**PYTHON PROJECT**

**Bike-Sharing Demand Analysis**

DESCRIPTION

**Objective**:  Use data to understand what factors affect the number of bike trips. Make a predictive model to predict the number of trips in a particular hour slot, depending on the environmental conditions.

**Problem Statement**:

Lyft, Inc. is a transportation network company based in San Francisco, California and operating in 640 cities in the United States and 9 cities in Canada. It develops, markets, and operates the Lyft mobile app, offering car rides, scooters, and a bicycle-sharing system. It is the second largest rideshare company in the world, second to only Uber.

Lyft’s bike-sharing service is also among the largest in the USA. Being able to anticipate demand is extremely important for planning of bicycles, stations, and the personnel required to maintain these. This demand is sensitive to a lot of factors like season, humidity, rain, weekdays, holidays, and more. To enable this planning, Lyft needs to rightly predict the demand according to these factors.

**Domain**: General

**Analysis to be done**: Rightly predict the bike demand

**Content**: Dataset: Lyft bike-sharing data (hour.csv)

**Fields in the data**:

- instant: record index

- dteday: date

- season: season (1:spring, 2:summer, 3:fall, 4:winter)

- yr: year (0: 2011, 1: 2012)

- mnth: month (1 to 12)

- hr: hour (0 to 23)

- holiday : whether the day is a holiday or not

- weekday : day of the week

- workingday : if the day is neither weekend nor a holiday is 1, otherwise is 0

- weathersit :

- 1: Clear, Few clouds, Partly cloudy

- 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

- 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds

- 4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog

- temp : normalized temperature in Celsius; the values are divided to 41 (max)

- atemp: normalized temperature felt in Celsius; the values are divided to 50 (max)

- hum: normalized humidity; the values are divided to 100 (max)

- windspeed: normalized wind speed; the values are divided to 67 (max)

- casual: count of casual users

- registered: count of registered users

- cnt: count of total rental bikes including both casual and registered

**Steps to perform**:

As the first step, look at the null values in the file. A sanity check, to ensure that you have clean records and the data is good to go ahead, is very important. Then, you’ll do univariate and bivariate analyses to identify the patterns in the data and the nature of the individual features. This is a very important step as this helps to not only identify features which could be interesting for the predictive model later, but also helps  understand what’s going on in the data. The EDA will help identify the need to apply transformations on the features before building the model. Finally, you will make a predictive model using linear regression.

1. Load the data file.
2. Check for null values in the data and drop records with NAs.
3. Sanity checks:
   1. Check if registered + casual = cnt for all the records. If not, the row is junk and should be dropped.
   2. Month values should be 1-12 only
   3. Hour values should be 0-23
4. The variables ‘casual’ and ‘registered’ are redundant and need to be dropped. ‘Instant’ is the index and needs to be dropped too. The date column dteday will not be used in the model building, and therefore needs to be dropped. Create a new dataframe named **inp1**.

5. Univariate analysis:

* Describe the numerical fields in the dataset using pandas describe method.
* Make density plot for temp. This would give a sense of the centrality and the spread of the distribution.
* Boxplot for atemp
  + Are there any outliers?
* Histogram for hum
  + Do you detect any abnormally high values?
* Density plot for windspeed
* Box and density plot for cnt – this is the variable of interest
  + Do you see any outliers in the boxplot?
  + Does the density plot provide a similar insight?

6. Outlier treatment:

1. Cnt looks like some hours have rather high values. You’ll need to treat these outliers so that they don’t skew the analysis and the model.
   1. Find out the following percentiles: 10, 25, 50, 75, 90, 95, 99
   2. Decide the cutoff percentile and drop records with values higher than the cutoff. Name the new dataframe as **inp2**.

7. Bivariate analysis

1. Make boxplot for cnt vs. hour
   1. What kind of pattern do you see?
2. Make boxplot for cnt vs. weekday
   1. Is there any difference in the rides by days of the week?
3. Make boxplot for cnt vs. month
   1. Look at the median values. Any month(s) that stand out?
4. Make boxplot for cnt vs. season
   1. Which season has the highest rides in general? Expected?
5. Make a bar plot with the median value of cnt for each hr
   1. Does this paint a different picture from the box plot?
6. Make a correlation matrix for variables atemp, temp, hum, and windspeed
   1. Which variables have the highest correlation?

8. Data preprocessing

A few key considerations for the preprocessing:

There are plenty of categorical features. Since these categorical features can’t be used in the predictive model, you need to convert to a suitable numerical representation. Instead of creating dozens of new dummy variables, try to club levels of categorical features wherever possible. For a feature with high number of categorical levels, you can club the values that are very similar in value for the target variable.

1. Treating mnth column
   1. For values 5,6,7,8,9,10, replace with a single value 5. This is because these have very similar values for cnt.
   2. Get dummies for the updated 6 mnth values
2. Treating hr column
   1. Create new mapping: 0-5: 0, 11-15: 11; other values are untouched. Again, the bucketing is done in a way that hr values with similar levels of cnt are treated the same.
3. Get dummy columns for season, weathersit, weekday, mnth, and hr. You needn’t club these further as the levels seem to have different values for the median cnt, when seen from the box plots.

9. Train test split: Apply 70-30 split.

- call the new dataframes df\_train and df\_test

10. Separate X and Y for df\_train and df\_test. For example, you should have X\_train, y\_train from df\_train. y\_train should be the cnt column from inp3 and X\_train should be all other columns.

10 . Model building

* Use linear regression as the technique
* Report the R2 on the train set

11. Make predictions on test set and report R2.