[A sridhar]

**Project**

**Bike**

**Renting**

in

Python

# Contents

# [Introduction](file:///C:\Users\Sridhar\Desktop\New%20folder\bike%20renting%20project%20python.rtf#_bookmark0)

1)[Problem Statement](file:///C:\Users\Sridhar\Desktop\New%20folder\bike%20renting%20project%20python.rtf#_bookmark1)

2)[Data](file:///C:\Users\Sridhar\Desktop\New%20folder\bike%20renting%20project%20python.rtf#_bookmark2)

[**Methodology**](file:///C:\Users\Sridhar\Desktop\New%20folder\bike%20renting%20project%20python.rtf#_bookmark3)

**1) Data Pre-processing**

1)Variable identification

**2)Missing Value Analysis**

**Visualization**

1)Univariate Analysis

2)Bivariate Analysis

**3)Outlier Analysis**

**4)Feature Selection**

1)Correlation, Anova

**5)Feature Scaling**

**Modelling**

**Model Selection**

1)Decision Tree

2)Random Forest

3)Linear Regression

**Visualizing models**

**Conclusion**

1)Model Evaluation

2)Mean Absolute Percentage Error(MAPE) Mean Squared Error(MSE)

3)Model Selection

**Python- Code**

***Introduction***

***Problem Statement:*** The objective of this Project is to Predict the count of the bikes to be rented on daily basis. This count will take environmental and seasonal settings from the historical data in account while forecasting the daily demand. We would be building a model that can successfully predict the count of rentals on relevant factors.

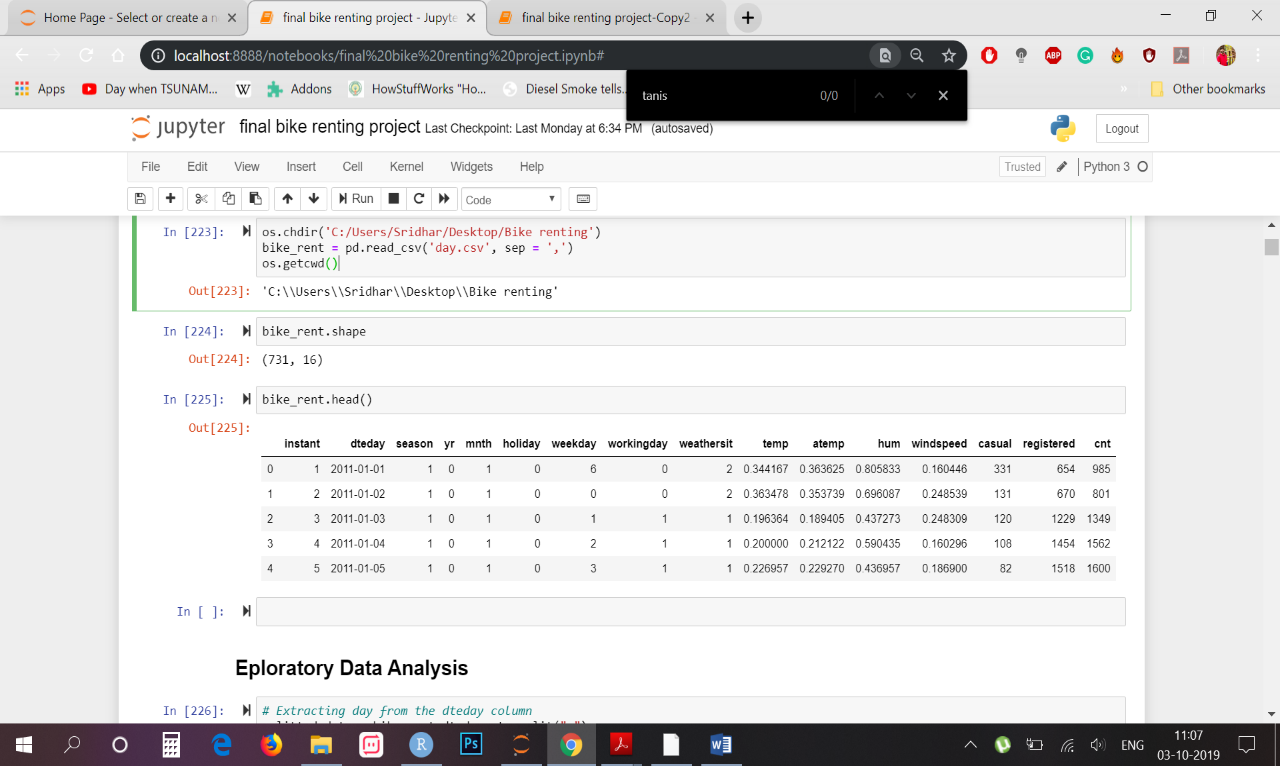
***Data:*** As the dataset given has dependent and independent values, it will come under supervise Machine learning. Our task is to build Regression models which will help us predicting the count of bikes which will get rented depending on the factors provided. Given below is a sample of the data set that we are using for our prediction.

|  |  |
| --- | --- |
| **Variable** | **Explanation** |
| instant | Daily customer index |
| dteday | Date index for both the years |
| season | Season (1:springer, 2:summer, 3:fall, 4:winter) |
| yr | Year (0: 2011, 1:2012) |
| mnth | Month (1 to 12) |
| holiday | weather day is holiday or not (extracted fromHoliday Schedule) |
| weekday | Day of the week |
| workingday | If day is neither weekend nor holiday is 1, otherwise is 0. |
| 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 |
| 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) |
| 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) |
| hum (Humidity) | 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 | The number of registered users at a given day |
| cnt (Count) | Total Rentals with both casual and registered users |

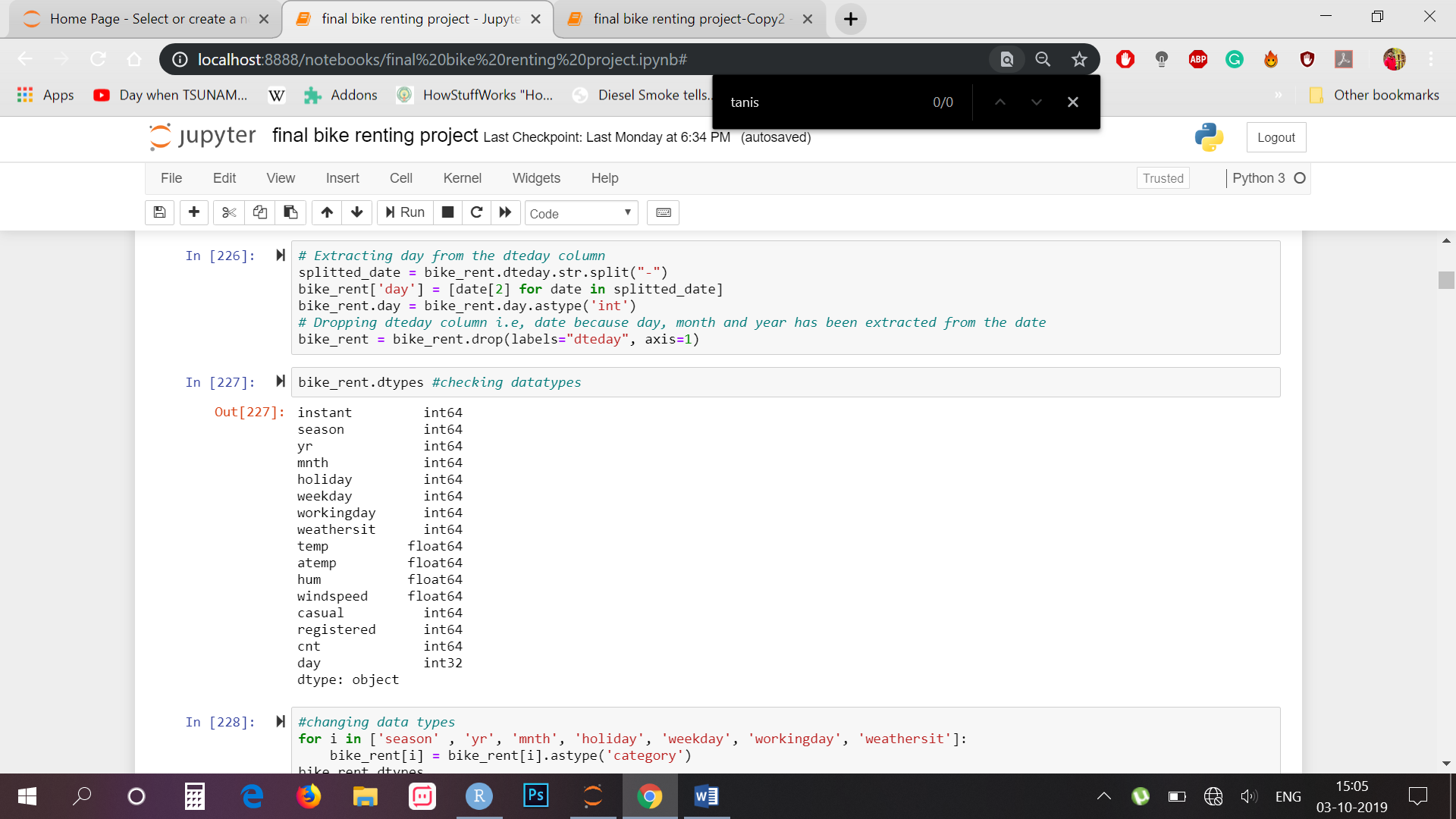
This dataset contains the rental count in between year 2011 and 2012 based on seasonal and environment etc... This is a new way for traditional bike rent. The whole process from registration to rental return back is automated. Here is our data explanation

* ***Methodology***

***Data Pre Processing***: Before we proceeding to create our model on top of the provided data. It is necessary to do Exploratory Data Analysis. EDA is very first and necessary step to take before proceeding further. As the result depends on the data, EDA makes sure the quality of input data is high which will lead to high quality results. We can perform EDA as follows:



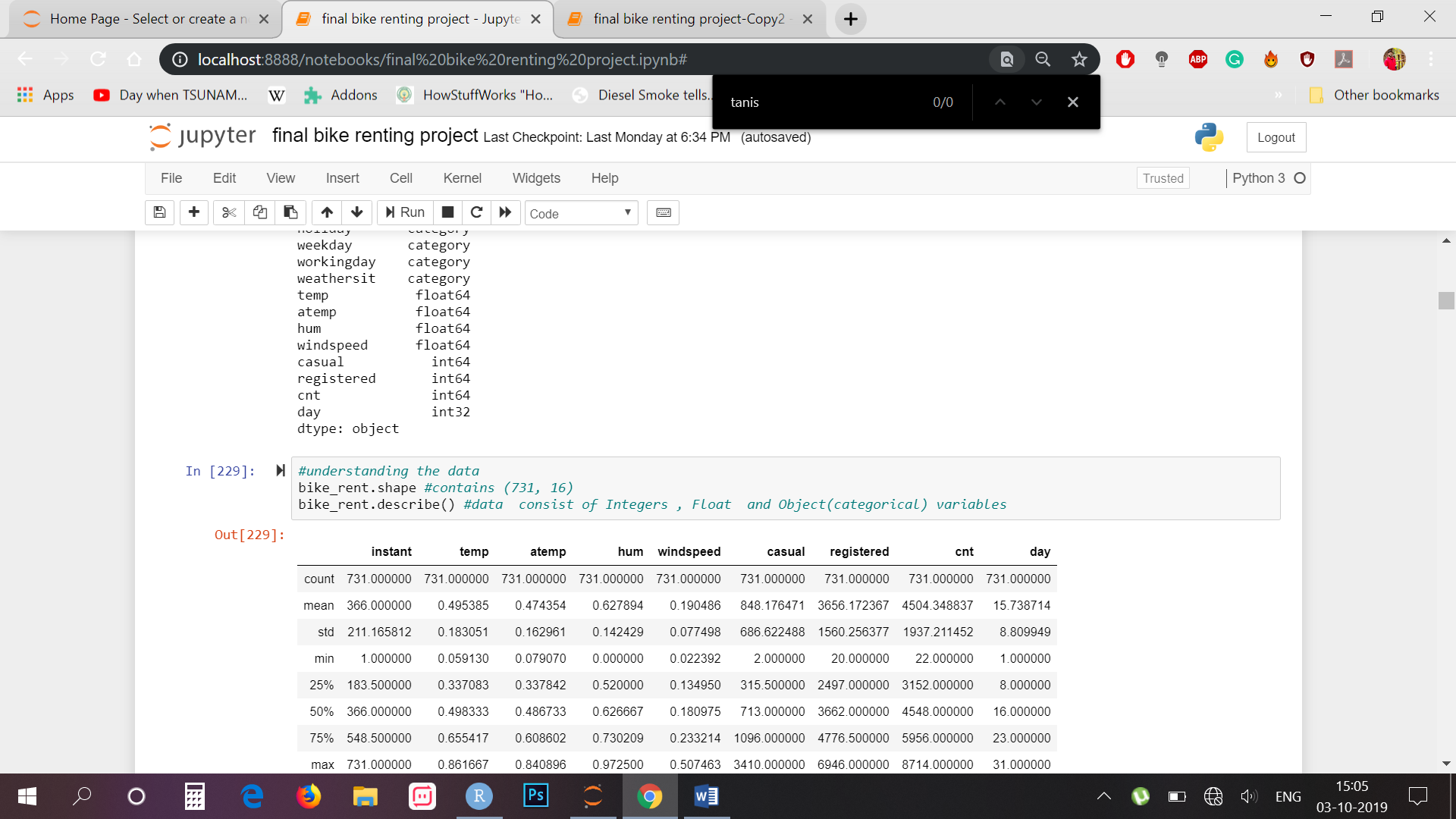
* **Variable Identification:**In Order to understand the data, we need to first, Identifying Predictor (Input) and Target (output) variables. Then, Identifying the data type and category of the variables.



*Types of Variable*: Our Target Variable is ‘CNT’ , and Predictor variables are (dteday,season,yr,mnth,holiday,weekday,workingday,weathersit, temp, atemp,hum,windspeed,casual,registered) .

*DataTypes*:Character(dteday),Numeric(instant,season,yr,mnth,holiday,weekday,workingday,weathersit,casual,registered,cnt ) ,factor(temp,atemp,windspeed).

We have converted the data to category for categorical variable.



*Variable Categories*: Categorical (season, yr, mnth, holiday, weekday, workingday, weathersit), Continious (temp, atemp, hum, windspeed, casual,registered)

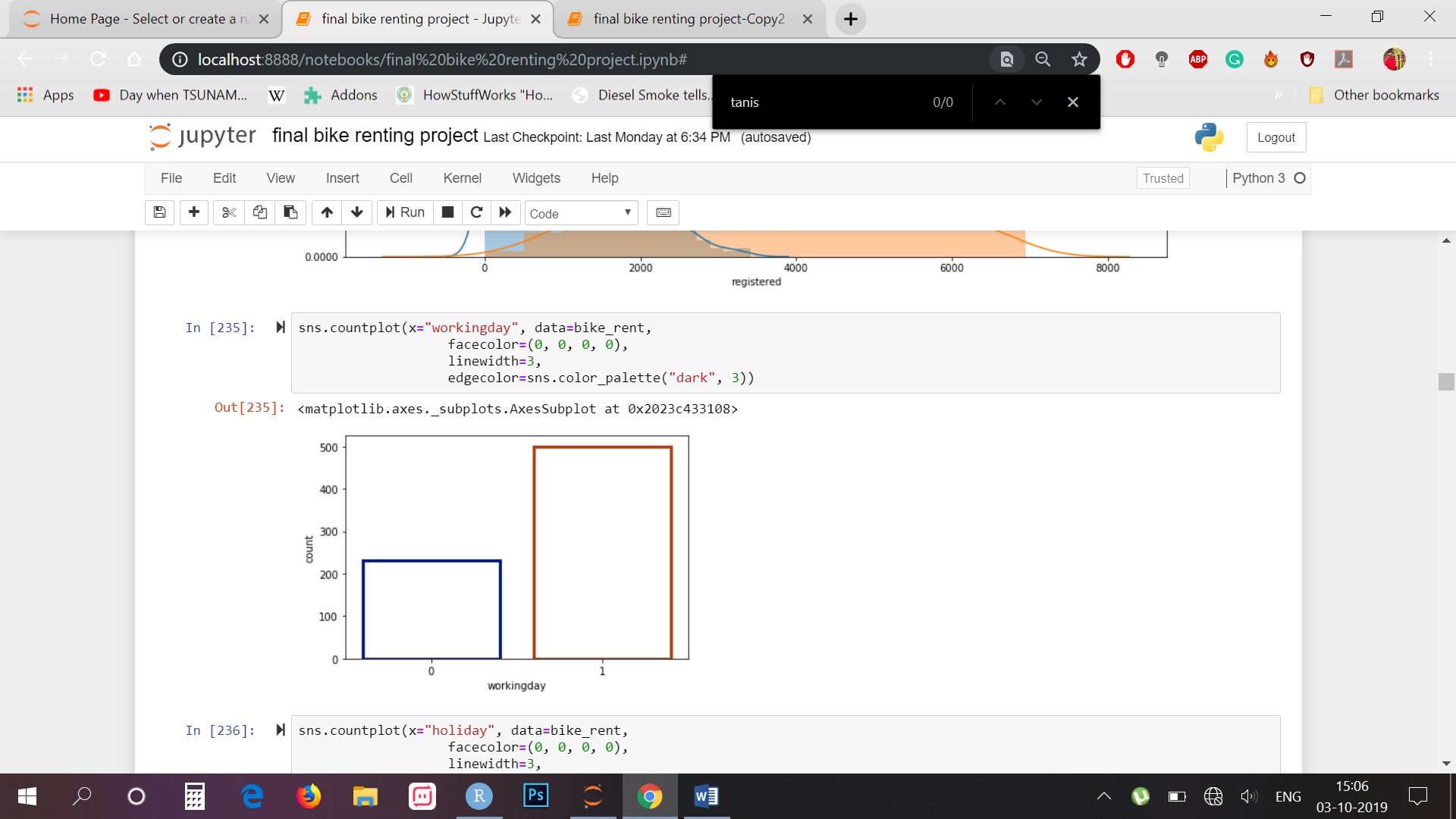
* **Missing values treatment:**
* Missing values occur when no data value is stored for the variable in an observation. Missing values are a common occurrence, and you need to have a strategy for treating them. A missing value can signify a number of different things in your data. Perhaps the data was not available or not applicable or the event did not happen. It could be that the person who entered the data did not know the right value, or missed filling in. Typically, ignore the missing values, or exclude any records containing missing values, or replace missing values with the mean, or infer missing values from existing values.

We check for missing values in our data and came to know that there are no missing values

* **Visualization:**
* Exploring Variables one by one to understand central tendency, spread of the variable, distribution of each category, association and disassociation between variables at a predefined significance level.
* Uni-variate Analysis: **Checking the distribution of individual variables.**

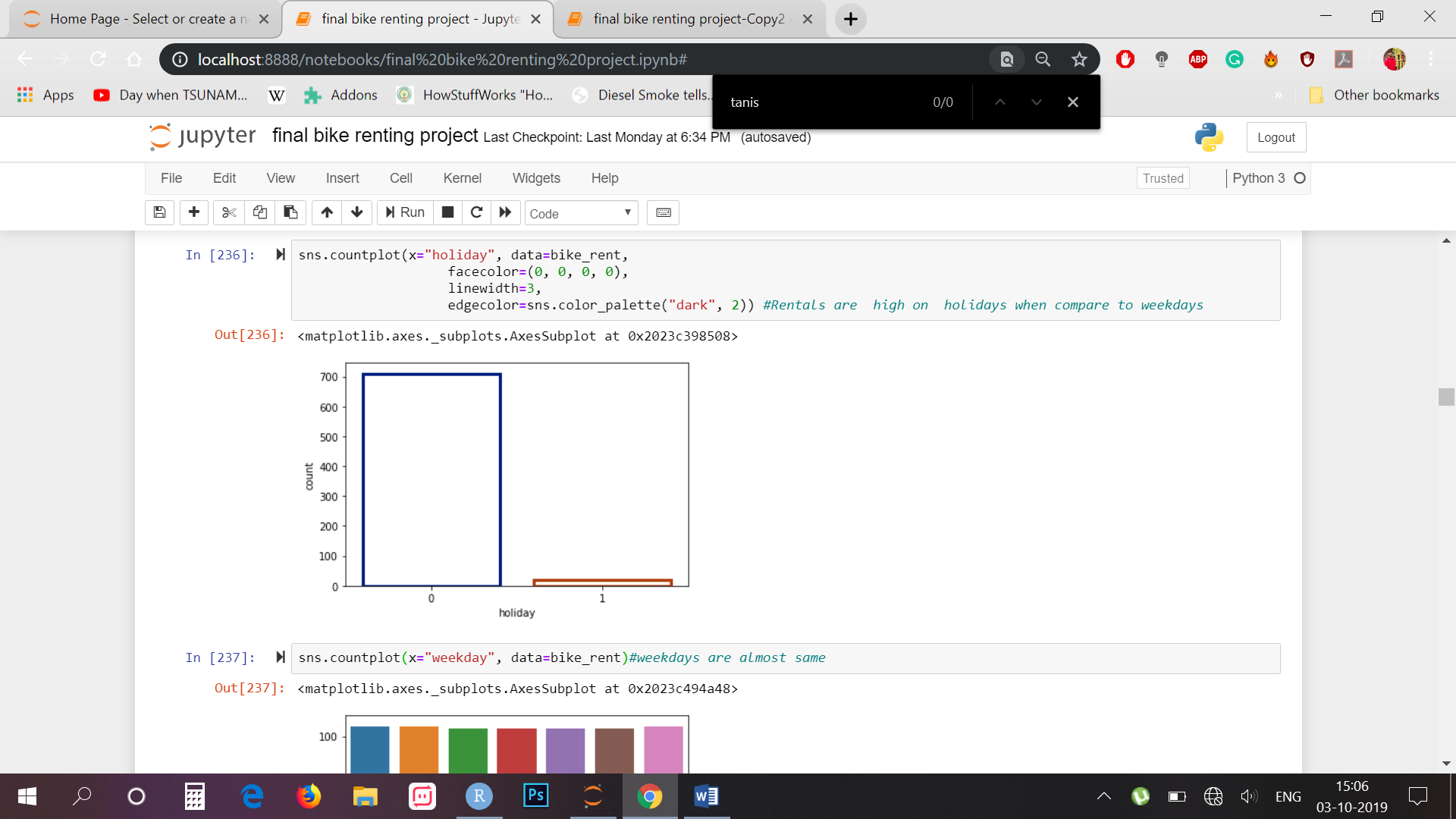


* **Bi-variate Analysis:**
* We are checking relation of variables with each other to understand the relationship of variables with our target variable
* From this, we can infer that the count of rental bikes have increased over the year 2012. In each particular year, month march, may, June, July, Sept have highest rentals. Also we see that rentals are higher in Season3 (fall) and least in Season1 (Spring).

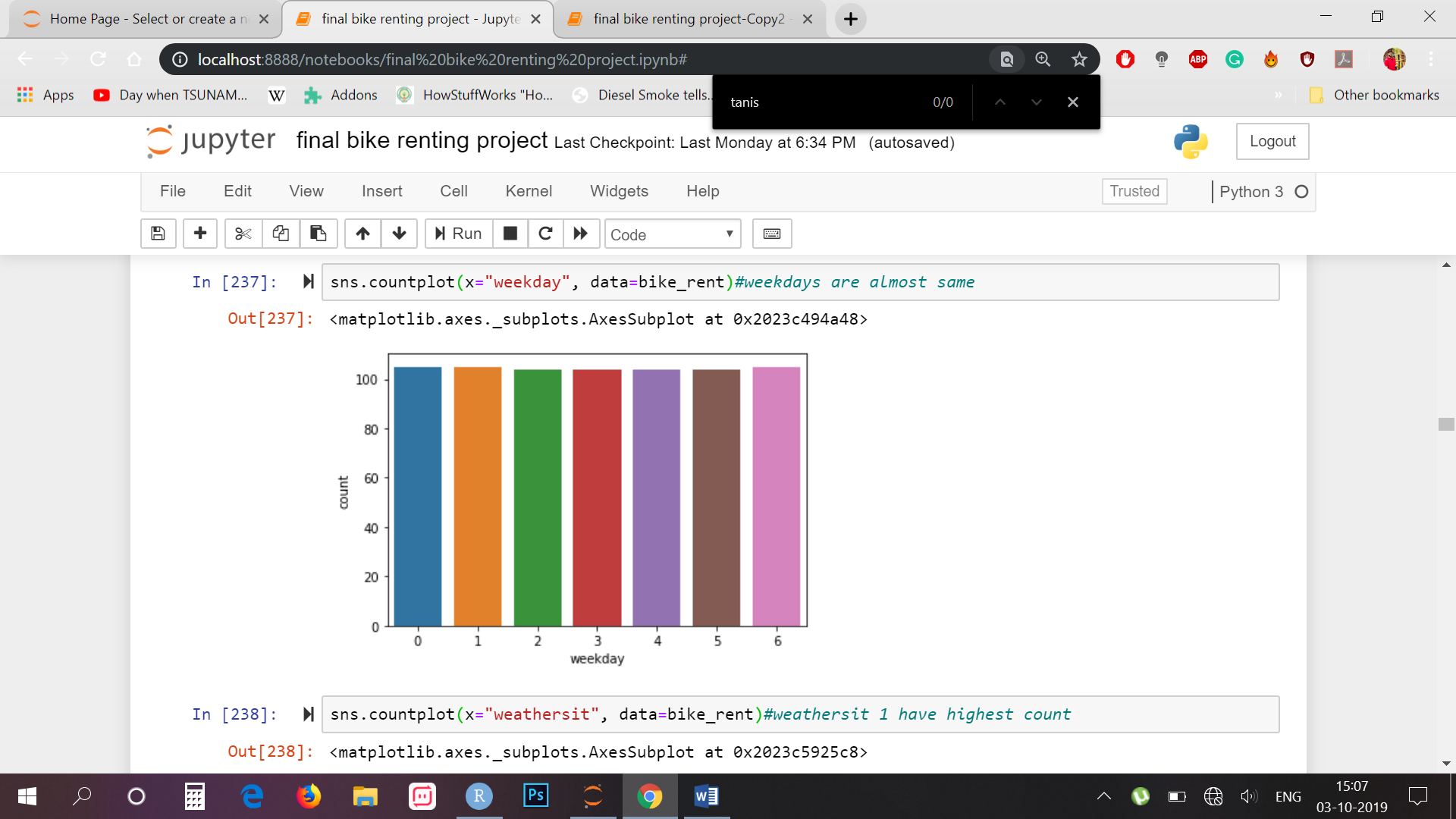


Bar graph of working day

(Bar graph of working day) we can infer that people rent bikes more on working days as there may be chances of rental requirements to commute rather than on non-working days.



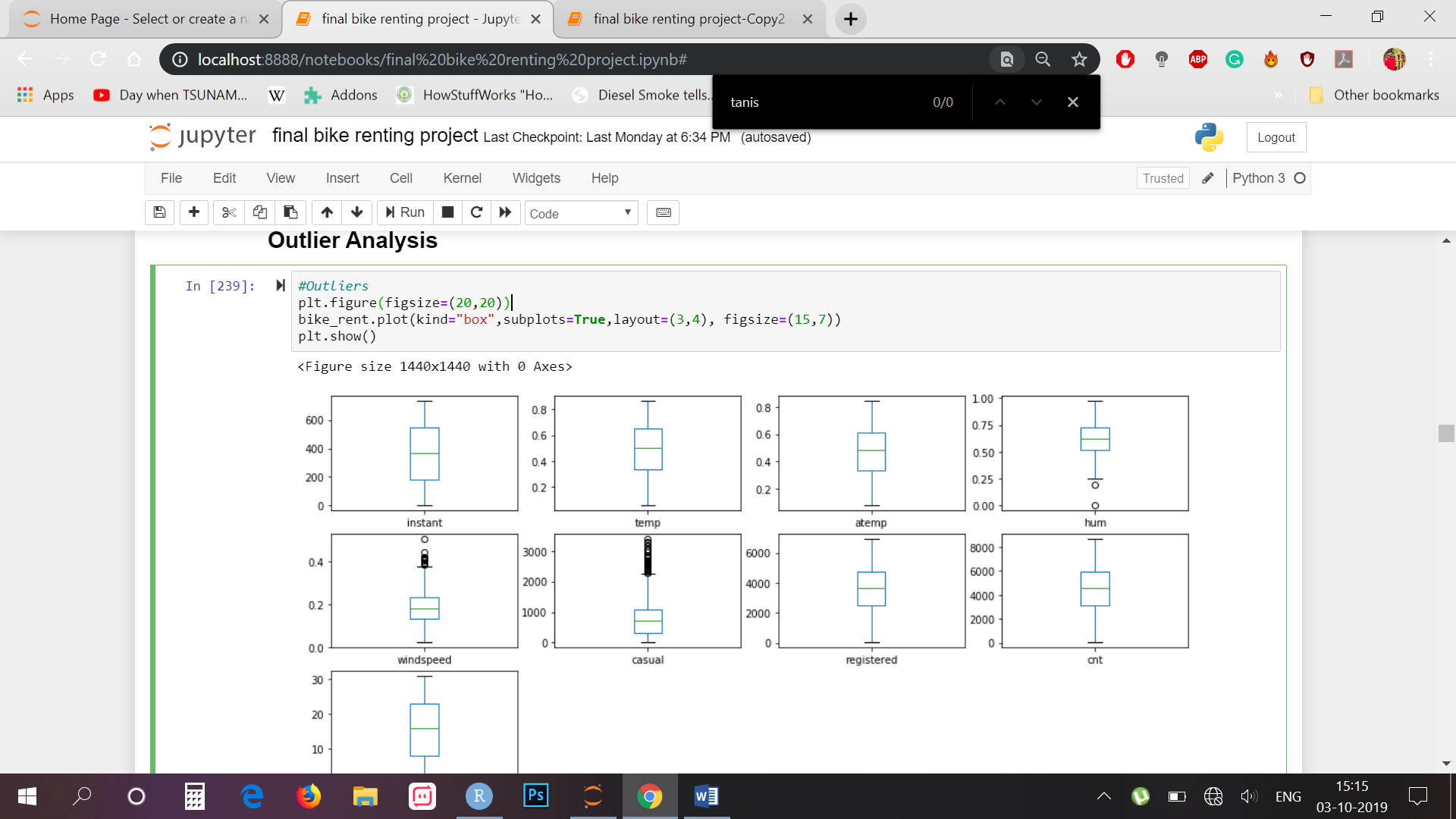
(Bar graph of weather sit) we can infer that on weatersit1 (clear weather), people prefer more to rent bikes rather on weathersit3(heavy rain).

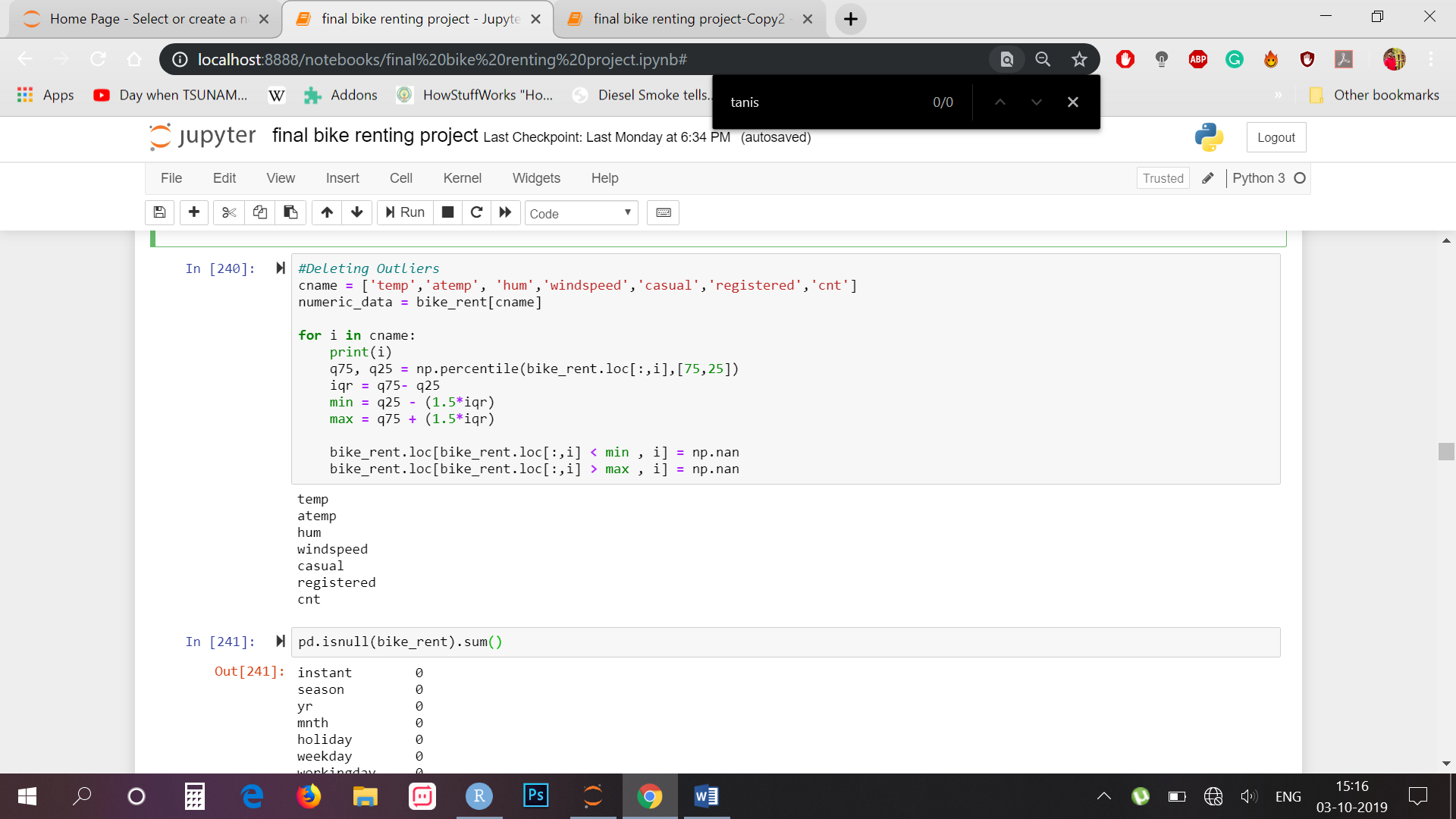


(Boxplot of weekdays) we can clearly see that there is almost no difference in rentals on each day, it means that all the days have almost same rental counts.

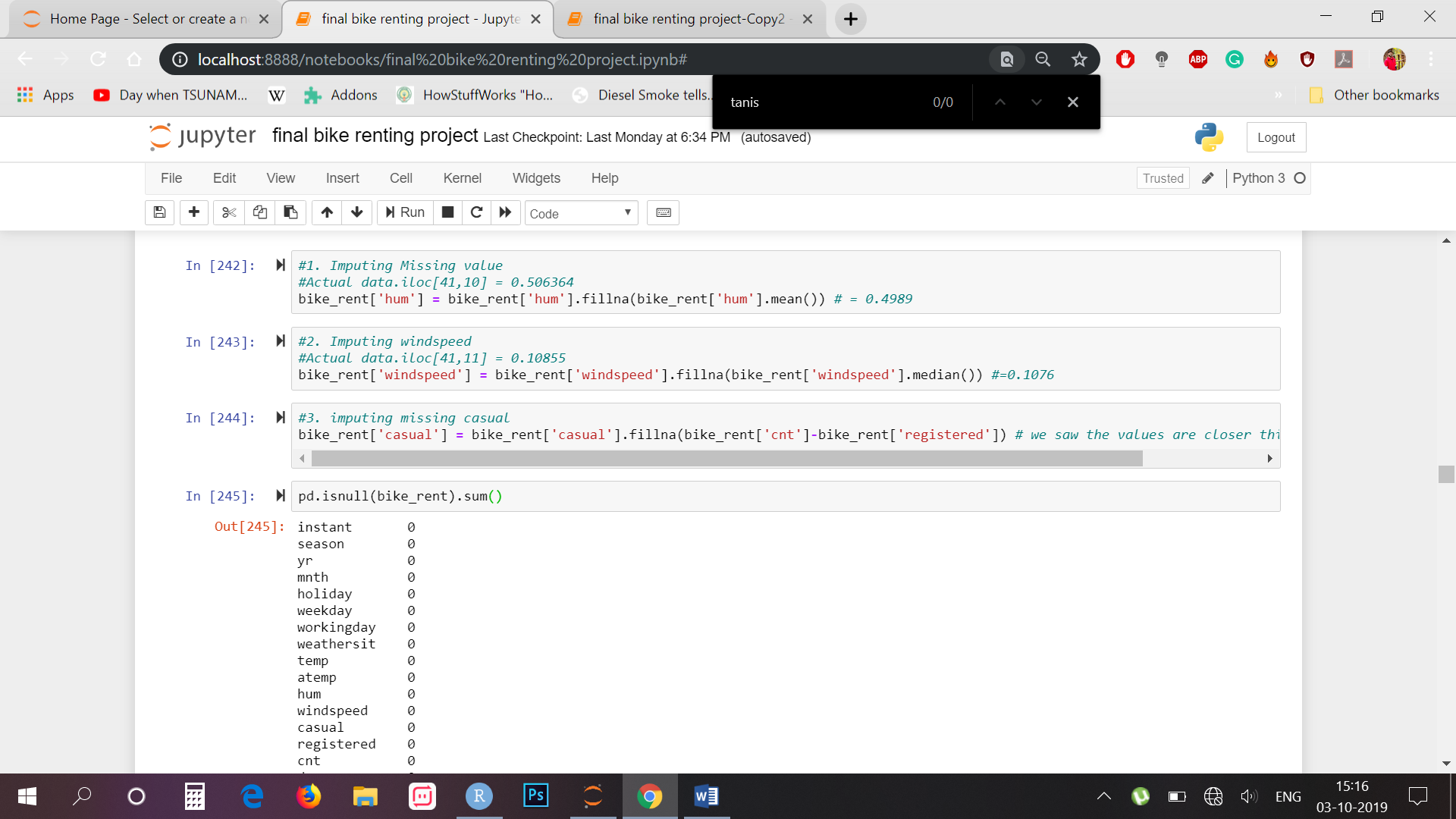
**Outlier treatment:**

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. Outliers can drastically change the results of the data analysis and statistical modelling. There are numerous unfavourable impacts of outliers in the data set. It increases the error variance and reduces the power of statistical tests. If the outliers are non-randomly distributed, they can decrease normality. They can also impact the basic assumption of Regression, ANOVA and other statistical model assumptions. The boxplot for our data could be seen as follows:





* The boxplot helps us to identify outliers in each column. In our data, outliers are found in humidity, windspeed and casual columns. Outliers in humidity were imputed with mean of that column. (only 2 outliers). Outliers in windspeed were imputed with season wise mean of windspeed. Outliers in casual is imputed by subtracting registration number from cnt, as I saw the casual counts are nearer to this equation.



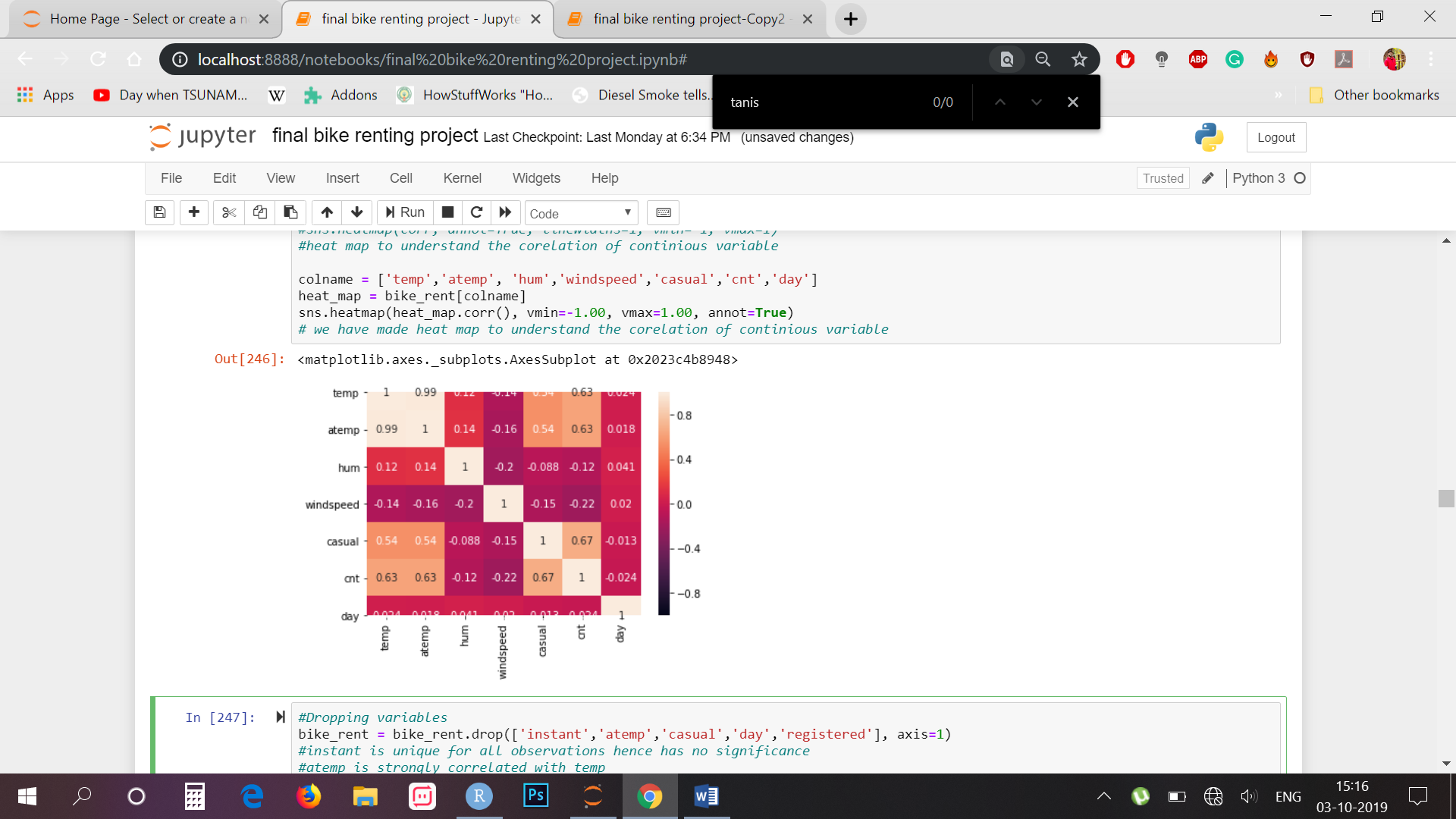
**Feature Selection :**

* Variable selection is an important aspect of model building. It helps in building predictive models free from correlated variables, biases and unwanted noise. It helps in selecting a subset of relevant features (variables, predictors) for use in model construction and subset of a learning algorithm’s input variables upon which it should focus attention, while ignoring the rest.

**Correlation Analysis :**

* We make heat map to understand the co relation of contiguous variable. A heatmap is a graphical representation of data where the individual values contained in a matrix are represented as colors. Here each numerical variable’s correlation is mapped with each other’s in a matrix which has been plotted in the following heatmap.

Our correlation matrix shows results as follows



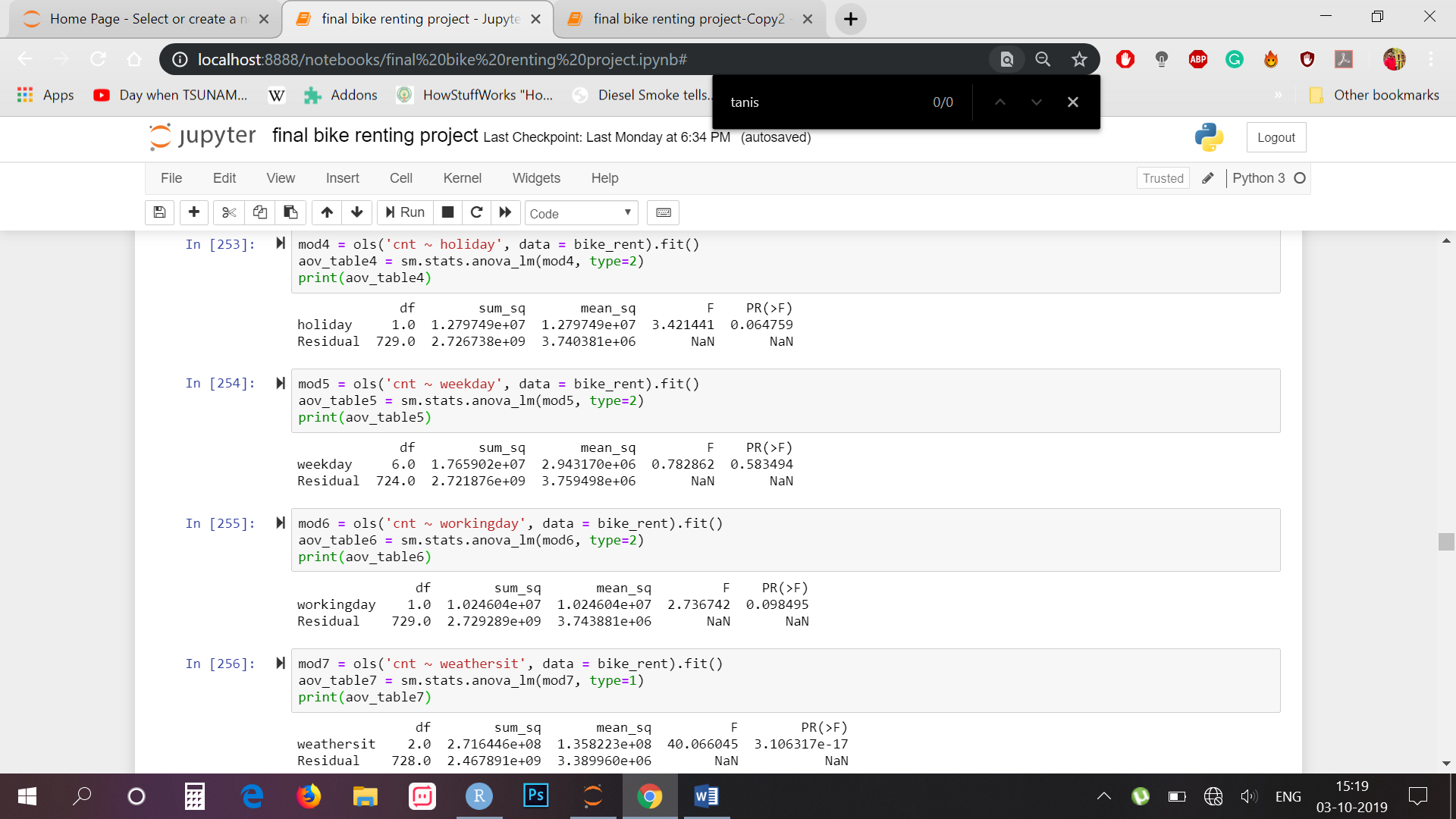
* Temp and atemp are highly correlated 2. Registered and count are highly correlated (since count is a sum of registered and casual)
* We can now remove one of the highly correlated variable so that our model can perform well with much accuracy.
* So, we are dropping variables('instant','atemp','casual','registered'). Instant is unique for all observations hence has no significance, atemp is strongly correlated with temp, cnt is sum of casual and registration.

Let’s check the importance of our features to our target variable.

**ANOVA** (Analysis of Variances)

Analysis of variance (ANOVA) is a statistical technique that is used to check if the means of two or more groups are significantly different from each other. ANOVA checks the impact of one or more factors by comparing the means of different samples. As our target variable is numerical we will use ANOVA for feature selection technique to see whether any categorical variable is related to target variable. The higher the variance between the variables, the less likely that they are related (or correlated). The result of anova is as follows:





* Here , we can see the the importance of Holiday(0.00344259) and workingday (0.0048055) is extremly low . We can remove these variables for our dataset as its importance is not much to be considred.

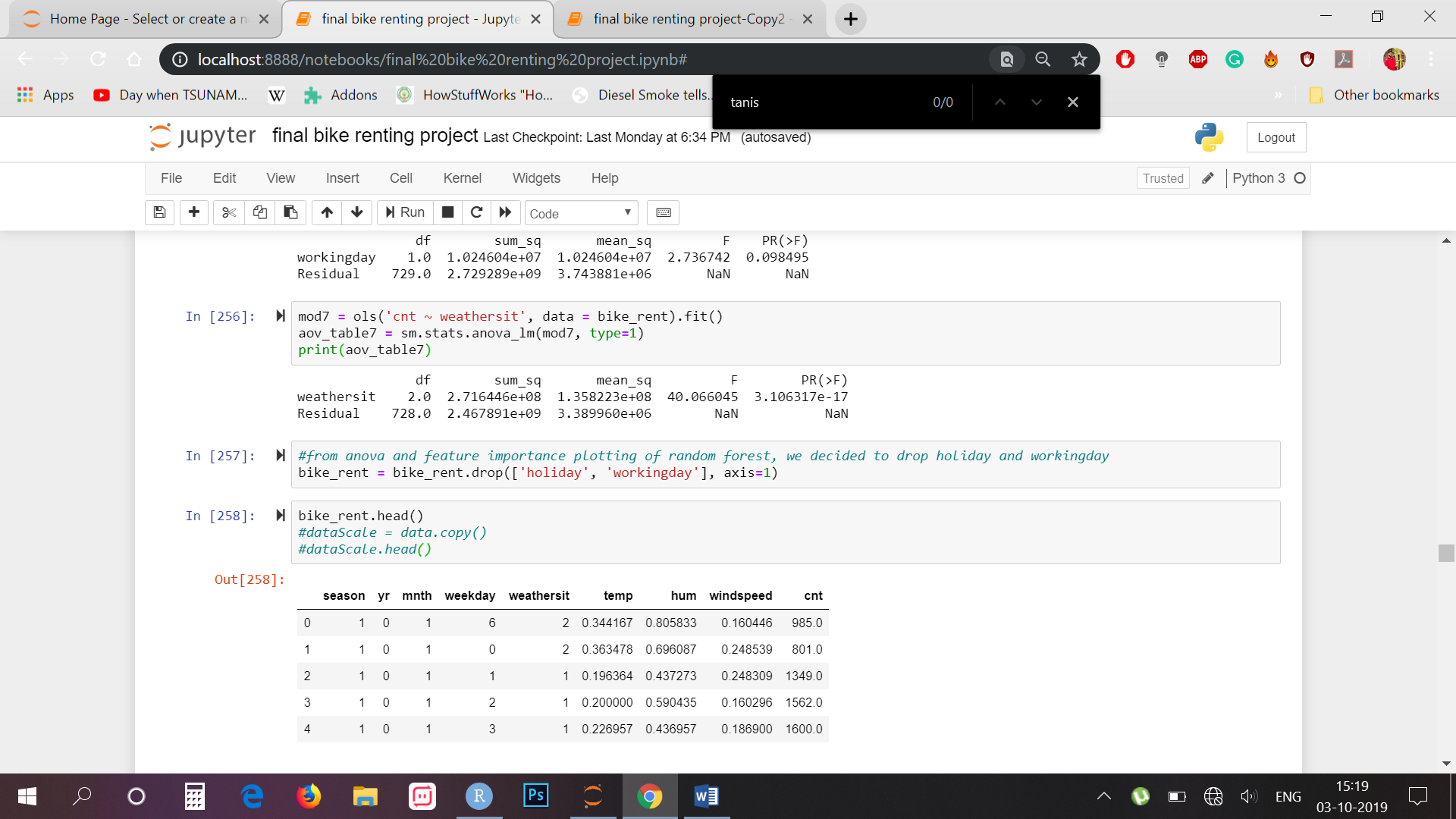
Ho = Categorical variable is Independent from the Target variable

Ha = Categorical variable is Dependent on the Target variable

--> If the p value of the categorical variable is less than 0.05 then we will consider that the target variable is dependent on the categorical variable for which we reject the null hypothesis.

--> From the above result we can see that only five variables are very much related to target variable hence we delete all the other variables.

--> Therefore from both the correlation analysis and ANOVA we got some variable which we shouldn’t consider for further processing. The variables that could be deleted are as follows



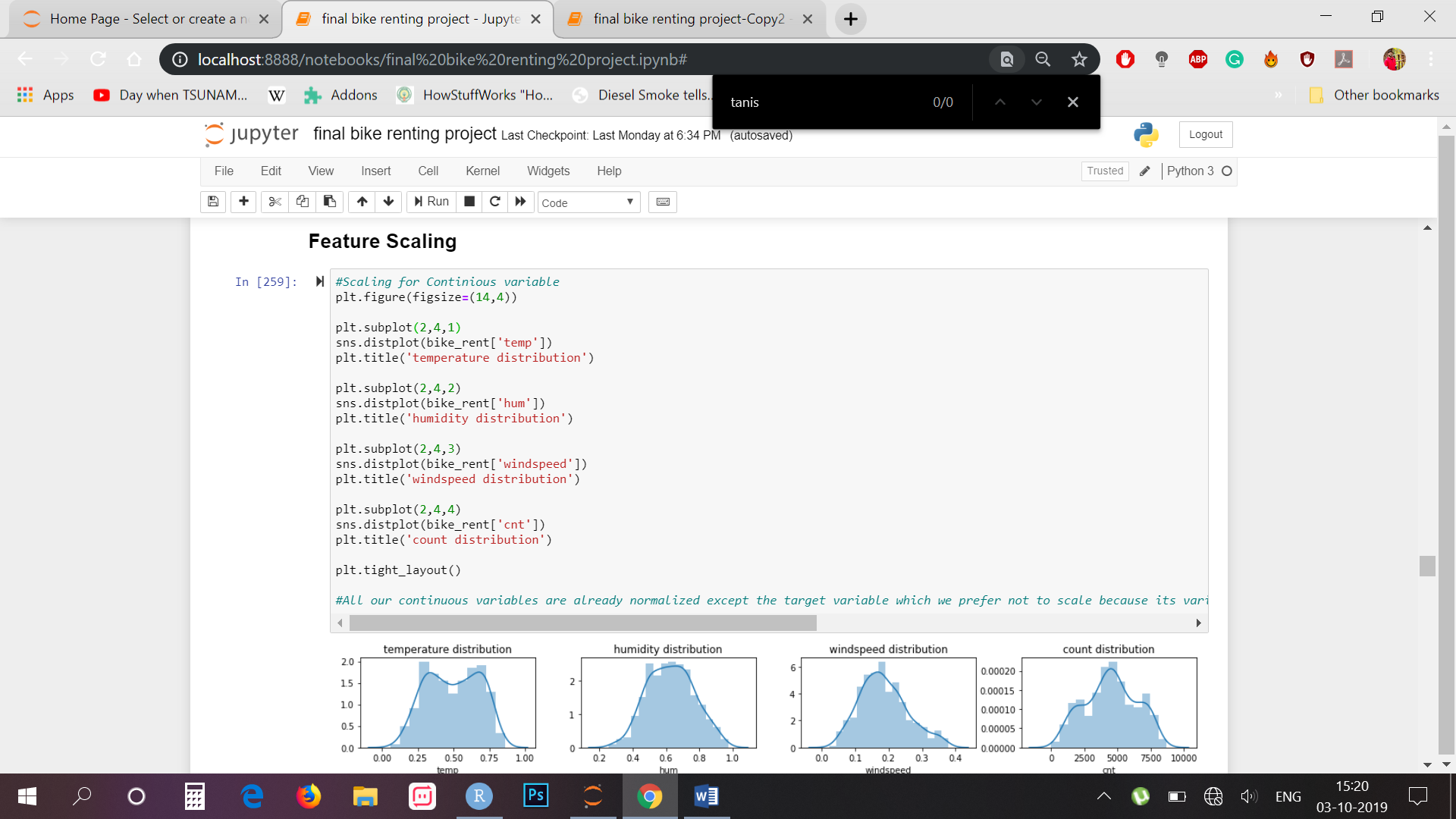
Numerical: Instant, atemp, casual, registered

Categorical: day, holiday, workingday

Hence, We have deleted these Variables from our data set.

**FeatureScaling:**

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization. Normalization also called Min-Max scaling. It is the process of reducing unwanted variation either within or between variables. Normalization brings all of the variables into proportion with one another. It transforms data into a range between 0 and 1. All our continuous variables are already normalized except the target variable which we prefer not to scale because its variation is spread quite widely and after scaling, the difference between the number is diminishing.



***Modeling***

**Model Selection:**For modelling, we are going to use some famous models to our data-set and will conclude the result according to it.

Cross validation

Cross Validation is a technique which involves reserving a particular sample of a dataset on which you do not train the model. Later, you test your model on this sample before finalizing it.

Here are the steps involved in cross validation:

1. You reserve a sample data set

2. Train the model using the remaining part of the dataset

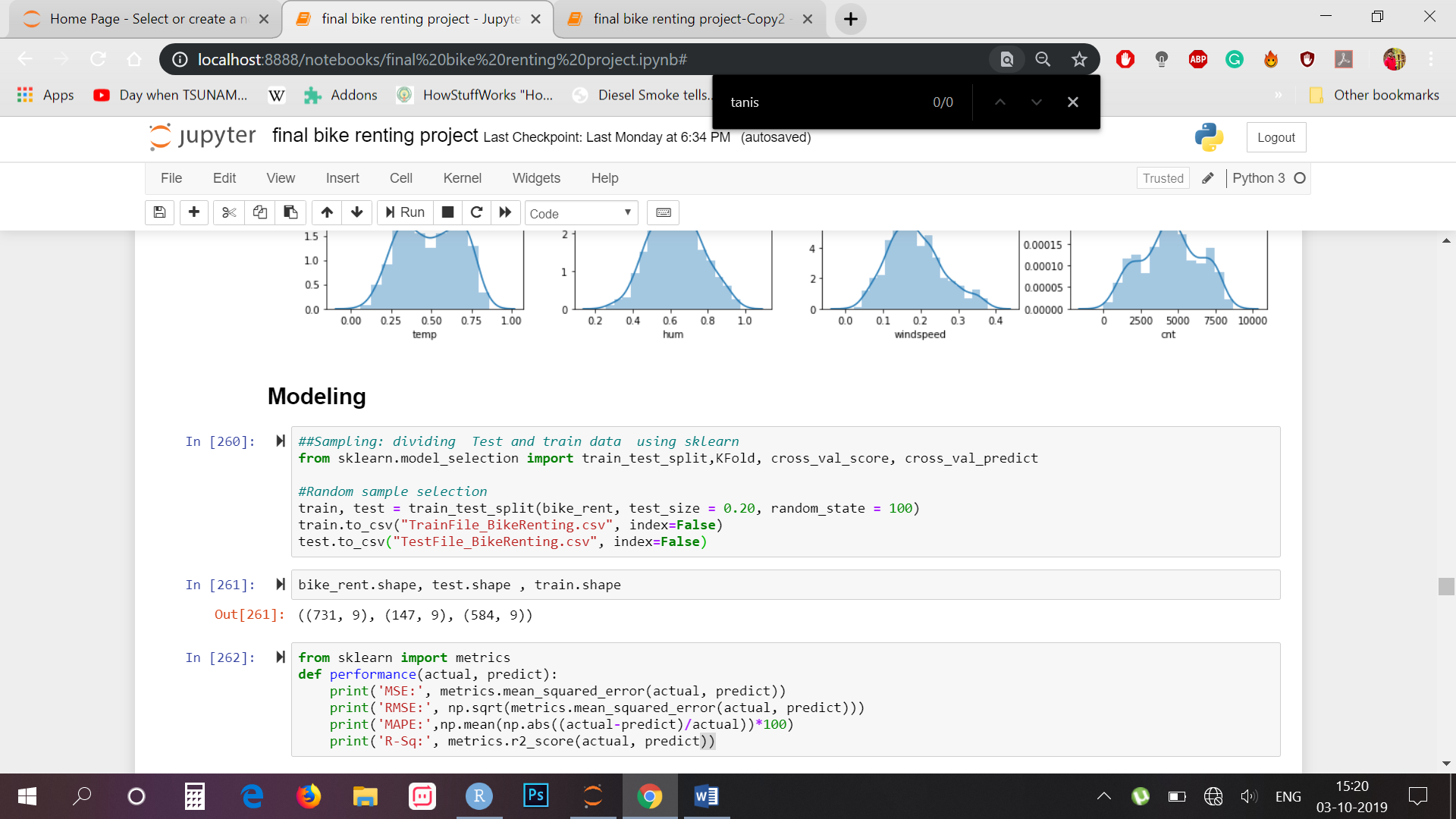
3. Use the reserve sample of the test (validation) set. This will help you in gauging the effectiveness of your model’s performance. If your model delivers a positive result on validation data, go ahead with the current model

We need to remember that:

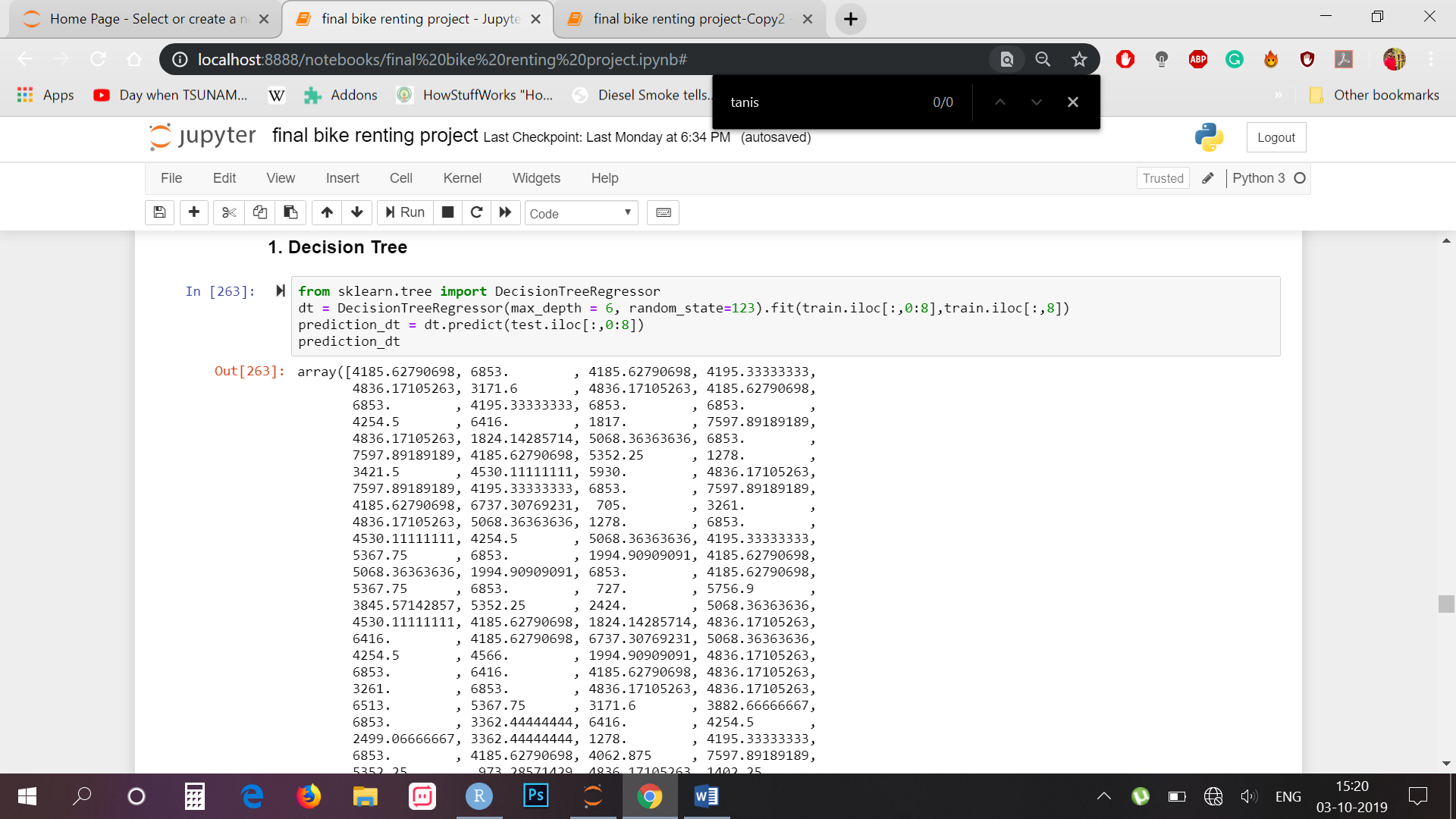
1. We should train the model on a large portion of the dataset. Otherwise we’ll fail to read and recognize the underlying trend in the data. This will eventually result in a higher bias

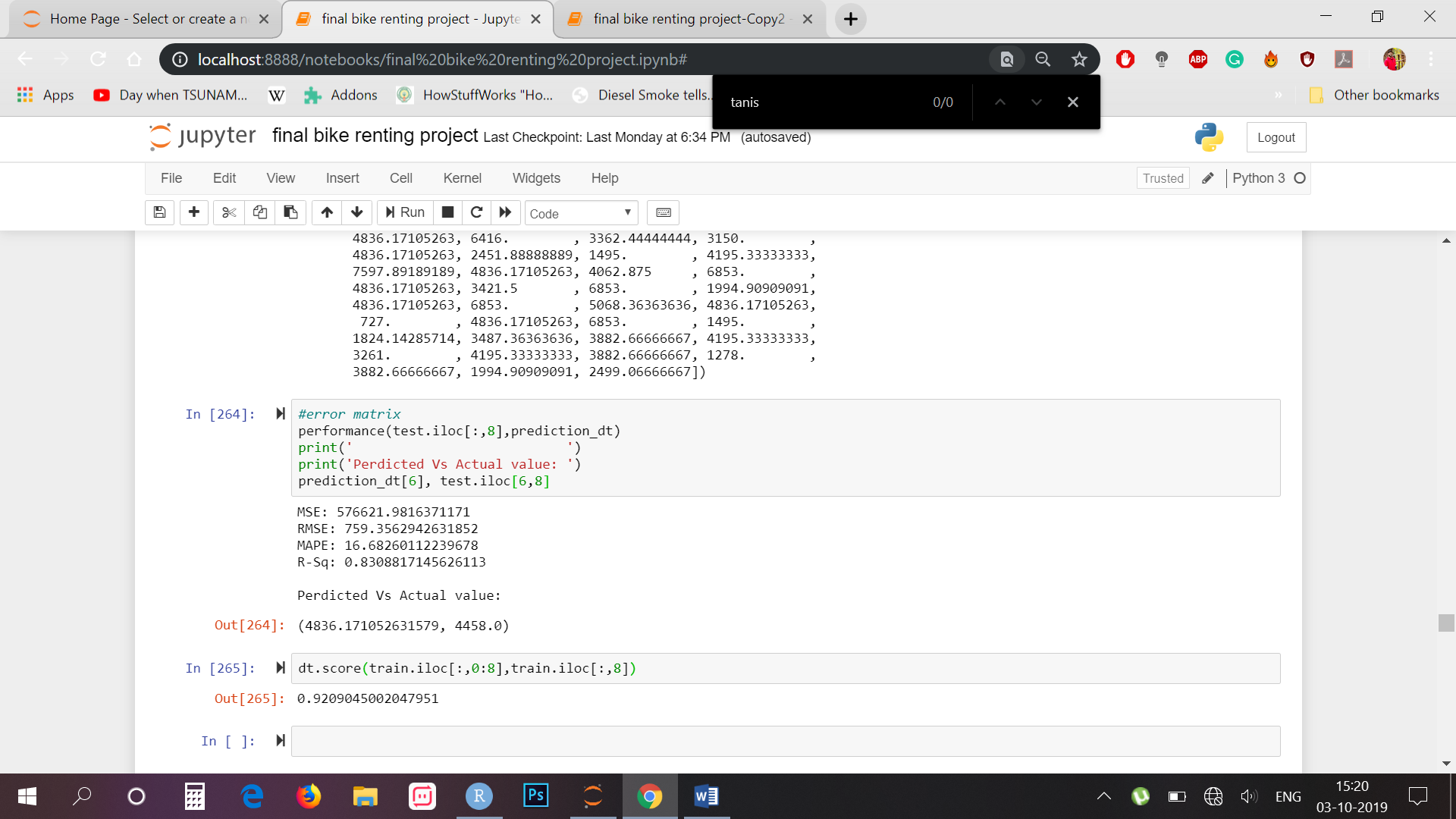
2. We also need a good ratio of testing data points. As we have seen above, less amount of data points can lead to a variance error while testing the effectiveness of the model

3. We should iterate on the training and testing process multiple times. We should change the train and test dataset distribution. This helps in validating the model effectiveness properly

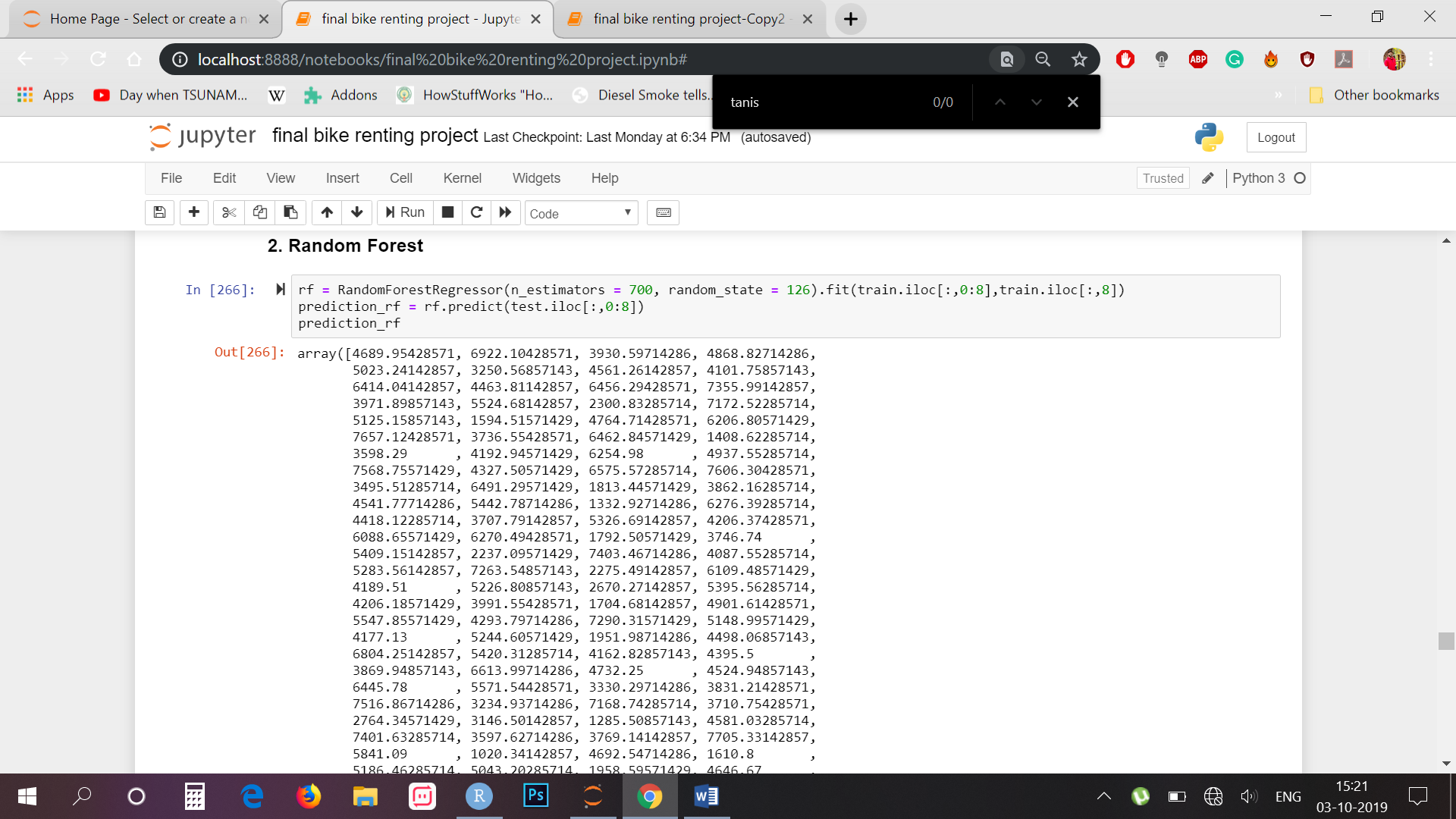


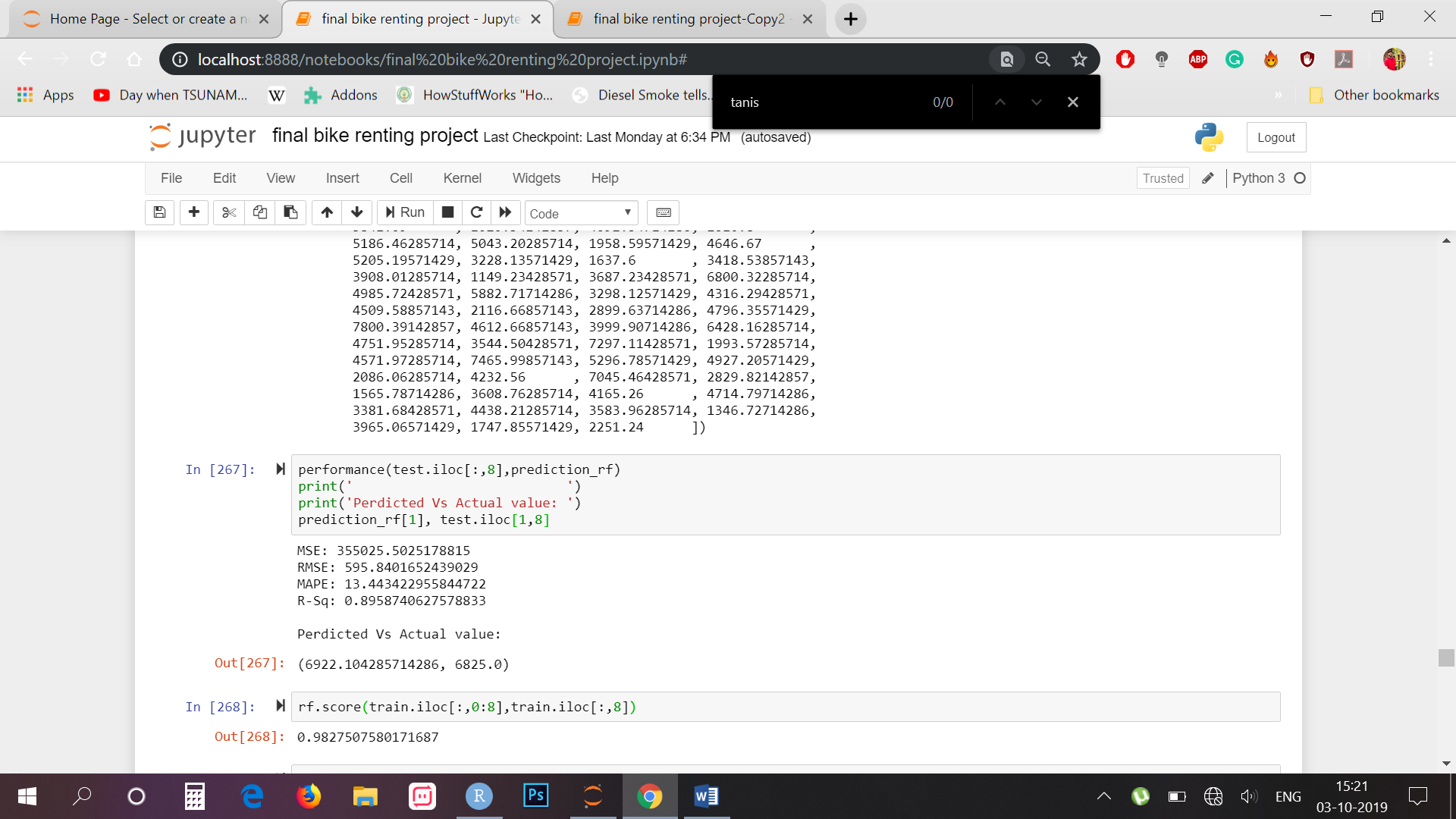
1. **Decision Tree:** Decision tree is a rule. Each branch connects nodes with “and” and multiple branches are connected by “or”. It can be used for classification and regression. It is a supervised machine learning algorithm. Accept continuous and categorical variables as independent variables. Extremely easy to understand by the business users. Split of decision tree is seen in the below tree. Decision tree regression is as follows





1. **Random Forest:**Random Forest or decision tree forests are an ensemble learning method for classification, regression and other tasks. It consists of an arbitrary number of simple trees, which are used to determine the final outcome. In the regression problem, their responses are averaged to obtain an estimate of the dependent variable. Using tree ensembles can lead to significant improvement in prediction accuracy (i.e., better ability to predict new data cases). The goal of using a large number of trees is to train enough that each feature has a chance to appear in several model.



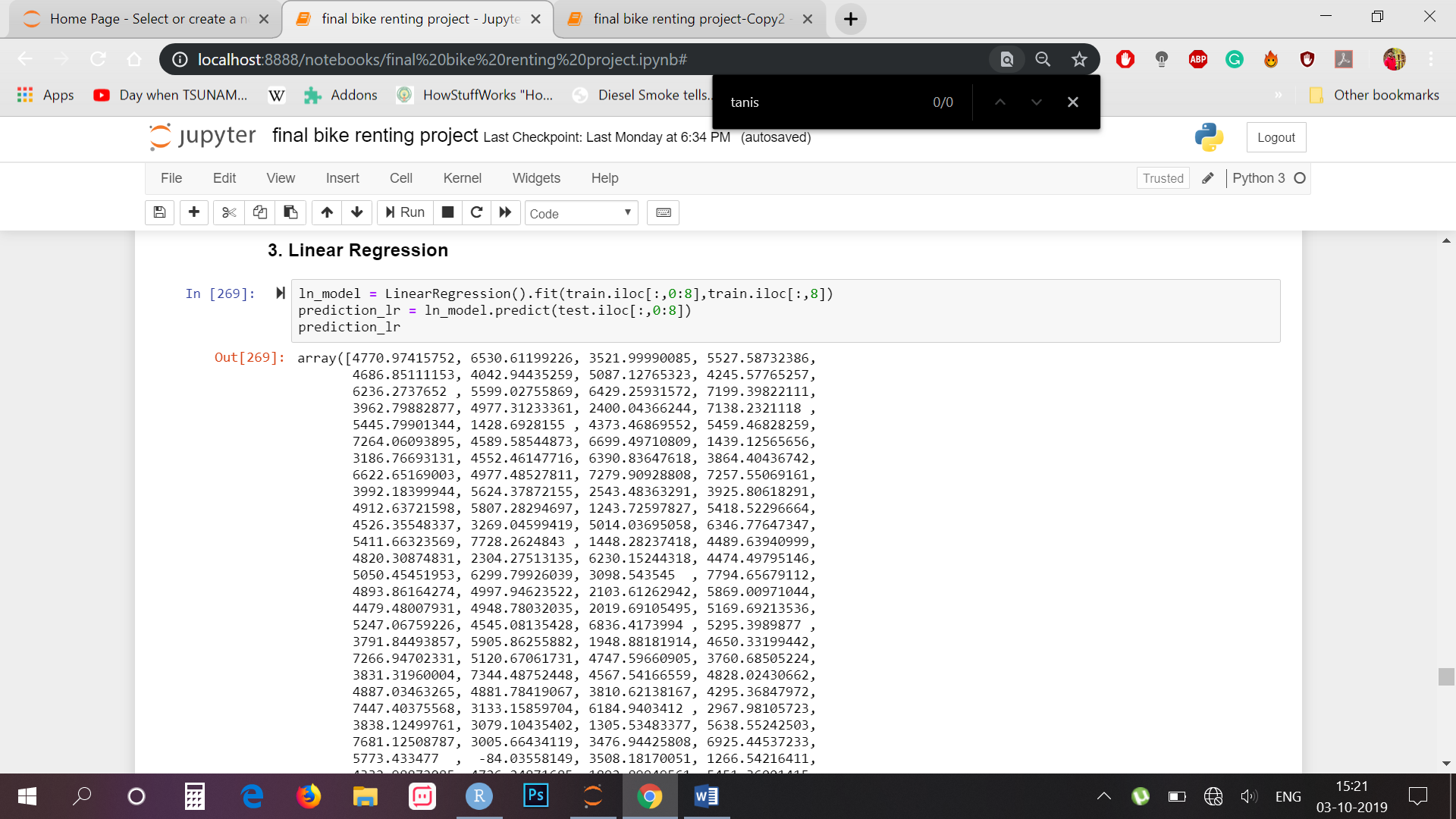


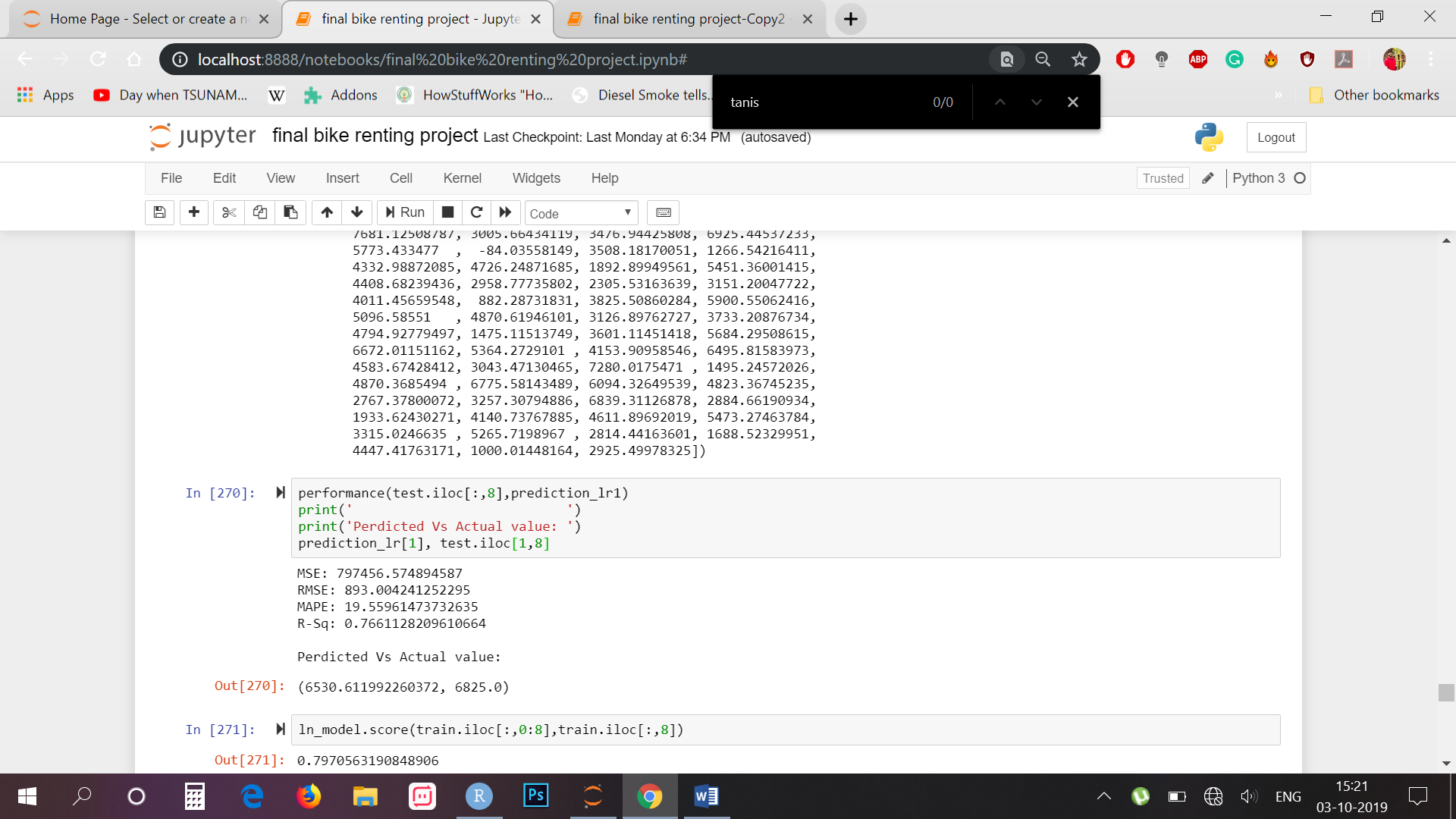
--> As we increase the number of trees the error count decrease until a point and then becomes constant. Error vs number of trees to be used graph is as follows:

It shows that if a variable is assigned values by random permutation by how much will the MSE increase. Higher the value, higher the importance. On the other hand, node purity is measured by the Gini index which is the difference between before and after split on that variable.

1. **Linear Regression:**Linear regression is the most basic type of regression and commonly used predictive analysis. Linear regression is an approach for modelling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

We have to convert all the categorical variable into Dummies because Machine learning algorithm require numbers as input. In regression analysis, a dummy variable (also known as an indicator variable, design variable, Boolean indicator, binary variable, or qualitative variable) is one that takes the value 0 or 1 to indicate the absence or presence of some categorical effect that may be expected to shift the outcome



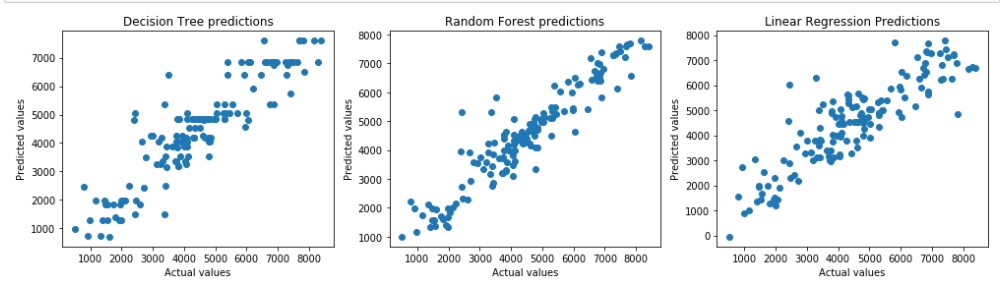


***Conclusion:***

***Model Evaluation:*** Model evaluation is done on basis of evaluation metrics or error metrics. Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results. Simply, building a predictive model is not our motive. But, creating and selecting a model which gives high accuracy on out of sample data. Hence, it is crucial to check accuracy or other metric of the model prior to computing predicted values. In our data as we applied regression models we have error metrics like Mean square error(MSE), MAPE, Root mean square error (RMSE), Mean absolute error (MAE).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Language/Model |  | **Python** |  | |  |
| **MODELS** | MSE | RMSE | MAPE | R-SQ |
| **Decision Tree** | 58079.5 | 762.09 | 16.6 | 0.8496 |
| **Random Forest** | 376333 | 613.46 | 13.62 | 0.8996 |
| **Linear regression** | 782200 | 884.42 | 19.25 | 0.7605 |

***Model Selection :*** We can see that all models perform comparatively on average and therefore we select Random forest classifier models for better prediction.



From the above plots of Actual Vs Predicted values, we can infer that values of Random forest falls on straight line indicating random forest fits better than the other three models.

**Python-Code**



*import os #Intraction local system directories*

*import pandas as pd #Data processing*

*import numpy as np #Linear alfrom sklearn.tree import DecisionTreeRegressor*

*from sklearn.ensemble import RandomForestRegressor*

*from sklearn.linear\_model import LinearRegression*

*import matplotlib.pyplot as plt # some plotting!*

*import seaborn as sns # so For Plots!*

*%matplotlib inline*

In [2]:



os.chdir('C:/Users/Sridhar/Desktop/Bike renting')

bike\_rent **=** pd.read\_csv('day.csv', sep **=** ',')

os.getcwd()

In [ ]:



bike\_rent.shape

In [ ]:



bike\_rent.head()

In [ ]:



​

## Eploratory Data Analysis

In [ ]:



*# Extracting day from the dteday column*

splitted\_date **=** bike\_rent.dteday.str.split("-")

bike\_rent['day'] **=** [date[2] **for** date **in** splitted\_date]

bike\_rent.day **=** bike\_rent.day.astype('int')

*# Dropping dteday column i.e, date because day, month and year has been extracted from the date*

bike\_rent **=** bike\_rent.drop(labels**=**"dteday", axis**=**1)

In [ ]:



bike\_rent.dtypes *#checking datatypes*

In [ ]:



*#changing data types*

**for** i **in** ['season' , 'yr', 'mnth', 'holiday', 'weekday', 'workingday', 'weathersit']:

bike\_rent[i] **=** bike\_rent[i].astype('category')

bike\_rent.dtypes

In [ ]:



*#understanding the data*

bike\_rent.shape *#contains (731, 16)*

bike\_rent.describe() *#data consist of Integers , Float and Object(categorical) variables*

In [ ]:



*#Numerical columns*

numerical\_columns **=** bike\_rent.select\_dtypes(include**=**[np.number]).columns.tolist()

numerical\_columns

In [ ]:



*#Categorical columns*

categorical\_columns **=** bike\_rent.select\_dtypes(exclude**=**[np.number]).columns.tolist()

categorical\_columns

## Missing Value Analysis

In [ ]:



*#missing value analysis*

total **=** bike\_rent.isnull().sum().sort\_values(ascending**=False**)

percent **=** (bike\_rent.isnull().sum()**/**bike\_rent.isnull().count()).sort\_values(ascending**=False**)

missing\_data **=** pd.concat([total, percent], axis**=**1, keys**=**['Total', 'Percent'])

missing\_data

*## There are NO Missing Values in our Data*

## Visulisation Of Data

### Univariate Analysis

In [ ]:



*# Target variable analysis*

*#Check whether target variable is normal or not*

sns.distplot(bike\_rent['cnt']);

In [ ]:



*#Check whether variable 'casual'is normal or not*

fig **=** plt.figure(figsize**=**(15,6))

sns.distplot(bike\_rent['casual']);

*#Check whether variable 'registered'is normal or not*

sns.distplot(bike\_rent['registered']);

fig.legend(labels**=**['casual','registered'])

plt.show()

In [ ]:



sns.countplot(x**=**"workingday", data**=**bike\_rent,

facecolor**=**(0, 0, 0, 0),

linewidth**=**3,

edgecolor**=**sns.color\_palette("dark", 3))

In [ ]:



sns.countplot(x**=**"holiday", data**=**bike\_rent,

facecolor**=**(0, 0, 0, 0),

linewidth**=**3,

edgecolor**=**sns.color\_palette("dark", 2)) *#Rentals are high on holidays when compare to weekdays*

In [ ]:



sns.countplot(x**=**"weekday", data**=**bike\_rent)*#weekdays are almost same*

In [ ]:



sns.countplot(x**=**"weathersit", data**=**bike\_rent)*#weathersit 1 have highest count*

## Outlier Analysis

In [ ]:



*#Outliers*

plt.figure(figsize**=**(20,20))

bike\_rent.plot(kind**=**"box",subplots**=True**,layout**=**(3,4), figsize**=**(15,7))

plt.show()

In [ ]:



*#Deleting Outliers*

cname **=** ['temp','atemp', 'hum','windspeed','casual','registered','cnt']

numeric\_data **=** bike\_rent[cname]

**for** i **in** cname:

print(i)

q75, q25 **=** np.percentile(bike\_rent.loc[:,i],[75,25])

iqr **=** q75**-** q25

min **=** q25 **-** (1.5**\***iqr)

max **=** q75 **+** (1.5**\***iqr)

bike\_rent.loc[bike\_rent.loc[:,i] **<** min , i] **=** np.nan

bike\_rent.loc[bike\_rent.loc[:,i] **>** max , i] **=** np.nan

In [ ]:



pd.isnull(bike\_rent).sum()

In [ ]:



*#1. Imputing Missing value*

*#Actual data.iloc[41,10] = 0.506364*

bike\_rent['hum'] **=** bike\_rent['hum'].fillna(bike\_rent['hum'].mean()) *# = 0.4989*

In [ ]:



*#2. Imputing windspeed*

*#Actual data.iloc[41,11] = 0.10855*

bike\_rent['windspeed'] **=** bike\_rent['windspeed'].fillna(bike\_rent['windspeed'].median()) *#=0.1076*

In [ ]:



*#3. imputing missing casual*

bike\_rent['casual'] **=** bike\_rent['casual'].fillna(bike\_rent['cnt']**-**bike\_rent['registered']) *# we saw the values are closer this way that is 245*

In [ ]:



pd.isnull(bike\_rent).sum()

## Feature Selection

In [ ]:



*#drawing correlation matrix between all numeric variables and analyse what are the variables are important*

*# heat map*

*#plt.figure(figsize=(15, 9))*

*#corr = data.corr()*

*#sns.heatmap(corr, annot=True, linewidths=1, vmin=-1, vmax=1)*

*#heat map to understand the corelation of continious variable*

​

colname **=** ['temp','atemp', 'hum','windspeed','casual','cnt','day','registered']

heat\_map **=** bike\_rent[colname]

sns.heatmap(heat\_map.corr(), vmin**=-**1.00, vmax**=**1.00, annot**=True**)

*# we have made heat map to understand the corelation of continious variable*

In [ ]:



*#Dropping variables*

bike\_rent **=** bike\_rent.drop(['instant','atemp','casual','day','registered'], axis**=**1)

*#instant is unique for all observations hence has no significance*

*#atemp is strongly correlated with temp*

In [ ]:



bike\_rent.head() *#After Drop out dataset contains*

## Anova test

In [ ]:



*#since the target variable is continuous*

**import** statsmodels.api **as** sm

**from** statsmodels.formula.api **import** ols

*#from scipy import stats*

In [ ]:



mod1 **=** ols('cnt ~ season', data **=** bike\_rent).fit()

aov\_table1 **=** sm.stats.anova\_lm(mod1, type**=**2)

print(aov\_table1)

In [ ]:



mod2 **=** ols('cnt ~ yr', data **=** bike\_rent).fit()

aov\_table2 **=** sm.stats.anova\_lm(mod2, type**=**2)

print(aov\_table2)

In [ ]:



mod3 **=** ols('cnt ~ mnth', data **=** bike\_rent).fit()

aov\_table3 **=** sm.stats.anova\_lm(mod3, type**=**2)

print(aov\_table3)

In [ ]:



mod4 **=** ols('cnt ~ holiday', data **=** bike\_rent).fit()

aov\_table4 **=** sm.stats.anova\_lm(mod4, type**=**2)

print(aov\_table4)

In [ ]:



mod5 **=** ols('cnt ~ weekday', data **=** bike\_rent).fit()

aov\_table5 **=** sm.stats.anova\_lm(mod5, type**=**2)

print(aov\_table5)

In [ ]:



mod6 **=** ols('cnt ~ workingday', data **=** bike\_rent).fit()

aov\_table6 **=** sm.stats.anova\_lm(mod6, type**=**2)

print(aov\_table6)

In [ ]:



mod7 **=** ols('cnt ~ weathersit', data **=** bike\_rent).fit()

aov\_table7 **=** sm.stats.anova\_lm(mod7, type**=**1)

print(aov\_table7)

In [ ]:



*#from anova and feature importance plotting of random forest, we decided to drop holiday and workingday*

bike\_rent **=** bike\_rent.drop(['holiday', 'workingday'], axis**=**1)

In [ ]:



bike\_rent.head()

*#dataScale = data.copy()*

*#dataScale.head()*

## Feature Scaling

In [ ]:



*#Scaling for Continious variable*

plt.figure(figsize**=**(14,4))

​

plt.subplot(2,4,1)

sns.distplot(bike\_rent['temp'])

plt.title('temperature distribution')

​

plt.subplot(2,4,2)

sns.distplot(bike\_rent['hum'])

plt.title('humidity distribution')

​

plt.subplot(2,4,3)

sns.distplot(bike\_rent['windspeed'])

plt.title('windspeed distribution')

​

plt.subplot(2,4,4)

sns.distplot(bike\_rent['cnt'])

plt.title('count distribution')

​

plt.tight\_layout()

​

*#All our continuous variables are already normalized except the target variable which we prefer not to scale because its variation are spread quite widely and after scaling, the difference between the number is diminishing.*

## Modeling

In [ ]:



*##Sampling: dividing Test and train data using sklearn*

**from** sklearn.model\_selection **import** train\_test\_split,KFold, cross\_val\_score, cross\_val\_predict

​

*#Random sample selection*

train, test **=** train\_test\_split(bike\_rent, test\_size **=** 0.20, random\_state **=** 100)

train.to\_csv("TrainFile\_BikeRenting.csv", index**=False**)

test.to\_csv("TestFile\_BikeRenting.csv", index**=False**)

In [ ]:



bike\_rent.shape, test.shape , train.shape

In [ ]:



**from** sklearn **import** metrics

**def** performance(actual, predict):

print('MSE:', metrics.mean\_squared\_error(actual, predict))

print('RMSE:', np.sqrt(metrics.mean\_squared\_error(actual, predict)))

print('MAPE:',np.mean(np.abs((actual**-**predict)**/**actual))**\***100)

print('R-Sq:', metrics.r2\_score(actual, predict))

### 1. Decision Tree

In [ ]:



**from** sklearn.tree **import** DecisionTreeRegressor

dt **=** DecisionTreeRegressor(max\_depth **=** 6, random\_state**=**123).fit(train.iloc[:,0:8],train.iloc[:,8])

prediction\_dt **=** dt.predict(test.iloc[:,0:8])

prediction\_dt

In [ ]:



*#error matrix*

performance(test.iloc[:,8],prediction\_dt)

print(' ')

print('Perdicted Vs Actual value: ')

prediction\_dt[6], test.iloc[6,8]

In [ ]:



dt.score(train.iloc[:,0:8],train.iloc[:,8])

In [ ]:



​

### 2. Random Forest

In [ ]:



rf **=** RandomForestRegressor(n\_estimators **=** 700, random\_state **=** 126).fit(train.iloc[:,0:8],train.iloc[:,8])

prediction\_rf **=** rf.predict(test.iloc[:,0:8])

prediction\_rf

In [ ]:



performance(test.iloc[:,8],prediction\_rf)

print(' ')

print('Perdicted Vs Actual value: ')

prediction\_rf[1], test.iloc[1,8]

In [ ]:



rf.score(train.iloc[:,0:8],train.iloc[:,8])

In [ ]:



​

### 3. Linear Regression

In [ ]:



ln\_model **=** LinearRegression().fit(train.iloc[:,0:8],train.iloc[:,8])

prediction\_lr **=** ln\_model.predict(test.iloc[:,0:8])

prediction\_lr

In [ ]:



performance(test.iloc[:,8],prediction\_lr)

print(' ')

print('Perdicted Vs Actual value: ')

prediction\_lr[1], test.iloc[1,8]

In [ ]:



ln\_model.score(train.iloc[:,0:8],train.iloc[:,8])

In [ ]:



*#Ploting to understand the spread of predicted data.*

​

plt.figure(figsize**=**(14,4))

​

plt.subplot(1,3,1)

plt.title('Decision Tree predictions')

plt.scatter(test.iloc[:,8] , prediction\_dt)

plt.xlabel('Actual values')

plt.ylabel('Predicted values')

​

plt.subplot(1,3,2)

plt.title('Random Forest predictions')

plt.scatter(test.iloc[:,8] , prediction\_rf)

plt.xlabel('Actual values')

plt.ylabel('Predicted values')

​

plt.subplot(1,3,3)

plt.title('Linear Regression Predictions')

plt.scatter(test.iloc[:,8] , prediction\_lr)

plt.xlabel('Actual values')

plt.ylabel('Predicted values')

​

plt.tight\_layout()