BIKE SHARING DEMAND PTEDITION

CHOUYIN ZHANG

Abstract. In this data mining task, I use the dataset from Kaggle. According the problem background given, I apply EDA, data preprocessing to the dataset, and finally apply the best model to test data, successfully finish this task.

Contents

Date: 2022-11-11.
2020 *Mathematics Subject Classification.* Artificial Intelligence.
Key words and phrases. Machine Learning, Data Mining, ...

1. Problem Introduction

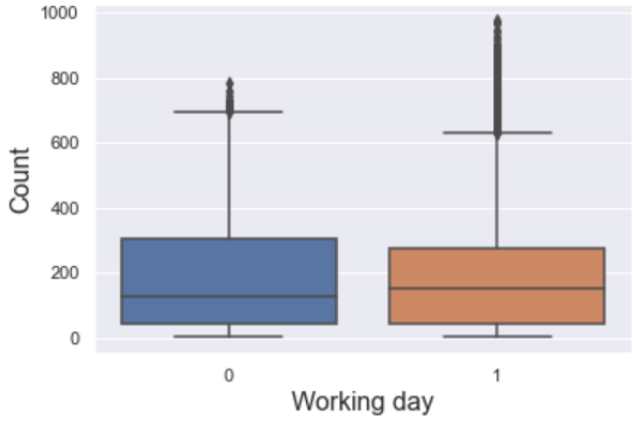
Bike sharing systems are a means of renting bicycles where the process of obtaining membership, rental, and bike return is automated via a network of kiosk locations throughout a city. Using these systems, people are able rent a bike from a one location and return it to a different place on an as-needed basis. Currently, there are over 500 bike-sharing programs around the world.

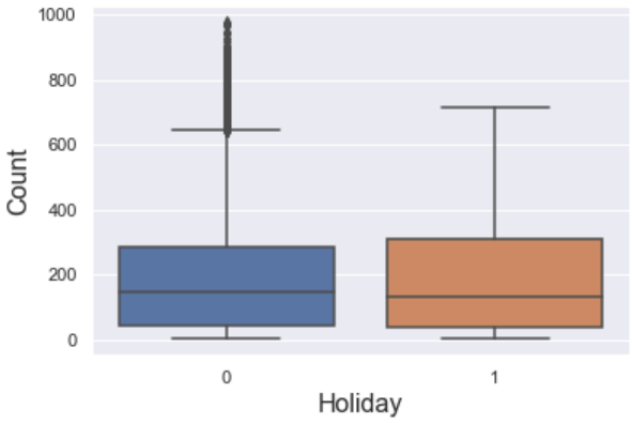
The data generated by these systems makes them attractive for researchers because the duration of travel, departure location, arrival location, and time elapsed is explicitly recorded. Bike sharing systems therefore function as a sensor network, which can be used for studying mobility in a city. In this competition, participants are asked to combine historical usage patterns with weather data in order to forecast bike rental demand in the Capital Bikeshare program in Washington, D.C.

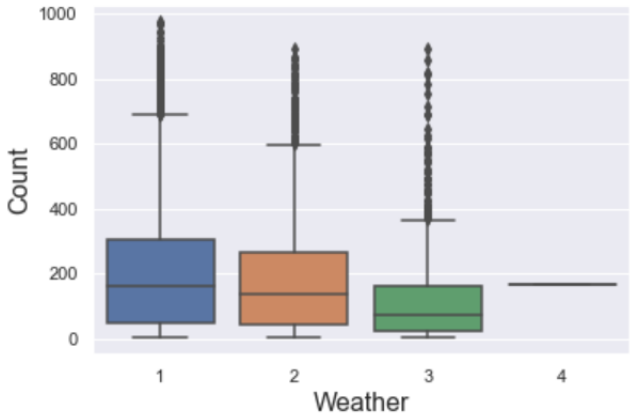
2. Exploratory Data Analysis

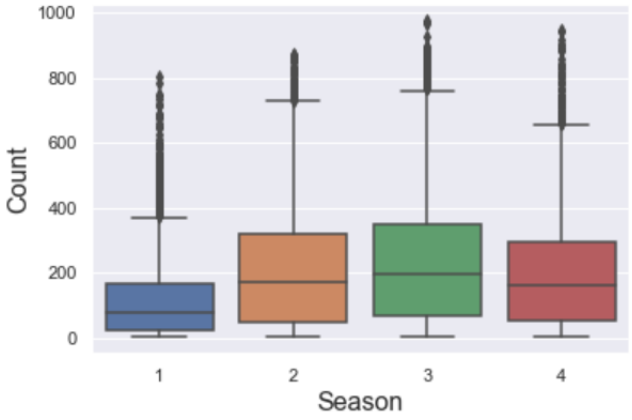
First we can look at the distribution of count, the variable which gives the total number of rented bikes for a given training sample. The distribution is heavily skewed right thus is not well approximated by a normal distribution.

We can also visualize the disribution of the total count for different categorical variables, such as season, weather, month, and working day. We Use box plots,

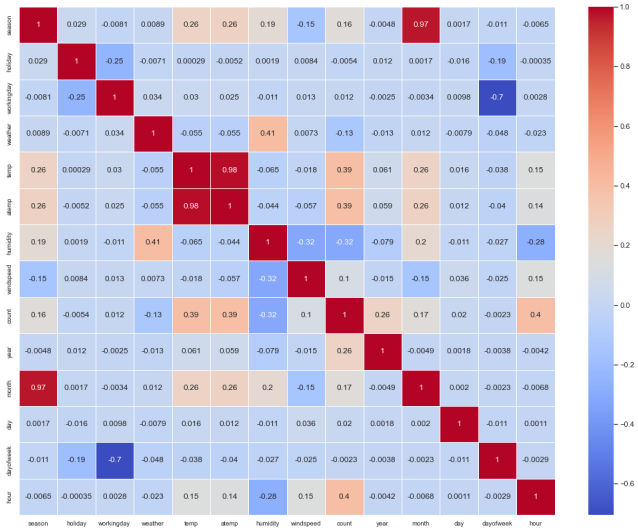








We can also plot a correlation heatmap showing the 2D correlation matrix for the features in our dataset.

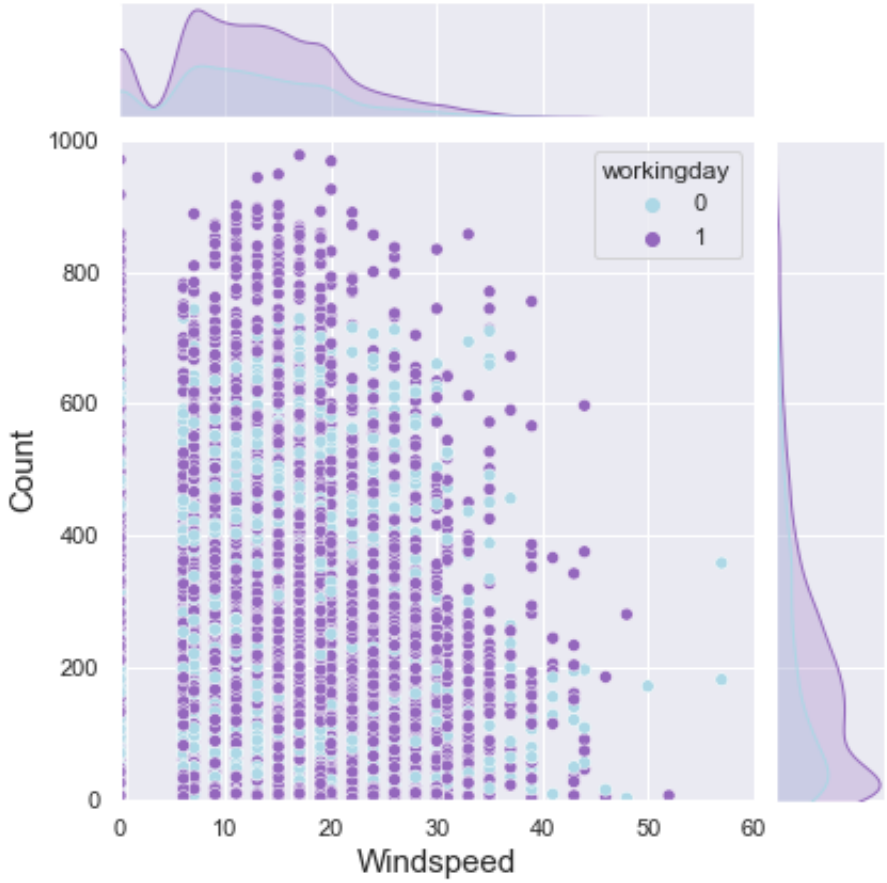


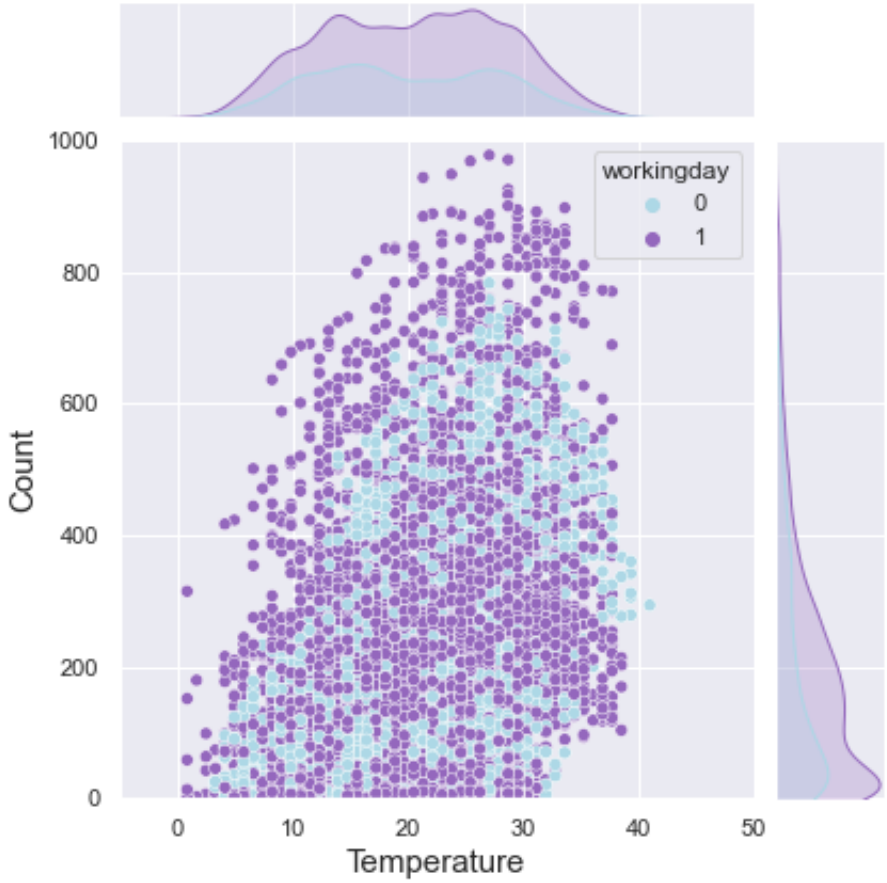
Several features have a positive correlation with the total count, such as the hour (+0.4) , temperature (+0.39) , and "atemp" (+0.39) , the feels-like temperature. However temperature and atemp have a correlation of almost 1 , so it may be better to simply remove the atemp data. There is also a clear negative correlation (−0.32) between total count and humidity.

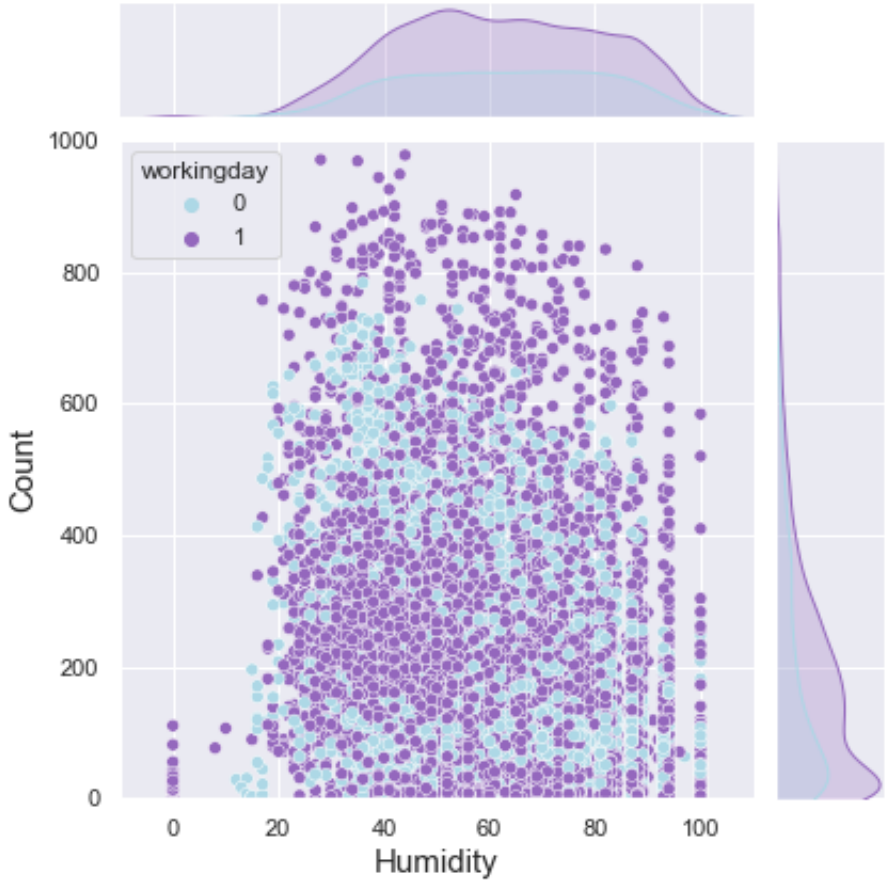
There is also a positive correlation (+0.26) between the year and bike usage. Our training set contains data over two years, 2011 and 2012.

Overall, there is little correlation between total count and the workingday variable.

We can visualize how daily usage varies with the continuous weather variables (temperature, atemp, wind speed, and humidity) using a jointplot.


workingday
0
1
Count
Windspeed
0
200
400
600
800
1000
0
10
20
30
40
50
60


workingday
0
1
Count
Temperature
0
200
400
600
800
1000
0
10
20
30
40
50



Here we can see visually the positive correlation between total count and the temperature, as well as the negative correlation between total count and humidity. The distributions for workdays and non-workdays are very similar as indicated by the accompanying kde plots.

3. Data preprocessing

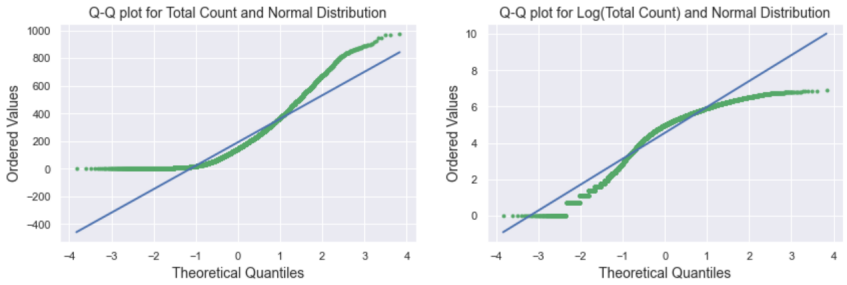
First, note that the total count (the variable we will aim to predict) is the sum of the casual count and the registered count. As such, the test set does not contain data for the casual count or the registered count. We should thus drop casual and registered from our dataframe. Let us also drop atemp since it has almost perfect correlation with temp.

Secondly, note that the testing and training data (with the split made by Kaggle for the purposes of the competition) is determined by the day of the month. The combined data sets represent records ranging from January 2011 to December 2012, with data collected on or before the Day 19 of each month assigned to the training set, and data collected from Day 20 onwards assigned to the testing set by Kaggle. Consequently, we drop all day of the month data from the training set before training our models.

The data set contains a mixture of continuous variables (temperature and windspeed) and categorical variables (such as season, weather type, month, and day of the week). However, the categorical data is expressed as an arbitrary numerical value. Therefore we can use dummy coding, replacing the categorical data with binary dummy variables using the pandas function pd.get_dummies. After this, we also drop one redundant category known as the reference category. For example, only six binary variables are necessary to fully specify the day of the week.

As noted previously, the natural logarithm of count is more normally distributed. Let us replace the target variable count with y=ln(count+1) . This ensures total counts of 0 are mapped to 0 and not $-\infty$. The training data X consists of all other data in our dataframe (now 52 features).

Taking the natural log of the count data gives a distribution slightly more normally distributed, as shown above, and also in the quantile-quantile (Q-Q) plots below. Here the quantiles of the data (i.e. the values in order) are plotted against the same quantilies of a theoretical normal distribution.



We split the labelled training set into two sets: one for training our models, and a validation set for determining the best set of hyperparameters.

Finally, we should define a function predict that will report prediction scores for a given model.

4. Train and Apply models

Firstly, we should train our models and select the best one by comparing their RMSLE score.

We apply train data and validation data to the chosen model, such as LinearRegression, Ridge, Lasso, RandomForestRegressor, GradientBoostingRegressor, XGBRegressor, LGBMRegressor.

The scores of each model are as below.

Model	RMSLE For Each Model
LinearRegression	0.136
Ridge	0.137
Lasso	0.148
RandomForestRegressor	0.084
GradientBoostingRegressor	0.080
XGBRegressor	0.085
LGBMRegressor	0.078

We select LGBMRegressor as our best model because of its lowest RMSLE value. We apply the model above to test data and predict the bike sharing demand.

	Unnamed: 0	datetime	count
0	0	2011-01-20 00:00:00	10.0
1	1	2011-01-20 01:00:00	4.0
2	2	2011-01-20 02:00:00	3.0
3	3	2011-01-20 03:00:00	3.0
4	4	2011-01-20 04:00:00	2.0

5. Conclusions

We serve the data set and submit our code and tex file to github.

References

List of Todos

(A. 1) Hunan University, Changsha, China
Email address, A. 1: 13907363568@139.com