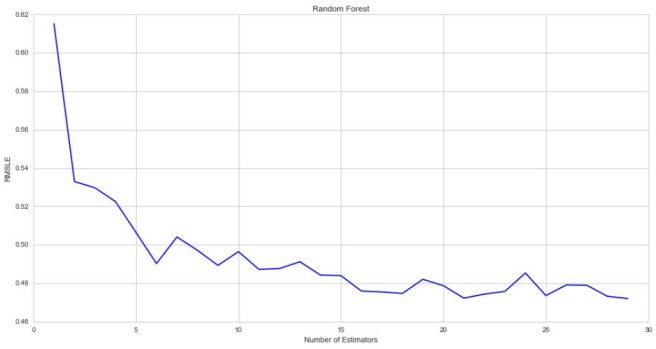

Fig. 9. Parameter tuning for Random Forest.

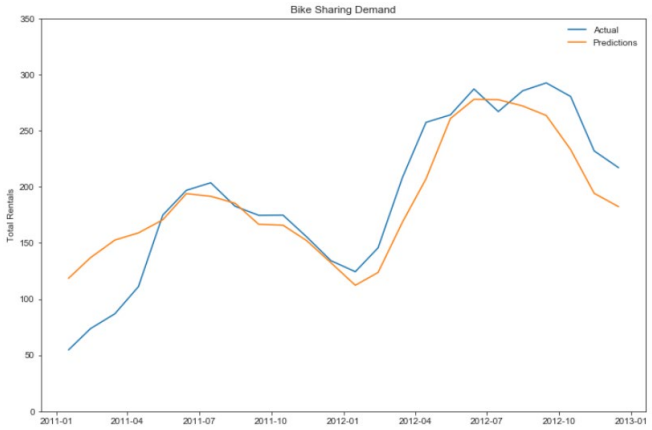

Fig. 10. Bike sahring demand prediction.

We see from the error plot Fig.10. that our model tends to underestimate both at the beginning and towards the end of the data set. This happens because of the nature of cross-validation and the lack of autocorrelation modeling. In cross-validation, we use part of the train data to predict the rest of the train data. If we are predicting the backend of the dataset, we will use the front 80% in this case to predict the back 20%. Because our model does not have trend information, if the trend is increasing, it won't be able to adjust for it. We could incorporate findings from time series analysis to address this problem.

V. CONCLUSION

Using feature engineering and different regression models, we approached the problem with Random Forest Regression to accurately predict the bike sharing demand.

In the future, we may apply more advanced models such as Gradient Boost and XGBoost and tune parameter through grid search to reduce the error. We could also predict casual and registered user count separately since they follow completely different patterns.

REFERENCES

[1] http://archive.ics.uci.edu/ml/datasets

[2] https://dc.curbed.com/2019/4/15/18311279/capital-bikeshare-dc-transportation-electric-bikes

[3] Fanaee-T, Hadi, and Gama, Joao, "Event labeling combining ensemble detectors and background knowledge", Progress in Artificial Intelligence (2013): pp.1-15, Springer Berlin Heidelberg doi:10.1007/s13748-013-0040-3.