Bike Rental Prediction

Submitted by:

Aditya Kapoor

Introduction

Problem Statement

The objective of this Case is to Predication of bike rental count on daily based on the environmental and seasonal settings.

Data

The details of data attributes in the dataset are as follows

1. nstant: Record index
2. dteday: Date
3. season: Season (1:springer, 2:summer, 3:fall, 4:winter)
4. yr: Year (0: 2011, 1:2012)
5. mnth: Month (1 to 12)
6. hr: Hour (0 to 23)
7. holiday: weather day is holiday or not (extracted fromHoliday Schedule)
8. weekday: Day of the week
9. workingday: If day is neither weekend nor holiday is 1, otherwise is 0.
10. weathersit: (extracted fromFreemeteo) 1: Clear, Few clouds, Partly cloudy, Partly cloudy 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
11. temp: Normalized temperature in Celsius. The values are derived via (t-t_min)/(t_max-t_min),t_min=-8, t_max=+39 (only in hourly scale)
12. atemp: Normalized feeling temperature in Celsius. The values are derived via (t-t_min)/(t_maxt_min), t_min=-16, t_max=+50 (only in hourly scale)
13. hum: Normalized humidity. The values are divided to 100 (max)
14. windspeed: Normalized wind speed. The values are divided to 67 (max)
15. casual: count of casual users
16. registered: count of registered users
17. cnt: count of total rental bikes including both casual and registered

Insights

Data Exploration

	instant	dteday	season	yr	mnth	holiday	weekday	workingday
0	1	2011-01-01	1	0	1	0	6	0
1	2	2011-01-02	1	0	1	0	0	0
2	3	2011-01-03	1	0	1	0	1	1
3	4	2011-01-04	1	0	1	0	2	1
4	5	2011-01-05	1	0	1	0	3	1

	weathersit	temp	atemp	hum	windspeed	casual	registered
0	2	0.344167	0.363625	0.805833	0.160446	331	654
1	2	0.363478	0.353739	0.696087	0.248539	131	670
2	1	0.196364	0.189405	0.437273	0.248309	120	1229
3	1	0.200000	0.212122	0.590435	0.160296	108	1454
4	1	0.226957	0.229270	0.436957	0.186900	82	1518

	cnt
0	985
1	801
2	1349
3	1562
4	1600

We see that the dataset is already pretty good and doesn't require a lot of pre-processing. In addition to the existing columns, we would be needing columns like month end. We will then try to find useful insights by these columns to see if they have an impact.

Feature Engineering

We extract features like 'day' and 'Month_End' to derive meaningful insights from the data.

	instant	dteday	season	yr	mnth	holiday	weekday	workingday
0	1	2011-01-01	1	0	1	0	6	0
1	2	2011-01-02	1	0	1	0	0	0
2	3	2011-01-03	1	0	1	0	1	1
3	4	2011-01-04	1	0	1	0	2	1
4	5	2011-01-05	1	0	1	0	3	1

	weathersit	temp	atemp	hum	windspeed	casual	registered
0	2	0.344167	0.363625	0.805833	0.160446	331	654
1	2	0.363478	0.353739	0.696087	0.248539	131	670
2	1	0.196364	0.189405	0.437273	0.248309	120	1229
3	1	0.200000	0.212122	0.590435	0.160296	108	1454
4	1	0.226957	0.229270	0.436957	0.186900	82	1518

	cnt	Day	Month_End
0	985	1	0.0
1	801	2	0.0
2	1349	3	0.0
3	1562	4	0.0
4	1600	5	0.0

Insights

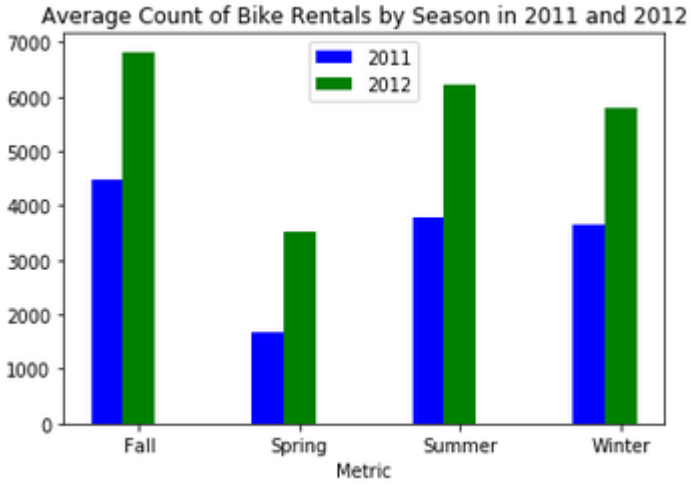
We would like to look at the summary of the data grouped at the following levels.

1. Year Level:

yr	Total Rentals	Average Rentals per Day	Median Rentals per Day
2011	1243103	3405.761644	3740
2012	2049576	5599.934426	5927

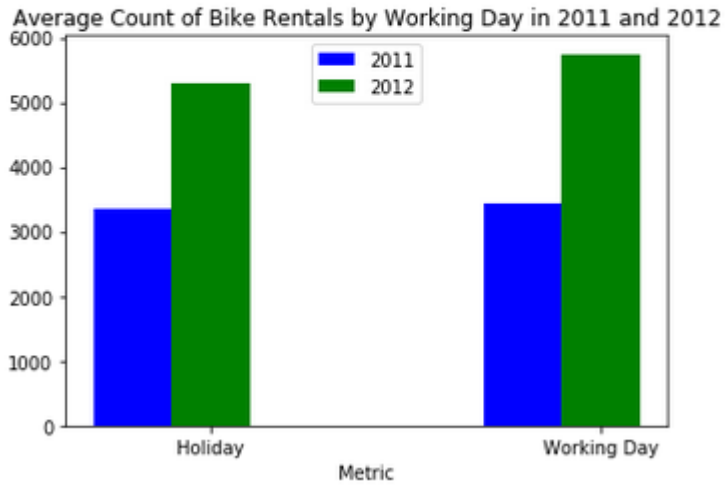
   Clearly, 2012 has more rentals than 2011.

2. By now, it is clear that year is an important parameter to consider with other parameters. First, we look at data at Year, Season level



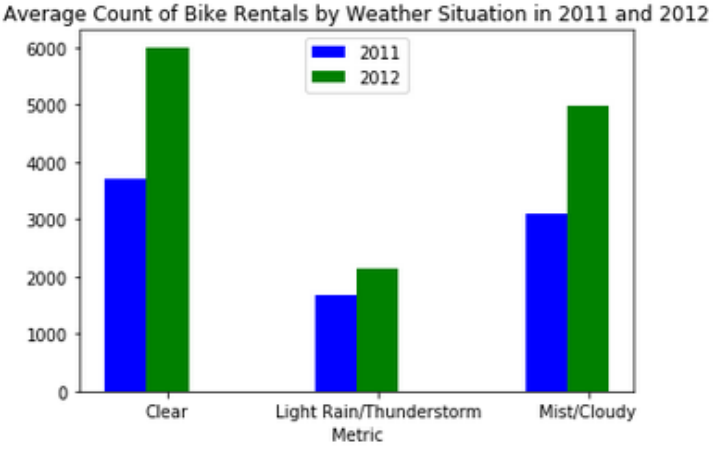
3. Metrics by year and working day.
   Please note that in the figure, 0 represents a non-working day while 1 represents a working day.

workingday	0	1
yr		
2011	3363.817391	3425.056
2012	5288.189655	5744.584



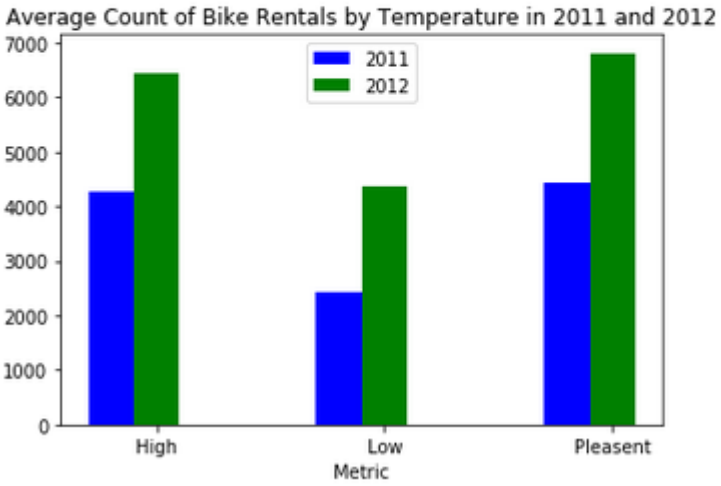
4. Metrics by Year and Weather Situation

weathersit	Clear, Few Clouds	Light Rain, Snow, Thunderstorm, Clouds	Mist, Cloudy, less clouds
yr			
2011	3694.986726	1674.133333	3088.096774
2012	6003.734177	2126.166667	4991.333333



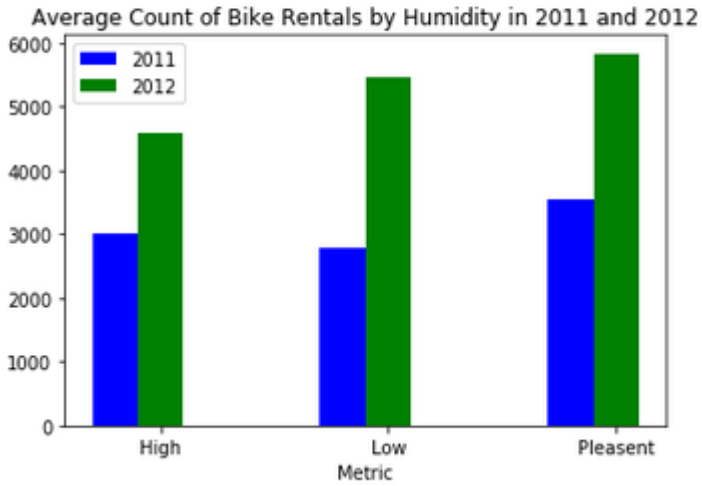
5. Metrics by temperature

temp_bucket	High	Low	Pleasent
yr			
2011	4281.137931	2426.913514	4436.894040
2012	6430.906250	4350.068571	6808.333333



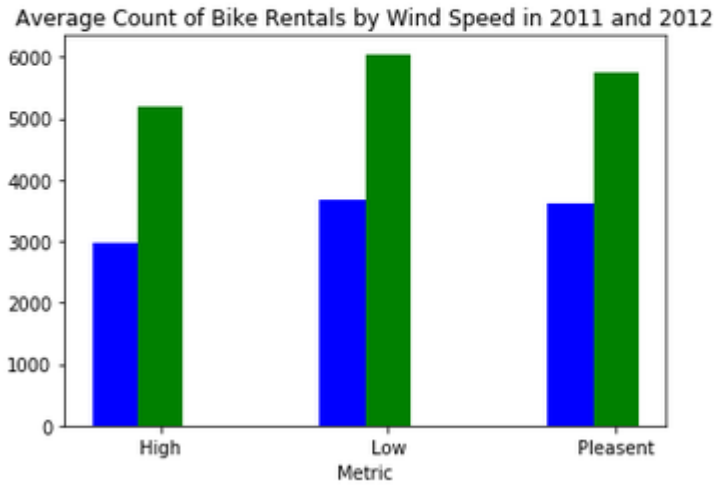
6. Metrics by humidity

hum_bucket	High	Low	Pleasent
yr			
2011	3016.870588	2781.583333	3557.052239
2012	4576.523810	5448.352941	5834.381119



7. Metrics by wind speed

windspeed_bucket	High	Low	Pleasent
yr			
2011	2966.669565	3668.833333	3603.004310
2012	5193.471154	6045.000000	5746.465863



Summary

1. We see that 2012 has a higher count of rentals in all seasons.
2. Among seasons, highest count of rentals is in Fall, while the lowest is in Spring (Season 1)
3. We do not see a major difference between rentals on Working Days/Holidays. However, rentals on Working Days are slightly higher.
4. As per expectations, average count of bike rental is highest when the weather is clear and lowest during rains and thunderstorms.
5. We see that more people prefer to rent a bike when the temperature is high than when it is low.
6. There are more number of rentals when humidity is moderate/low than when it is high
7. Count of Rentals is highest when the wind speed is low

Feature Selection

We start with correlation analysis. Look at the following table.

	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	cnt	Month_End
season	1.000000	-0.001844	0.831440	-0.010537	-0.003080	0.012485	0.019211	0.334315	0.342876	0.205445	-0.229046	0.406100	0.006121
yr	-0.001844	1.000000	-0.001792	0.007954	-0.005461	-0.002013	-0.048727	0.047604	0.046106	-0.110651	-0.011817	0.566710	0.002637
mnth	0.831440	-0.001792	1.000000	0.019191	0.009509	-0.005901	0.043528	0.220205	0.227459	0.222204	-0.207502	0.279977	0.011085
holiday	-0.010537	0.007954	0.019191	1.000000	-0.101960	-0.253023	-0.034627	-0.028556	-0.032507	-0.015937	0.006292	-0.068348	-0.009072
weekday	-0.003080	-0.005461	0.009509	-0.101960	1.000000	0.035790	0.031087	-0.000170	-0.007537	-0.052232	0.014282	0.067443	-0.004303
workingday	0.012485	-0.002013	-0.005901	-0.253023	0.035790	1.000000	0.061200	0.052660	0.052182	0.024327	-0.018796	0.061156	0.007061
weathersit	0.019211	-0.048727	0.043528	-0.034627	0.031087	0.061200	1.000000	-0.120602	-0.121583	0.591045	0.039511	-0.297391	-0.026300
temp	0.334315	0.047604	0.220205	-0.028556	-0.000170	0.052660	-0.120602	1.000000	0.991702	0.126963	-0.157944	0.627494	0.019858
atemp	0.342876	0.046106	0.227459	-0.032507	-0.007537	0.052182	-0.121583	0.991702	1.000000	0.139988	-0.183643	0.631066	0.013039
hum	0.205445	-0.110651	0.222204	-0.015937	-0.052232	0.024327	0.591045	0.126963	0.139988	1.000000	-0.248489	-0.100659	-0.005944
windspeed	-0.229046	-0.011817	-0.207502	0.006292	0.014282	-0.018796	0.039511	-0.157944	-0.183643	-0.248489	1.000000	-0.234545	0.041908
cnt	0.406100	0.566710	0.279977	-0.068348	0.067443	0.061156	-0.297391	0.627494	0.631066	-0.100659	-0.234545	1.000000	-0.033359
Month_End	0.006121	0.002637	0.011085	-0.009072	-0.004303	0.007061	-0.026300	0.019858	0.013039	-0.005944	0.041908	-0.033359	1.000000

We see that:

1. Season is highly correlated with month
2. Year is highly correlated with instant
3. Temp is highly correlated with atemp

Also, columns like day, dteday and holiday are adding little value to our dataset. Hence, we drop them. Registered and Casual imply the target variable. Hence, we remove them also.

Now that our dataset has been prepared, we will split it into train and test in the ratio of 4:1 rows. After splitting the data, we proceed for model building.

Model Building

We use GridSearchCV for developing and tuning hyper parameters of 4 algorithms provided by Scikit Learn library.

1. LinearRegression()
2. Ridge()
3. DecisionTreeClassifier()
4. RandomForestClassifier()

The idea is to test these models with different hyper parameters using GridSearchCV with CV=5 and selecting the one which gives the least mean RMSE.

After running training all the algorithms, we exported the results. You may find the results as part of the same zipped folder. Here is a snippet of the same.

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	\
13	0.984304	0.115948	0.037500	0.012497	
12	0.328141	0.009883	0.012501	0.006250	
16	0.821917	0.096631	0.034375	0.006251	

	params	split0_test_score	\
13	{'min_samples_leaf': 2, 'min_samples_split': 2...	-730.578968	
12	{'min_samples_leaf': 2, 'min_samples_split': 2...	-733.862016	
16	{'min_samples_leaf': 2, 'min_samples_split': 1...	-748.578988	

	split1_test_score	split2_test_score	split3_test_score	\
13	-654.617544	-712.972370	-638.728783	
12	-655.700387	-727.597464	-640.086592	
16	-659.992062	-740.930721	-644.203005	

	split4_test_score	mean_test_score	std_test_score	rank_test_score	\
13	-706.369143	-688.653362	35.535672	1	
12	-709.148507	-693.278993	38.257454	2	
16	-731.233537	-704.987663	43.818562	3	

	param_alpha	param_min_samples_leaf	param_min_samples_split	\
13	NaN	2	2	
12	NaN	2	2	
16	NaN	2	10	

	param_n_estimators
13	250
12	100
16	250

Summary

1. We have trained LinearRegression, Ridge, Decision Tree Regressor and Random Forest Regressor algorithm with different combinations of hyper parameters.
2. We have used RMSE (Root Mean Squared Error) because we want to penalise high deviations more.
3. The Random Forest Classifier with 2 min_samples_leaf, 2 min_samples_split and 250 estimators has the best performance. However, we see that the Classifier with the same min_samples_leaf and min_samples_split but less estimators performs just as good. Hence, in order to avert avoidable computational cost, we will go with the model with less trees

Model Building

Now that we know the best model and corresponding hyperparameters, we train the model using them and find RMSE, MAE and MAPE on test set. Note that we used RMSE as a metric for comparing models because RMSE score penalises bigger deviations more, and since this is a regression problem with number of rental prediction, we are better to do with an algorithm that makes many small mistakes rather than some bigger ones. Hence, it is useful to choose a metric that penalises big deviations more. This is the rationale behind using RMSE for comparing models. However, after model development we would like to understand more about the nature of these deviations, which is why we have used MAPE and MAE.

Conclusion

We have trained and built a Random Forest Model and evaluated its performance on testing set. The RMSE score of this model is 636, MAE Score is 438 and MAPE is approximately 14%.

The hyper parameters used for the regressor are:

min_sample_leaf: 2

min_sample_split: 2

n_estimators: 100

random_state: 1024

The notebook containing the code to do all of the above may be found in the same compressed folder along with R Script, README file and hyper parameter tuning report.