Employee Absenteeism

Project Report

Sharang Shrivastava

28th February , 2019

**Contents**

1. Overview

2. Data Summary

3. Exploratory Data Analysis

3.1 Missing Values Analysis

3.2 Feature Engineering

3.3 Outliers Analysis

4. Correlation Analysis

5. Preparing data for modelling

6. Model building

6.1 Linear regression Model

6.2 Random Forest Model

6.3 GradientBoost Model

6.4 XGBoost Model

7. Model Performance Comparison

8. Model Predictions

**1.Overview**

Employee Absenteeism is the absence of an employee from work. Its a major problem faced by almost all employers of today. Employees are absent from work and thus the work suffers. Absenteeism of employees from work leads to back logs, piling of work and thus work delay. There are various laws been enacted for safeguarding the interest of both Employers and Employees but they too have various constraints.

Absenteeism is of two types -

* **Innocent absenteeism -** Is one in which the employee is absent from work due to genuine cause or reason. It may be due to his illness or personal family problem or any other real reason
* **Culpable Absenteeism -** is one in which a person is absent from work without any genuine reason or cause. He may be pretending to be ill or just wanted a holiday and stay at home. The employers have got every right to enquire as to why an employee is absent from work. If an employee is absent because of illness he should be able to produce a doctor's letter as and when demanded.

As per the survey conducted by US-based human capital services provider Careerbuilder, 30% of workers have called in sick when not actually ill in the past year. The sick days, legitimate or otherwise, also become more frequent around the winter holidays, with nearly one-third of employers reporting more employees call in sick during the holiday season, the survey found. At the same time, 29 per cent of employers have checked up on an employee to verify that the illness is legitimate, usually by requiring a doctor's note or calling the employee later in the day.

**2. Data Summary**

As a first step let’s do three simple steps on the dataset

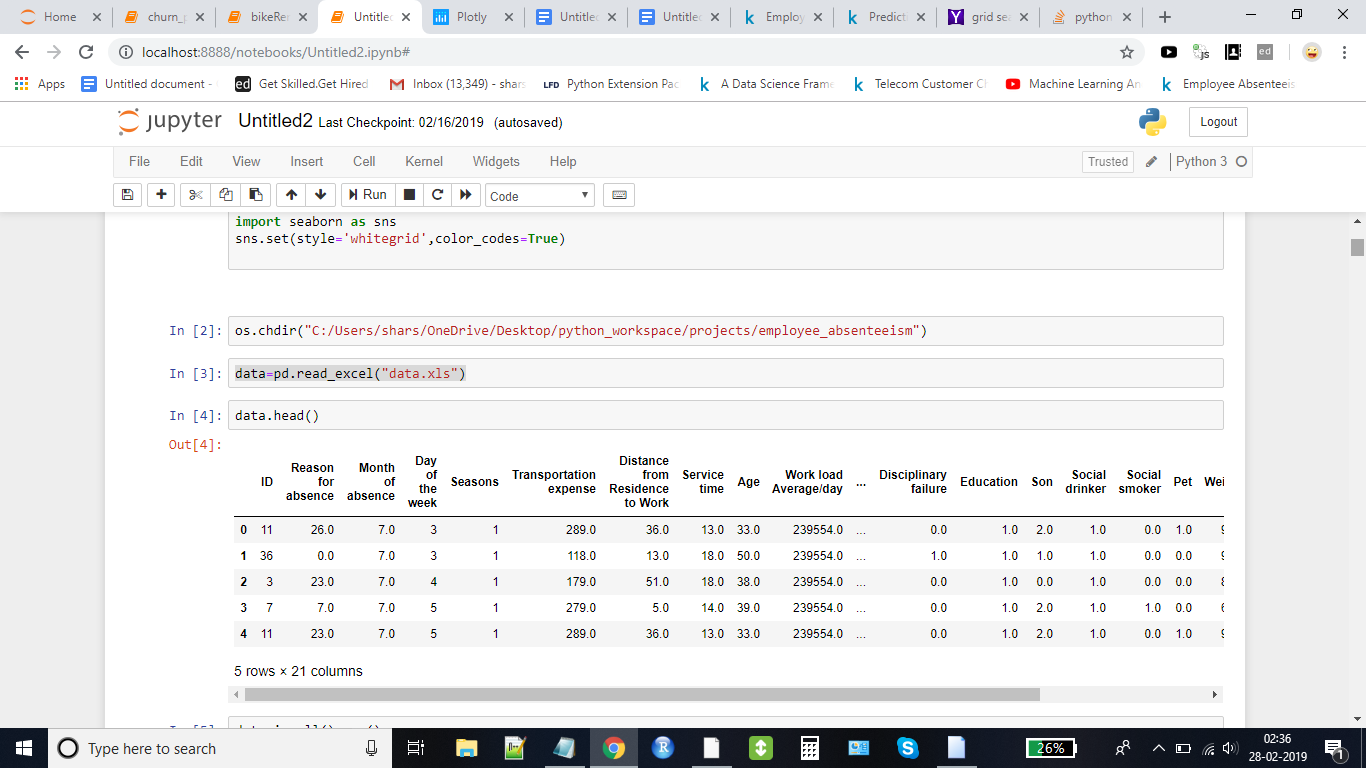
* Size of the dataset
* Get a glimpse of data by printing few rows of it.
* What type of variables contribute our data

**Shape of data** : 740 rows , 21 columns(740, 21)

(740, 21

#### **Sample Of First Few Rows**

**data.head()**



**Data Analysis**

***Check the properties of the data***

RangeIndex: 740 entries, 0 to 739  
Data columns (total 21 columns):  
ID 740 non-null int64  
Reason for absence 737 non-null float64  
Month of absence 739 non-null float64  
Day of the week 740 non-null int64  
Seasons 740 non-null int64  
Transportation expense 733 non-null float64  
Distance from Residence to Work 737 non-null float64  
Service time 737 non-null float64  
Age 737 non-null float64  
Work load Average/day 730 non-null float64  
Hit target 734 non-null float64  
Disciplinary failure 734 non-null float64  
Education 730 non-null float64  
Son 734 non-null float64  
Social drinker 737 non-null float64  
Social smoker 736 non-null float64  
Pet 738 non-null float64  
Weight 739 non-null float64  
Height 726 non-null float64  
Body mass index 709 non-null float64  
Absenteeism time in hours 718 non-null float64  
dtypes: float64(18), int64(3)

*what we can infer:*  
*> There are null values in the dataset*  
*> The data types are int and float*

***Check for any invalid data inputs***

From above observations data does not seem to have any invalid data types to be handled  
 However feature 'Absence\_Month' have an invalid value 0. Let’s drop it.  
 ALso, as we can see, 'Absent\_Hours' are 0 in some places.  
 This could be result of cancelled or withdrawn leaves. We will drop all these rows.

**3. Exploratory Data Analysis(EDA)**

**3.1 Missing Value Analysis**

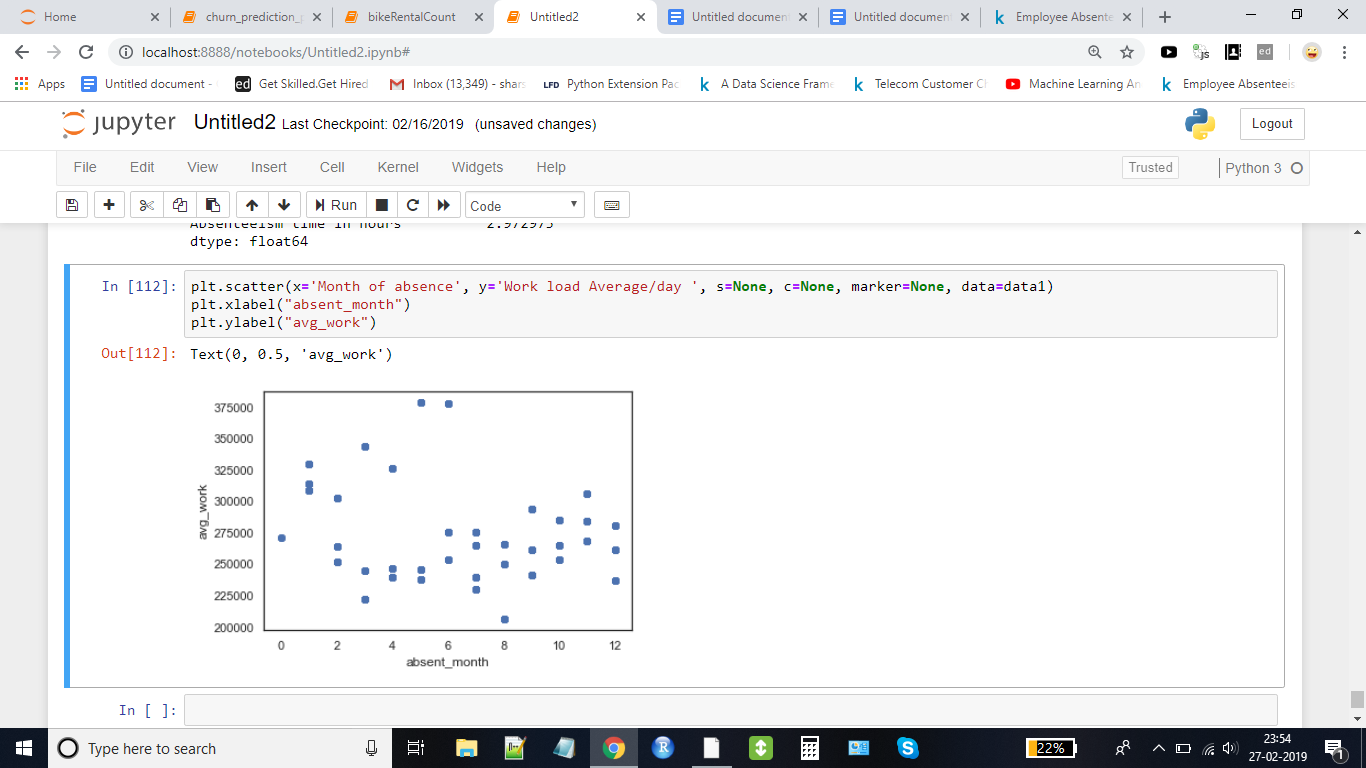
*Calculating % of nulls*

ID 0.000000  
Reason for absence 0.405405  
Month of absence 0.135135  
Day of the week 0.000000  
Seasons 0.000000  
Transportation expense 0.945946  
Distance from Residence to Work 0.405405  
Service time 0.405405  
Age 0.405405  
Work load Average/day 1.351351  
Hit target 0.810811  
Disciplinary failure 0.810811  
Education 1.351351  
Son 0.810811  
Social drinker 0.405405  
Social smoker 0.540541  
Pet 0.270270  
Weight 0.135135  
Height 1.891892  
Body mass index 4.189189  
Absenteeism time in hours 2.972973  
dtype: float64

***what we can infer:***

->There are null values in almost all the columns of the dataset, although in small amount.  
-> We'll drop all the null value rows for target variable and   
-> We'll will impute null values for all other features.

Replace missing of any any employee with information of same employee from other instances  
Example if 'Age' of employee 1 is missing, then impute it with 'Age' from other instance of employee 1



*From above, we can deduce that 'Average\_Workload' is distributed mostly by month.*  
*So, let's impute missing 'Average\_Workload' by mode of that month*

now only absent reason and hit target are left

ID 0  
absent\_reason 3  
absent\_month 0  
day 0  
Seasons 0  
transport\_expense 0  
dist\_work 0  
serv\_time 0  
age 0  
avg\_work 0  
hit\_targ 6  
displn\_failure 0  
education 0  
son 0  
drinker 0  
smoker 0  
pet 0  
weight 0  
height 0  
bmi 0  
absent\_hours 22

We will impute hit\_target values by grouping by season month and day of the week and delete those rows which have null values in absent reason as these rows are only 3 in numbers. We will fill NA values in target with 0’s.

***Missing Value handling ENDS here***

**3.2 Feature extraction** .

Extract any new features from existing features if required  
**Converting data to proper formats**

features like 'Absence\_Month','Education',etc are categories here. Lets convert these variables to categories.

ID 677 non-null int64

Absence\_Reason 677 non-null category

Absence\_Month 677 non-null category

Absence\_Day 677 non-null category

Seasons 677 non-null category

Transportation\_Expense 677 non-null float64

Work\_Distance 677 non-null float64

Service\_Time 677 non-null float64

Age 677 non-null float64

Average\_Workload 677 non-null float64

Hit\_Target 677 non-null float64

Disciplinary\_Failure 677 non-null category

Education 677 non-null category

Son 677 non-null category

Drinker 677 non-null category

Smoker 677 non-null category

Pet 677 non-null category

Weight 677 non-null float64

Height 677 non-null float64

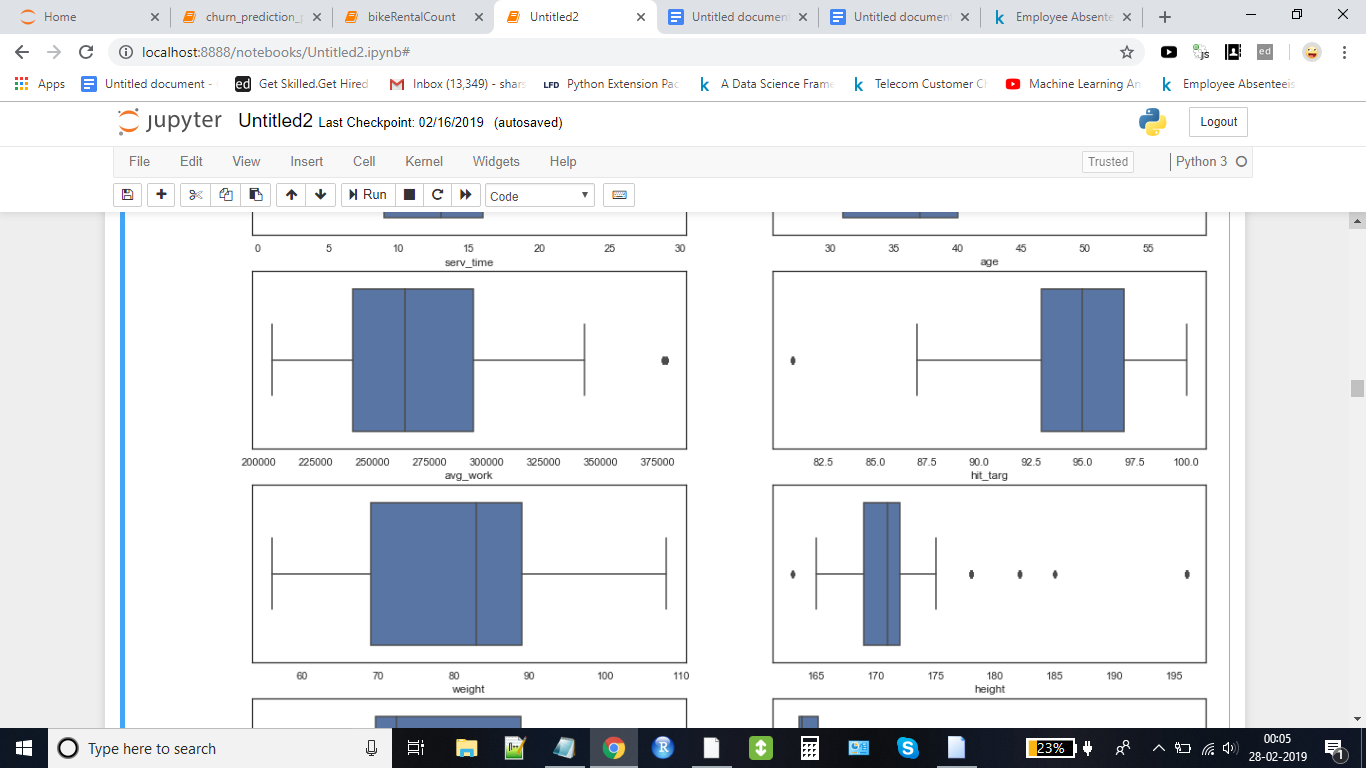
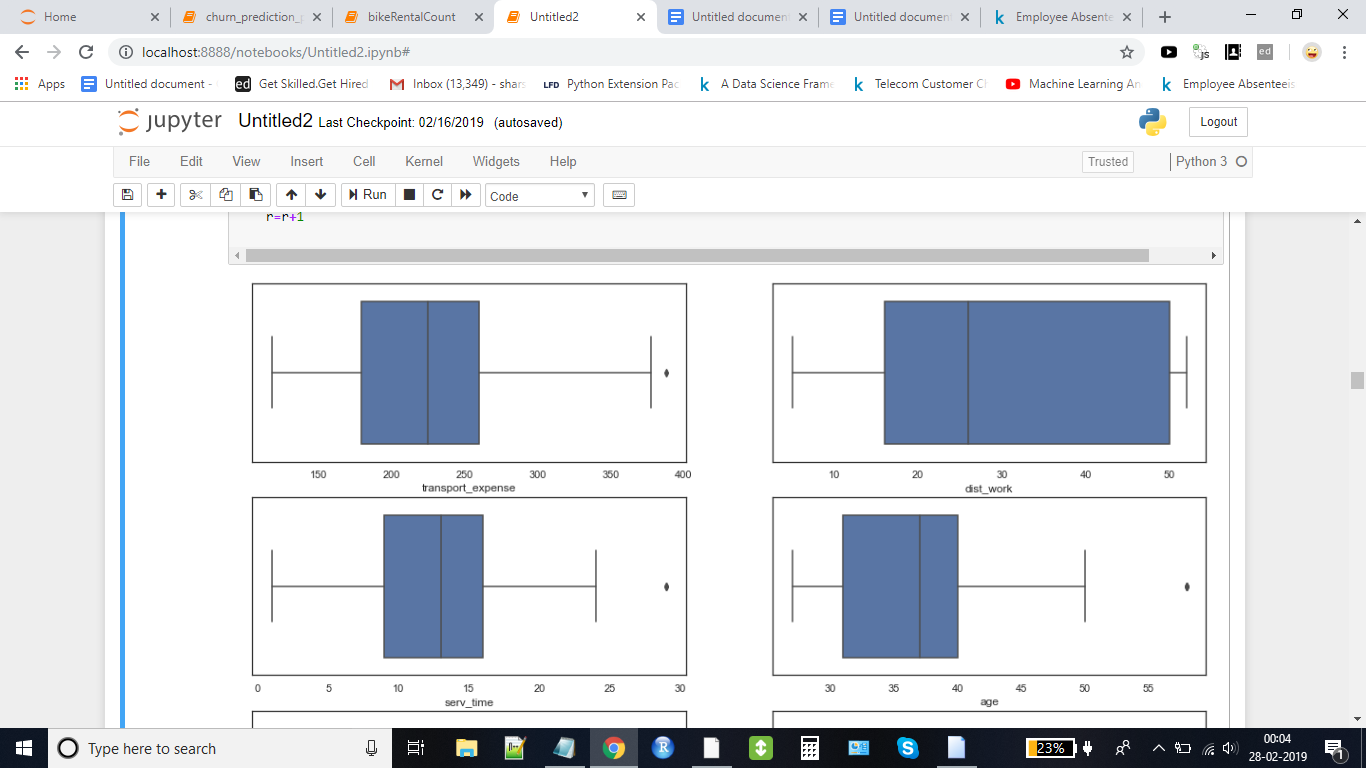
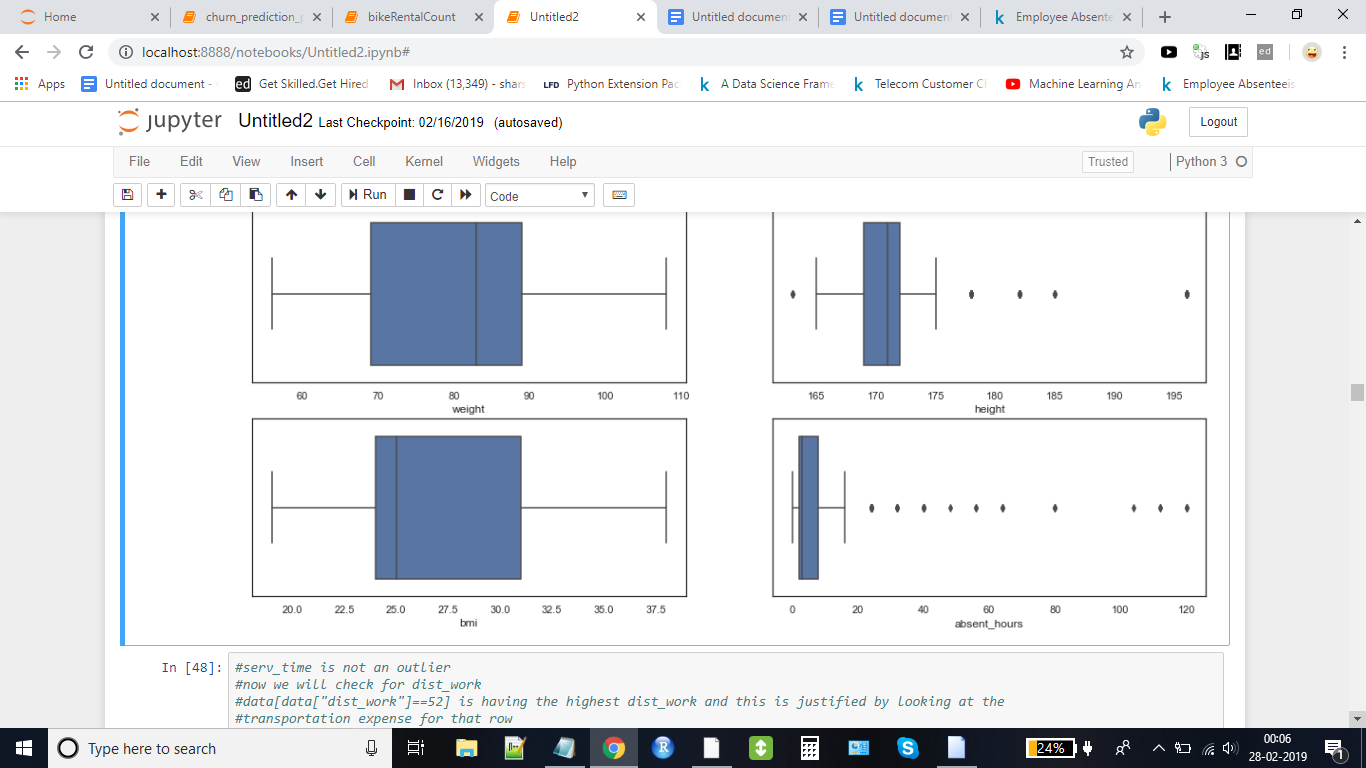
BMI 677 non-null float64

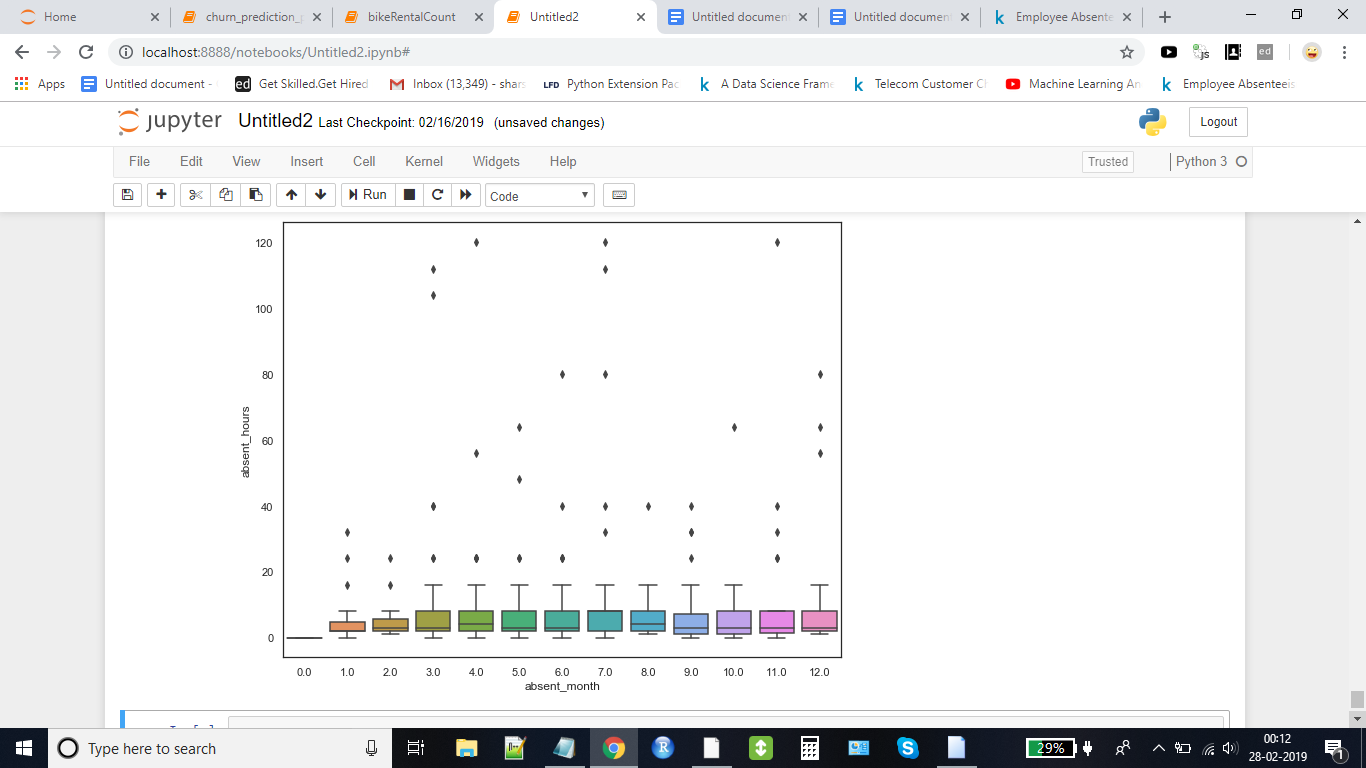
Absent\_Hours 677 non-null float64

dtypes: category(10), float64(10), int64(1)

Here we do not need any feature extraction.  
However, before feeding to model, we might need to aggregate the data.

**3.3 Outlier Analysis**





serv\_time(Service\_time) is not an outlier

now we will check for dist\_work(distance from work)

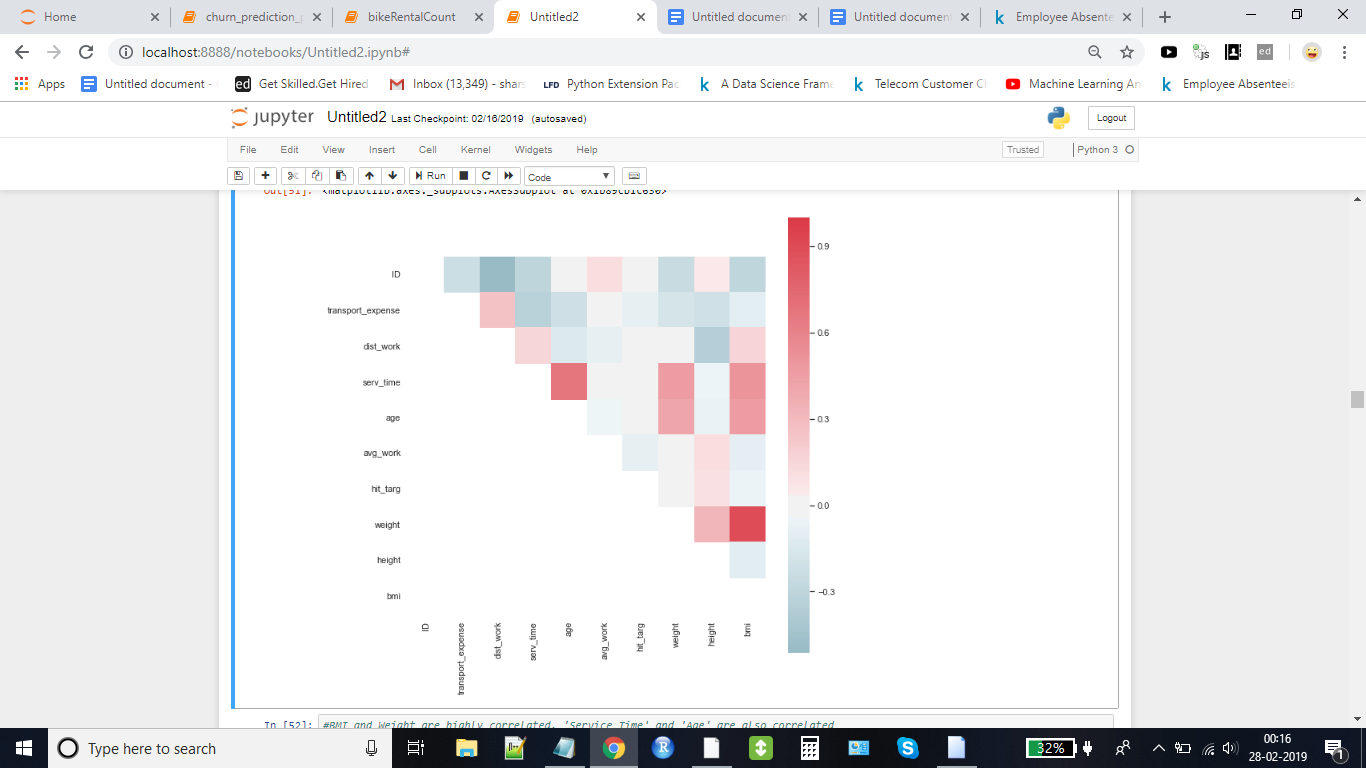
data[data["dist\_work"]==52] is having the highest dist\_work and this is justified by looking at the transportation expense for that row

so dist\_work is also not an outlier

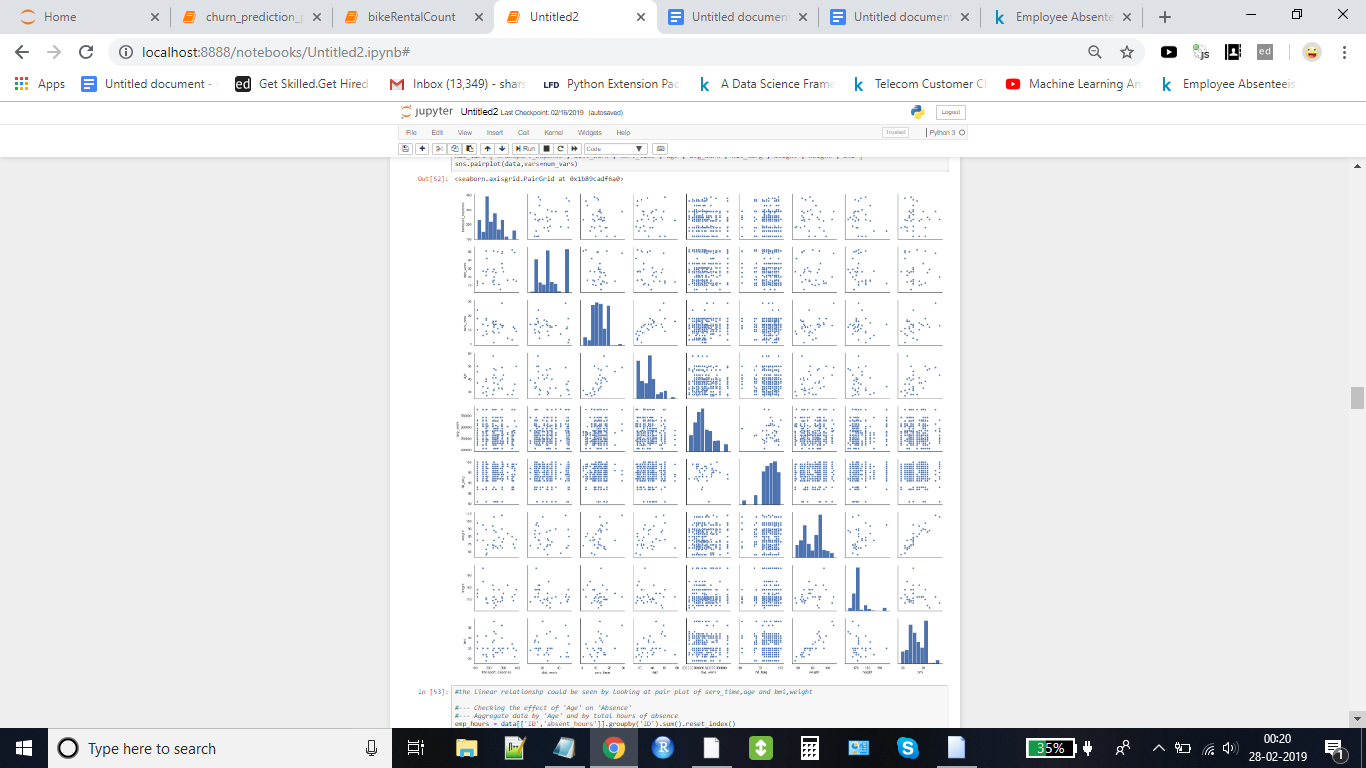
***what we can infer from above boxplots:***  
-> Target feature 'Absent\_hours', has many outliers. It needs to be handled( will handle it after exploratory analysis)  
-> Not many outliers in independent features. Data seems balanced.

**4. Correlation Analysis**

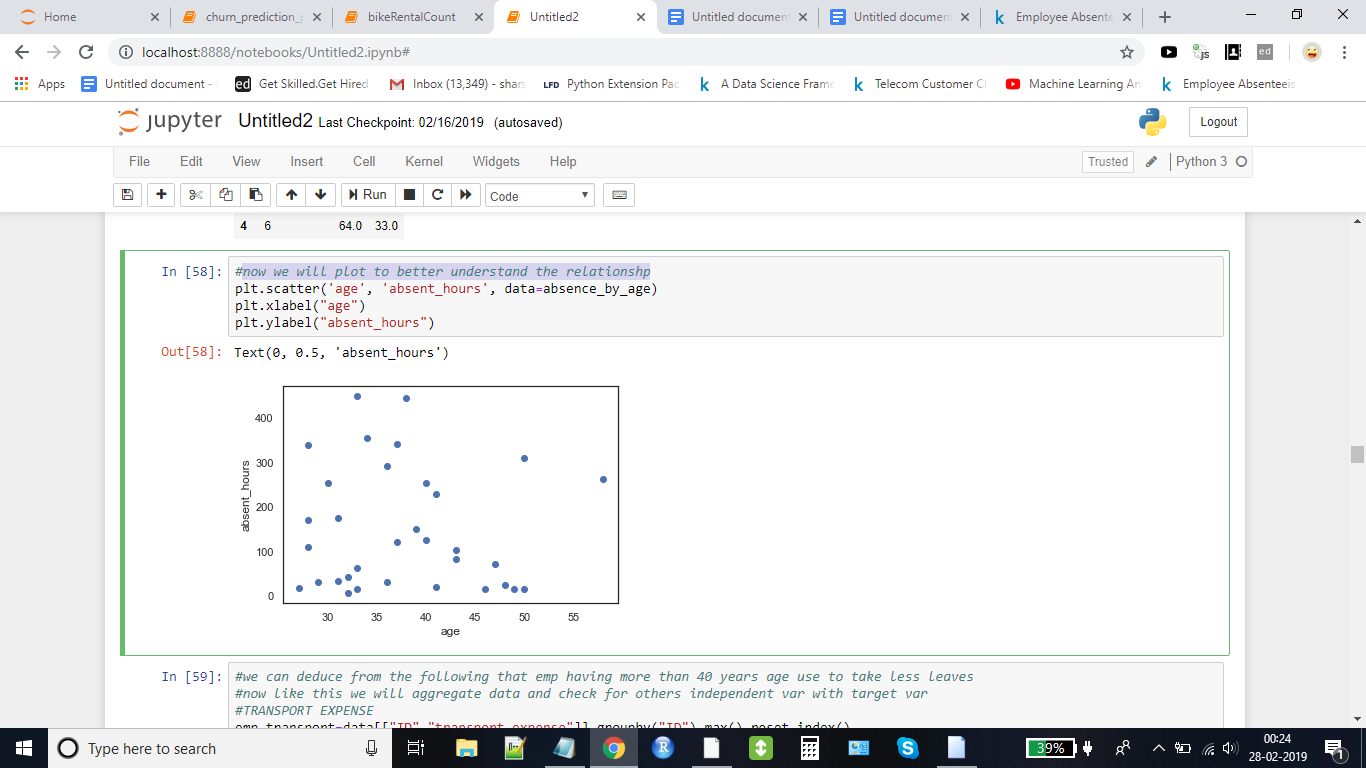
*correlation between the independent continuous features with target variable*



This shows that there is multicollinearity in the dataset. BMI and Weight are highly correlated. 'Service\_Time' and 'Age' are also correlated  
Will have to deal with multicollinearity by removing few features from the dataset.

**The linear relationship could be seen by looking at pair plot of (serv\_time,age) and (bmi,weight).**

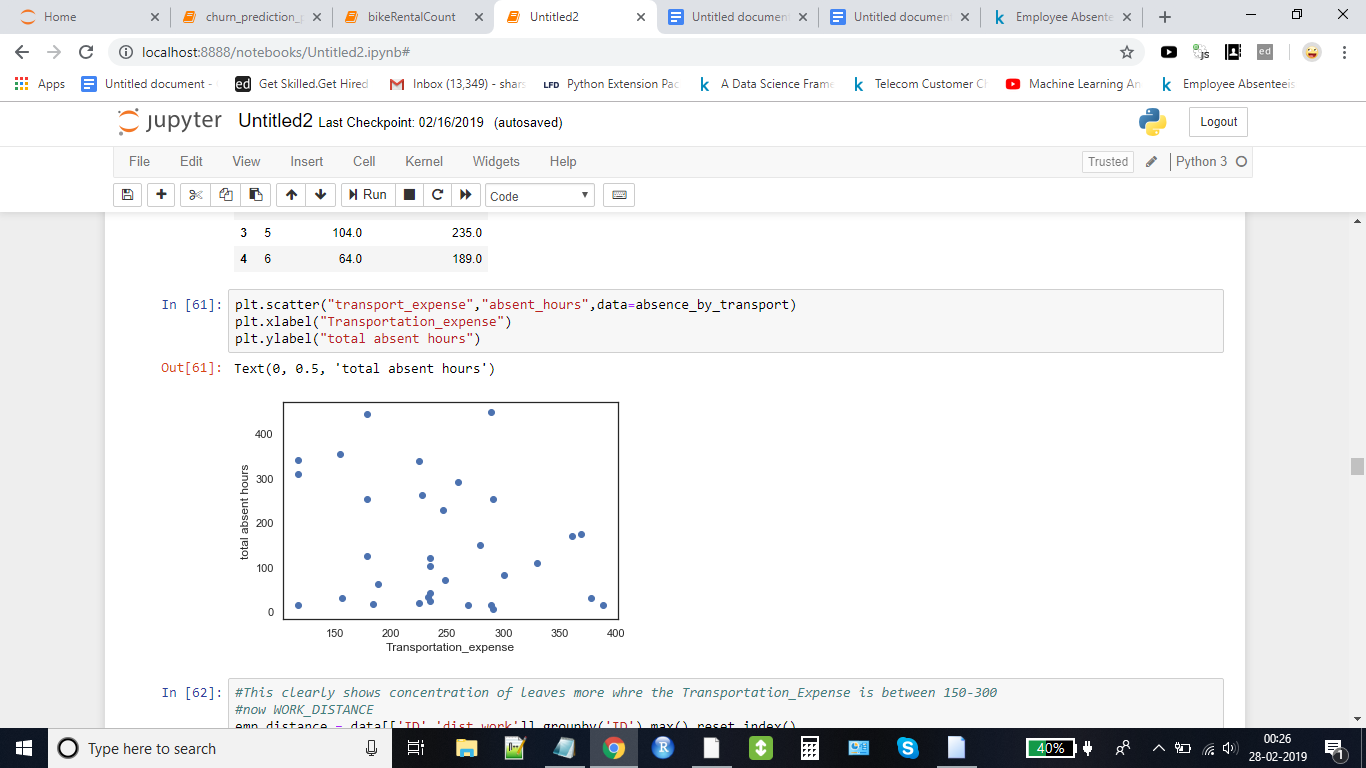
**now we will look at the plot of age and absent hours to better understand the relationship**



we can deduce from the following that employee having more than 40 years age use to take less leaves

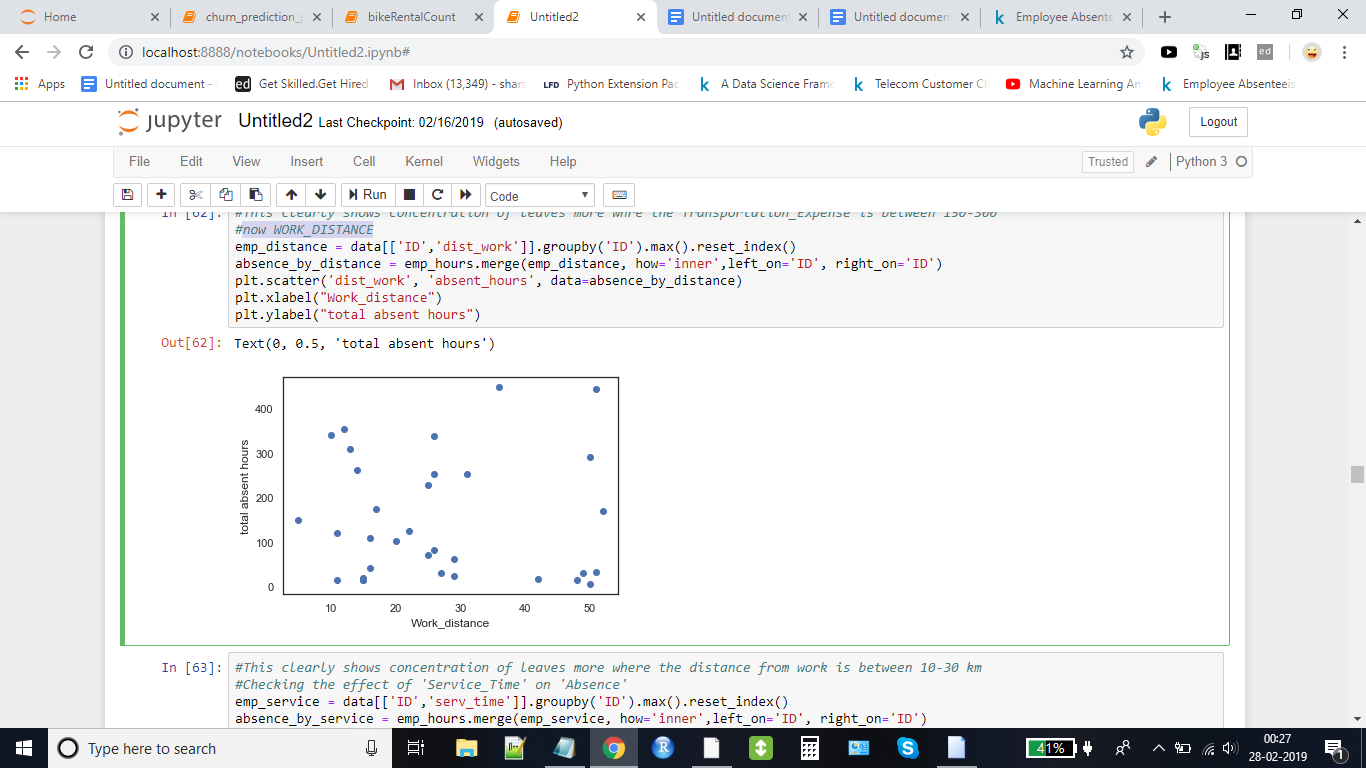
now like this we will aggregate data and check for others independent var with target var

**Transport expense with absent hours plot**



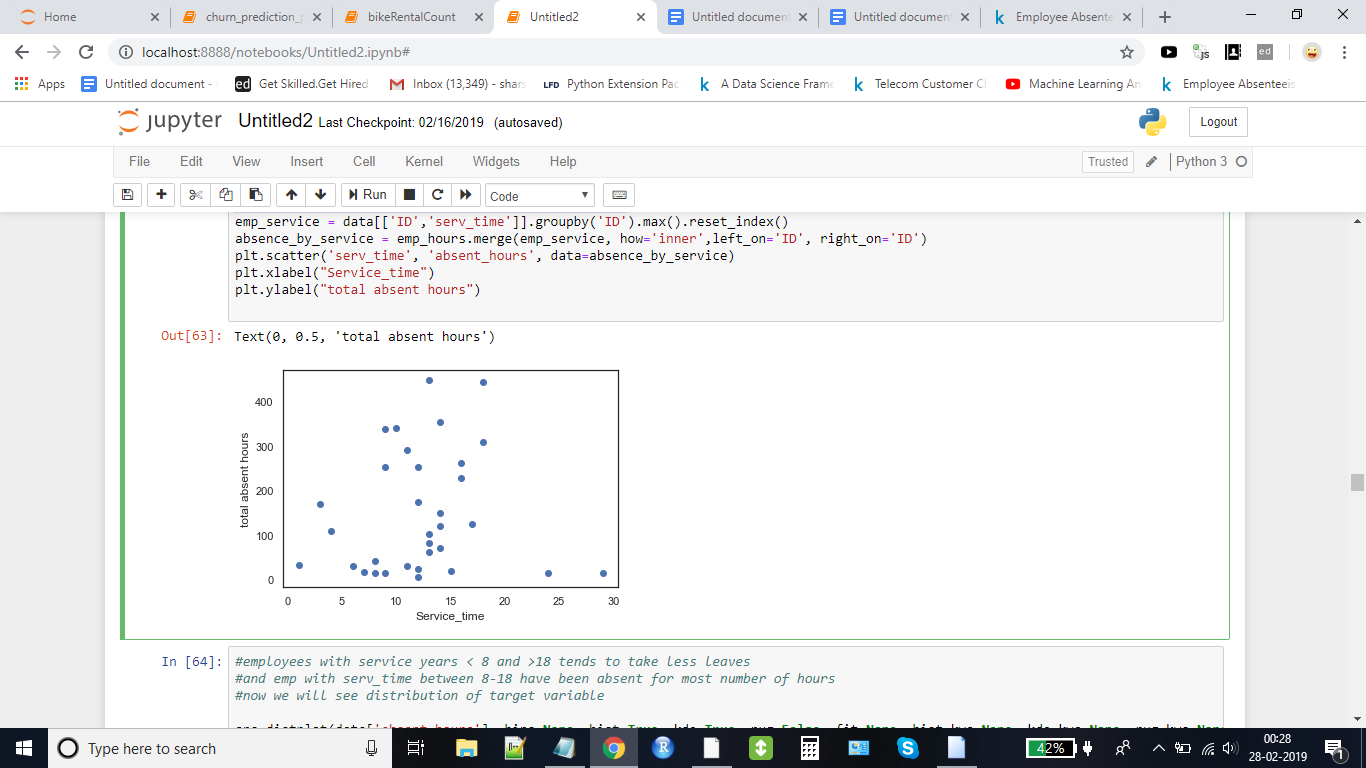
This clearly shows concentration of leaves more where the Transportation\_Expense is between 150-300

**now Work\_distance and absent\_hours plot**



This clearly shows concentration of leaves more where the distance from work is between 10-30 km

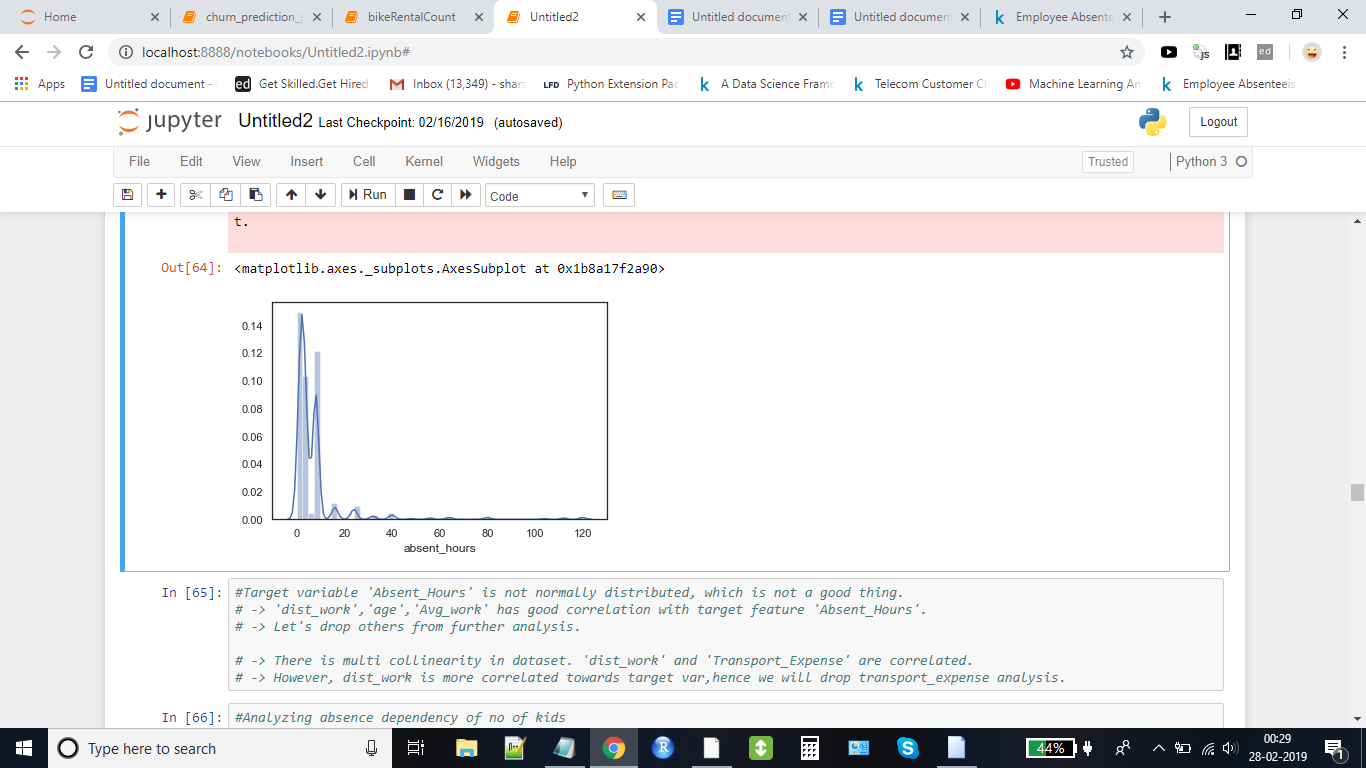
**Checking the effect of 'Service\_Time' on 'Absence\_hours**



employees with service years < 8 and >18 tends to take less leaves

and employees with serv\_time between 8-18 have been absent for most number of hours

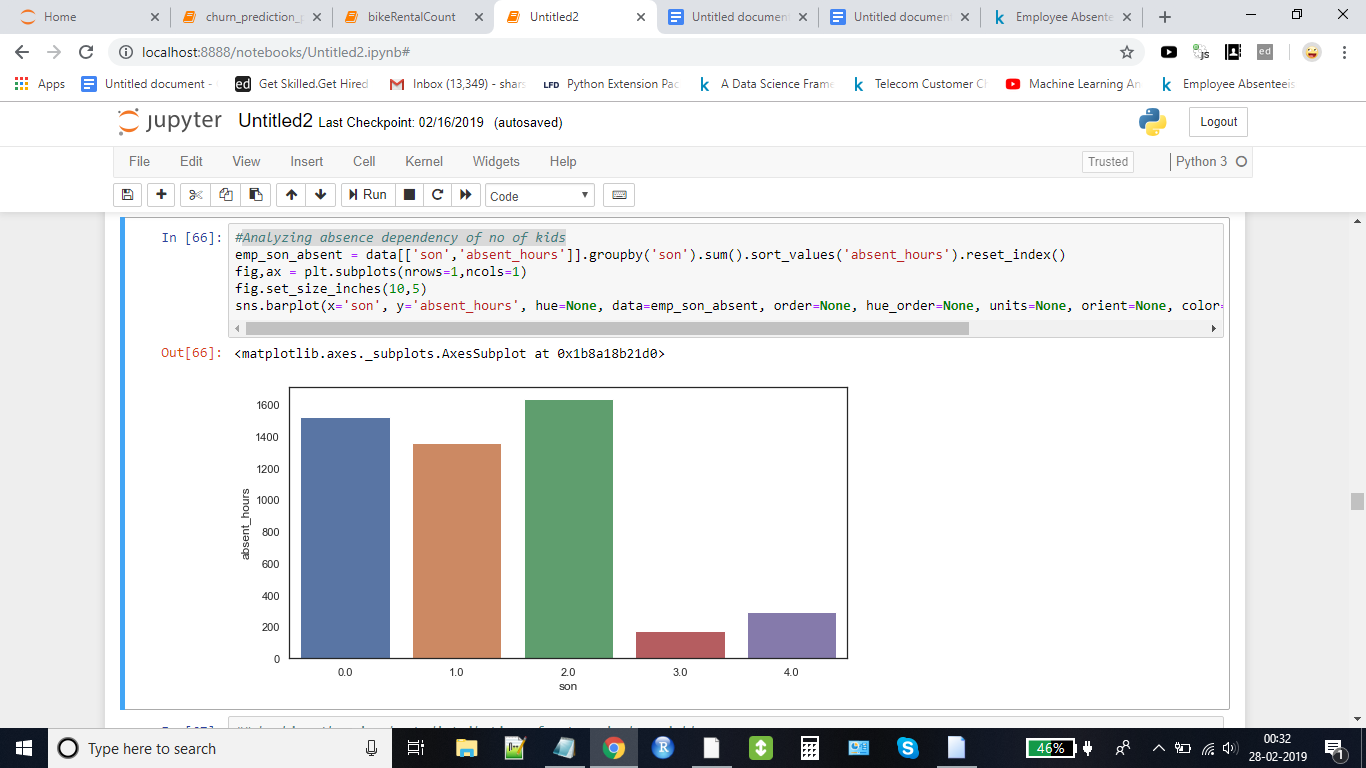
**Now we will see distribution of target variable.**



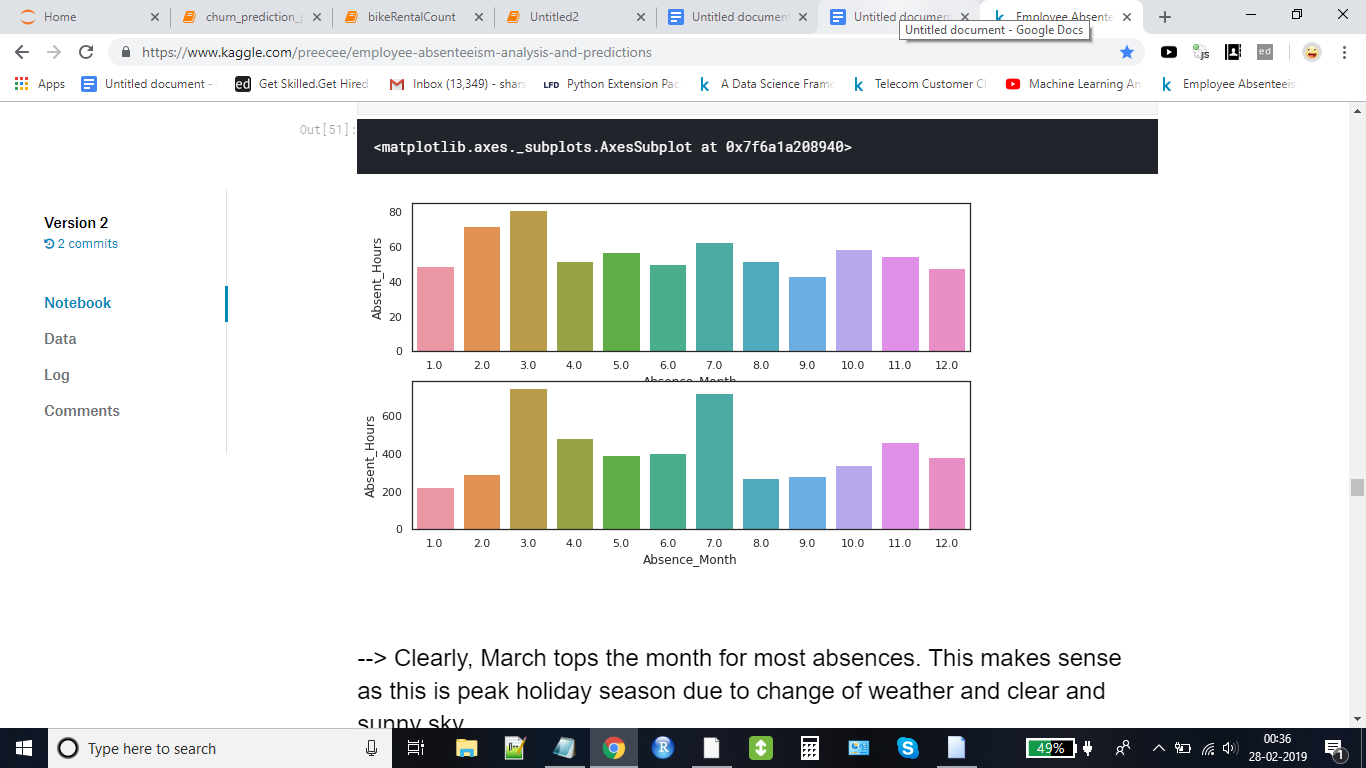
***What we can infer from above analysis of continuous variables:***  
-> Target variable 'Absent\_Hours' is not normally distributed, which is not a good thing.   
-> We have to look in to this, before feeding the data to model.  
  
-> 'Work\_Distance','Age','Average\_Workload' has good correlation with target feature 'Absent\_Hours'.  
> Let's drop others from further analysis.  
  
-> There is multicollinearity in dataset. 'Work\_Distance' and 'Transportation\_Expense' are correlated.

-> However, dist\_work is more correlated towards target var,hence we will drop transport\_expense analysis.

**Analyzing absence dependency of no of kids**

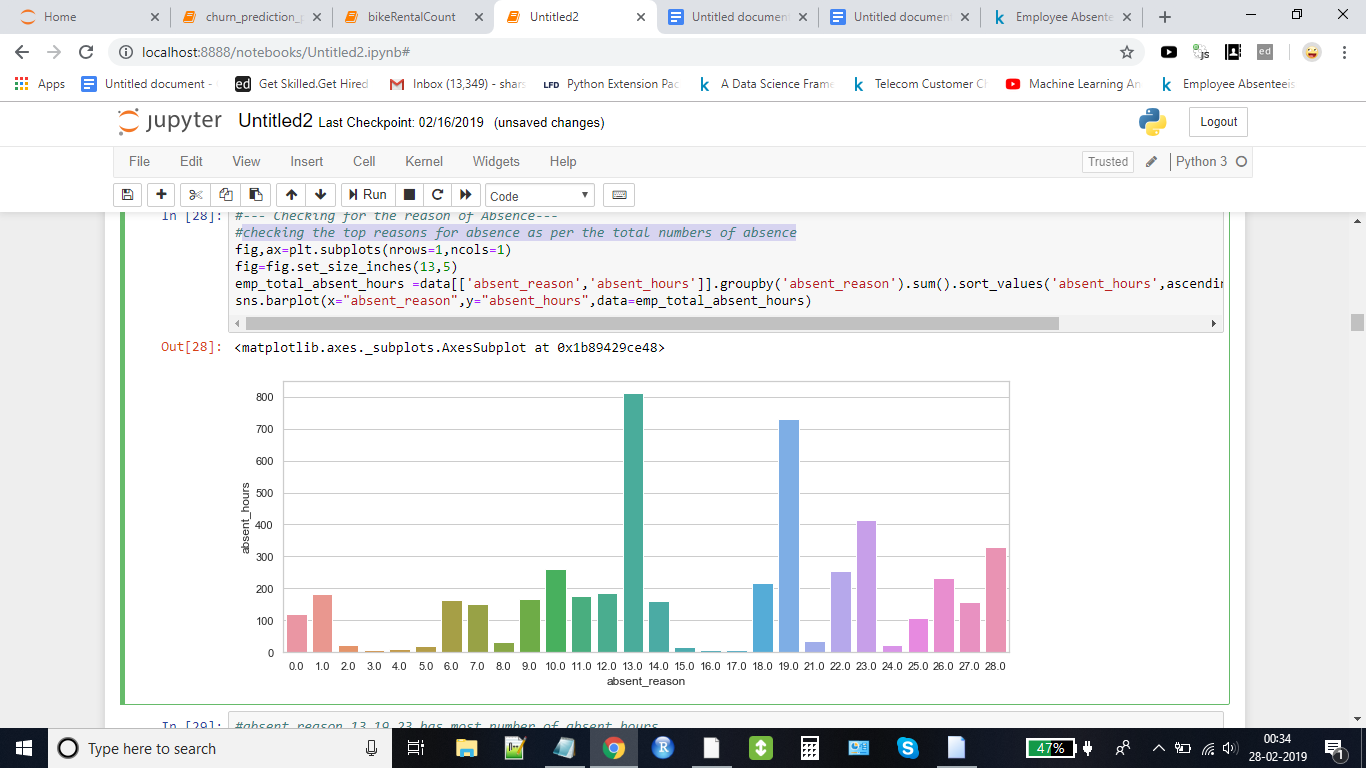


*Clearly, employee with 3-4 kids tend to take less hours of absence*

***Analyzing absence dependency of month of year***

*March and july month clearly tops the list having most numbers of absent hours*

***Analyzing reason of absence with respect to sum of absent hours for that reason***



# **Longest hours of absences for reason 13,19,23,28**

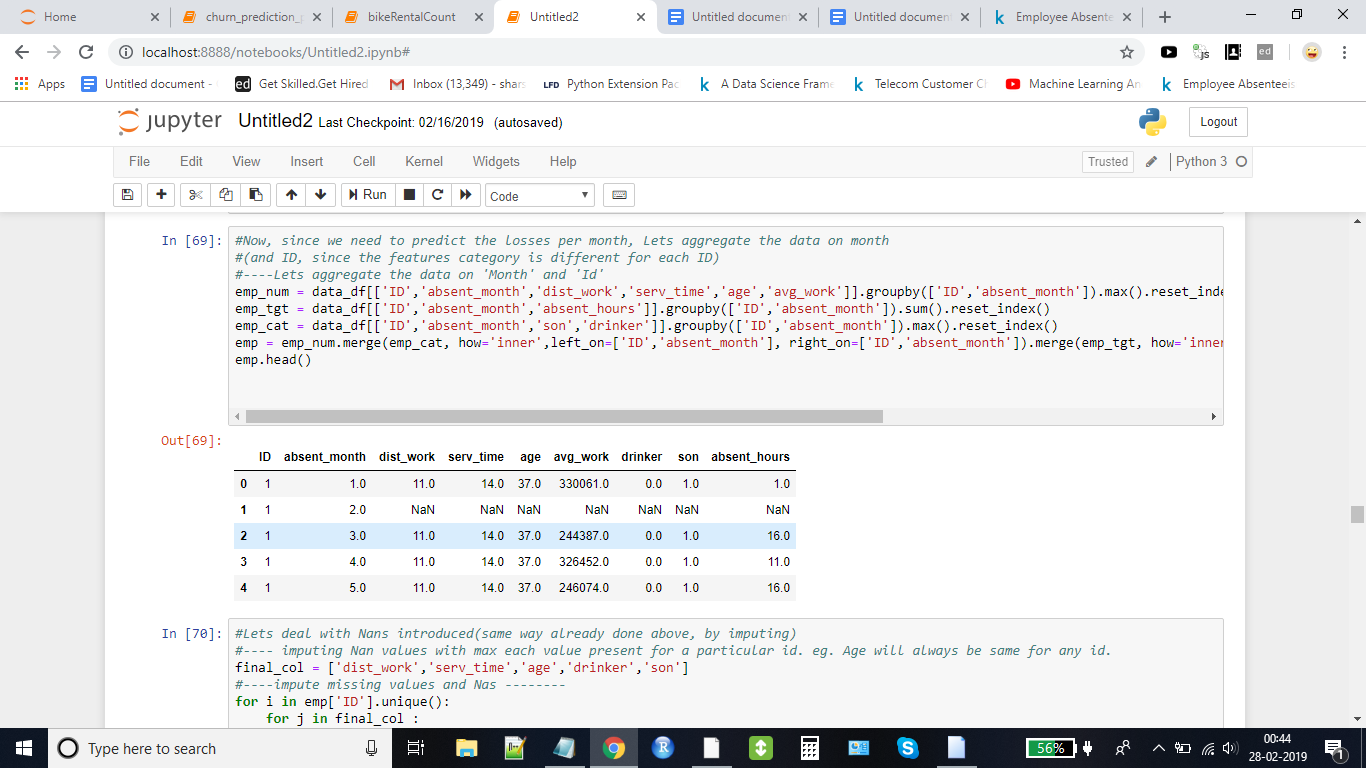
# #23 - medical consultation

# #28 - dental consultation #13 - Diseases of the musculoskeletal system and connective tissue #19 - Injury, poisoning and certain other consequences of external causes

# ***Seems like employee takes most absences for medical consultations/dental consultation and physiotherapy.These hours can be reduced by setting up a medical consultation/dental consultation/physiotherapy booth(with visiting doctors may be) at office/facility.In long term, introducing exercise/yoga sessions in office once/twice a week will help reduce physiotherapy issues***

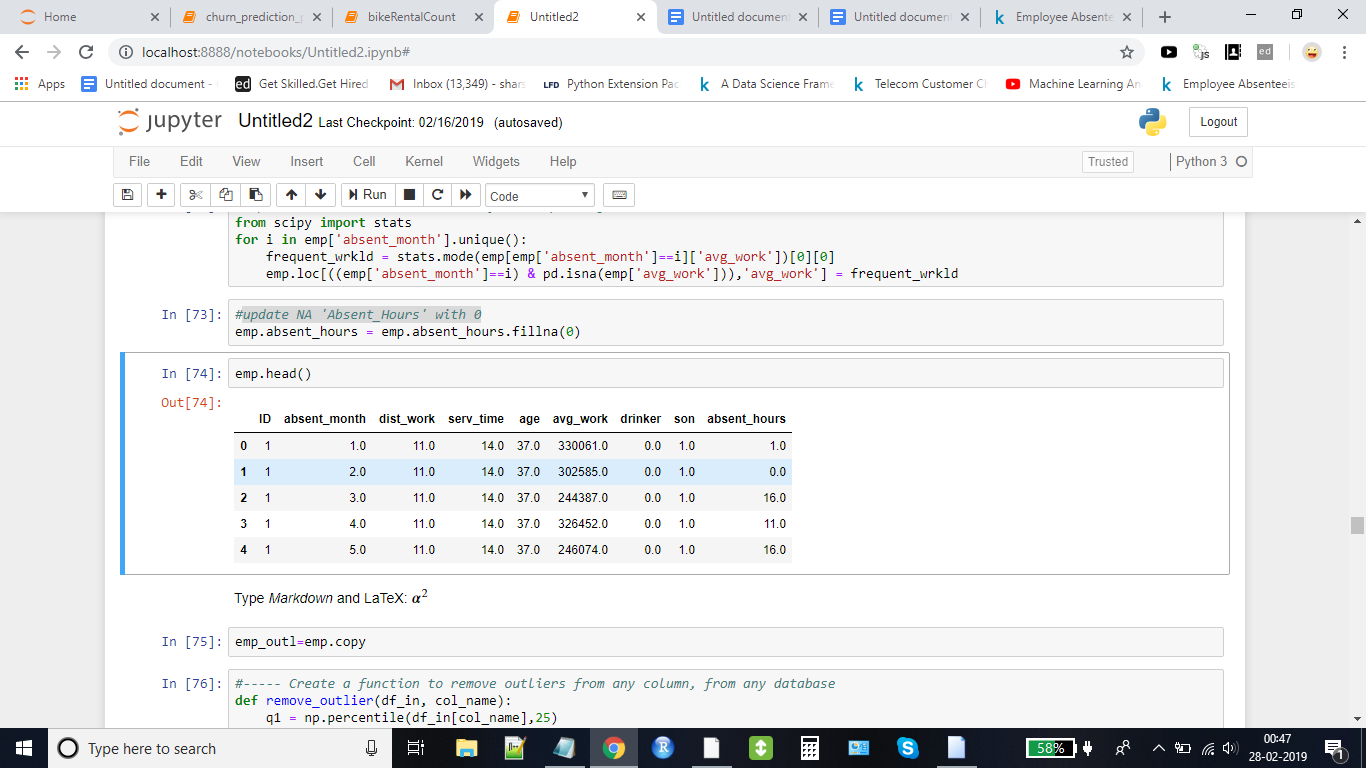
**5. Preparing data for modelling**

# Now, since we need to predict the losses per month, Lets aggregate the data on month and ID before feeding the data to model.

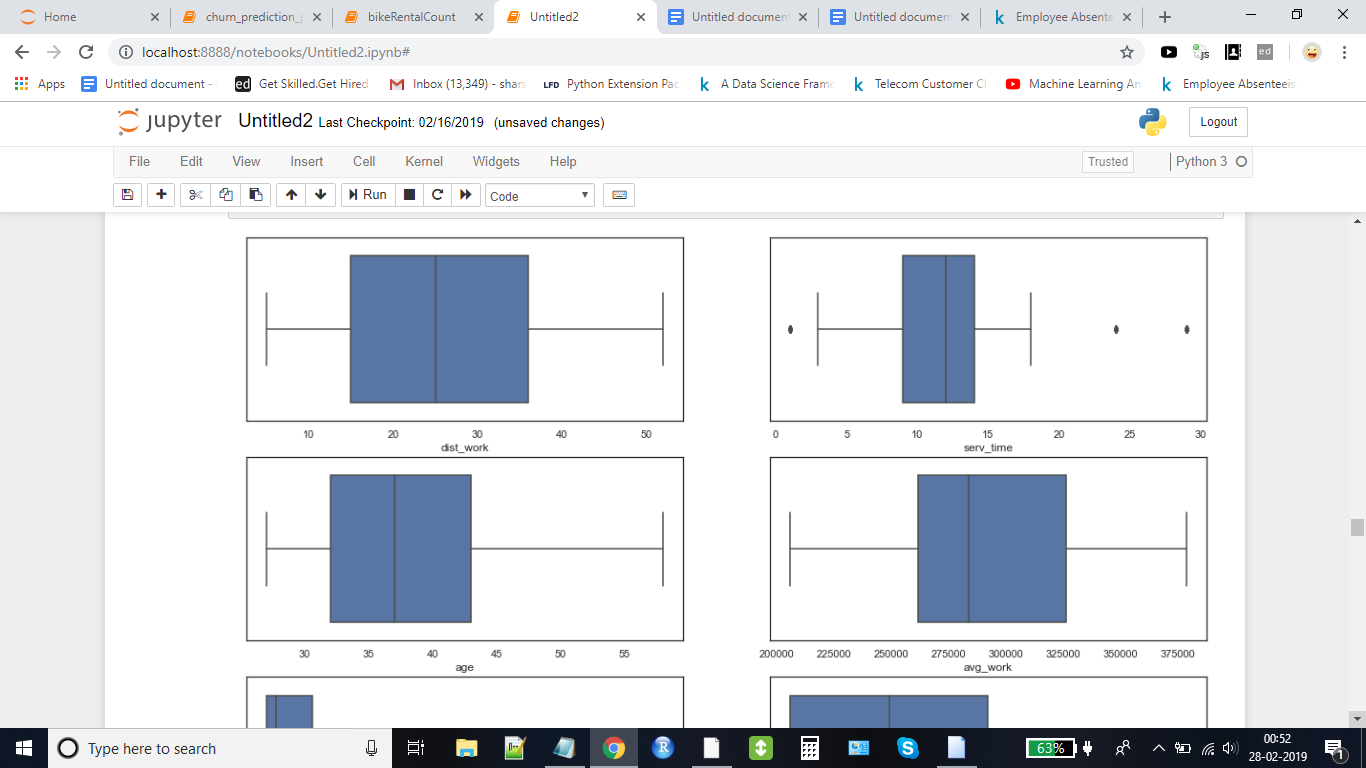
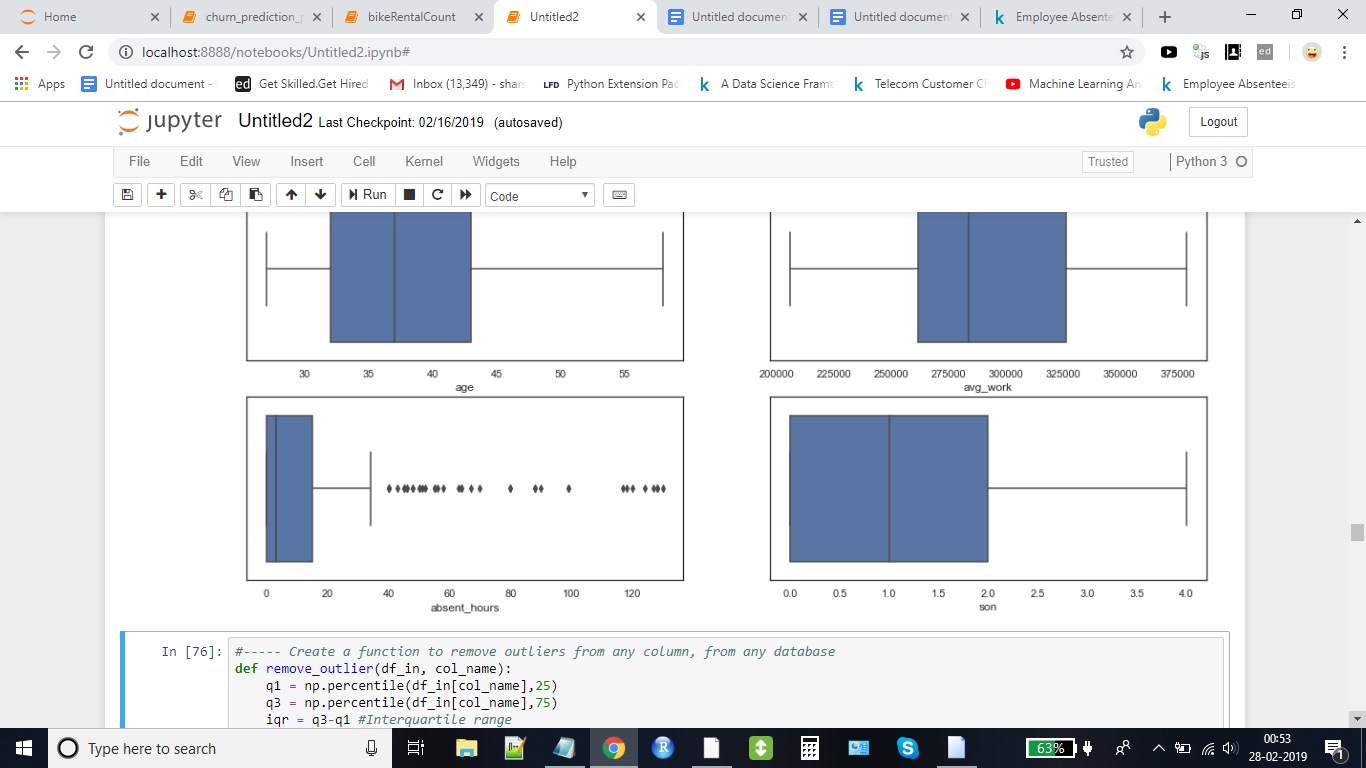


Lets deal with Nans introduced(same way already done above, by imputing)  
We will impute Nan values with max each value present for a particular id. eg. Age will always be same for any id.   
And update workload with the mode of corresponding month's workload, as we did earlier.

Now the shape of the data got reduced to (396 rows, 9 columns) after aggregating the data on basis of month and ID.



***Let’s check for any outliers in the aggregated data***

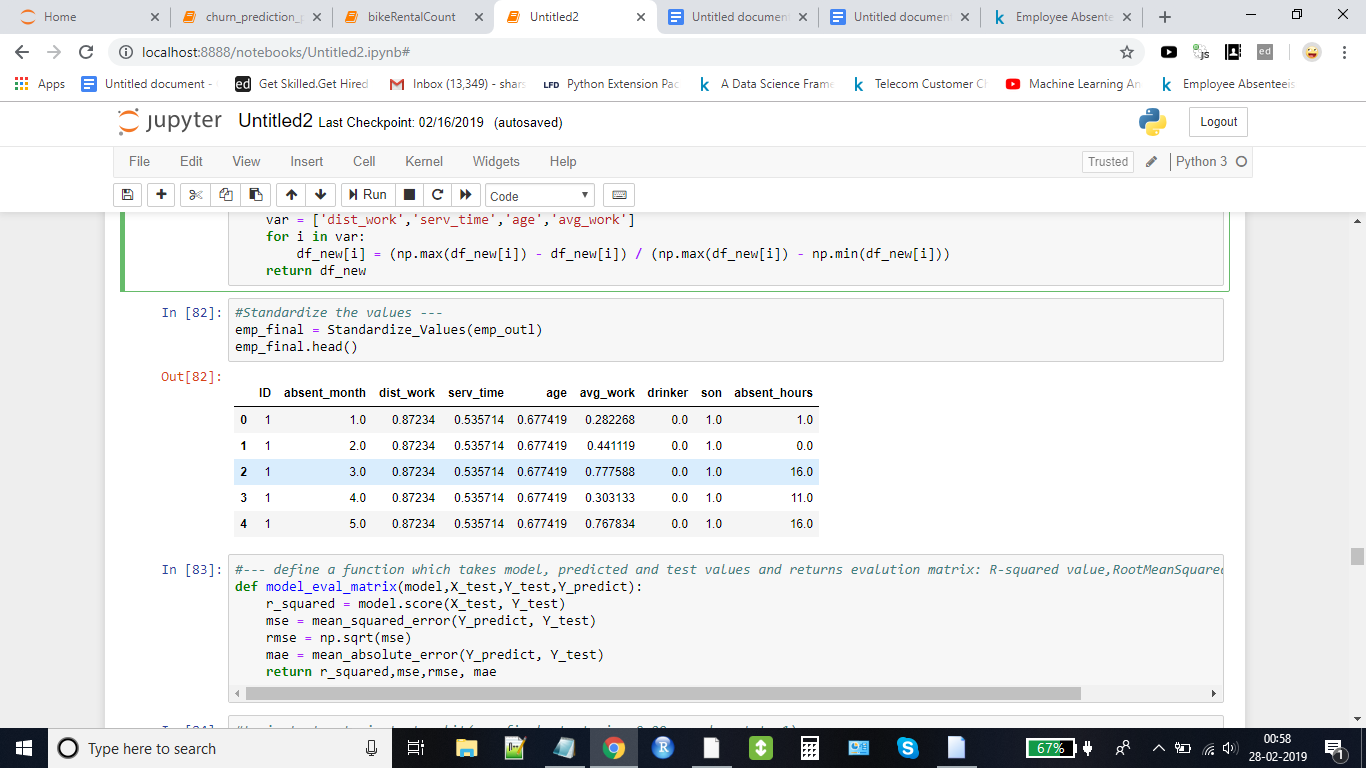


Clearly, 'Absent\_Hours' has so many outliers, this will affect model. So, extreme outliers needs to be removed to make the model more generic. We are not removing outliers in service time, since the input data for 2011 is going to be same as 2010(except 'Age' and 'Service Time')

**Standardization of data**

As we can clearly see that the dataset has different features of different range/scale.

Lets standardise the range/scale for better performance of model



*Now we are done preparing data for modelling, we will start building our models on top of it*

**6. Model building**

**6.1 Linear regression Model**

lrm\_regressor = LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None, normalize=True)

**6.2 Random Forest Model (Ensemble method using Bagging technique)**

forest\_reg = RandomForestRegressor(n\_estimators=2000, criterion='mse', max\_depth=10, min\_samples\_split=5, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_features='auto', max\_leaf\_nodes=20, min\_impurity\_decrease=0.00, min\_impurity\_split=None, bootstrap=True, oob\_score=False, n\_jobs=1, random\_state=1, verbose=0, warm\_start=False)

**6.3 GradientBoost Model (Ensemble method using Boosting technique)**

**# without parameter hypertuning**

gb\_reg = GradientBoostingRegressor(ranxgb = xgboost.XGBRegressor(n\_estimators=100, learning\_rate=0.08, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=7)

dom\_state=1)

**# Following model is with parameter hypertuning**

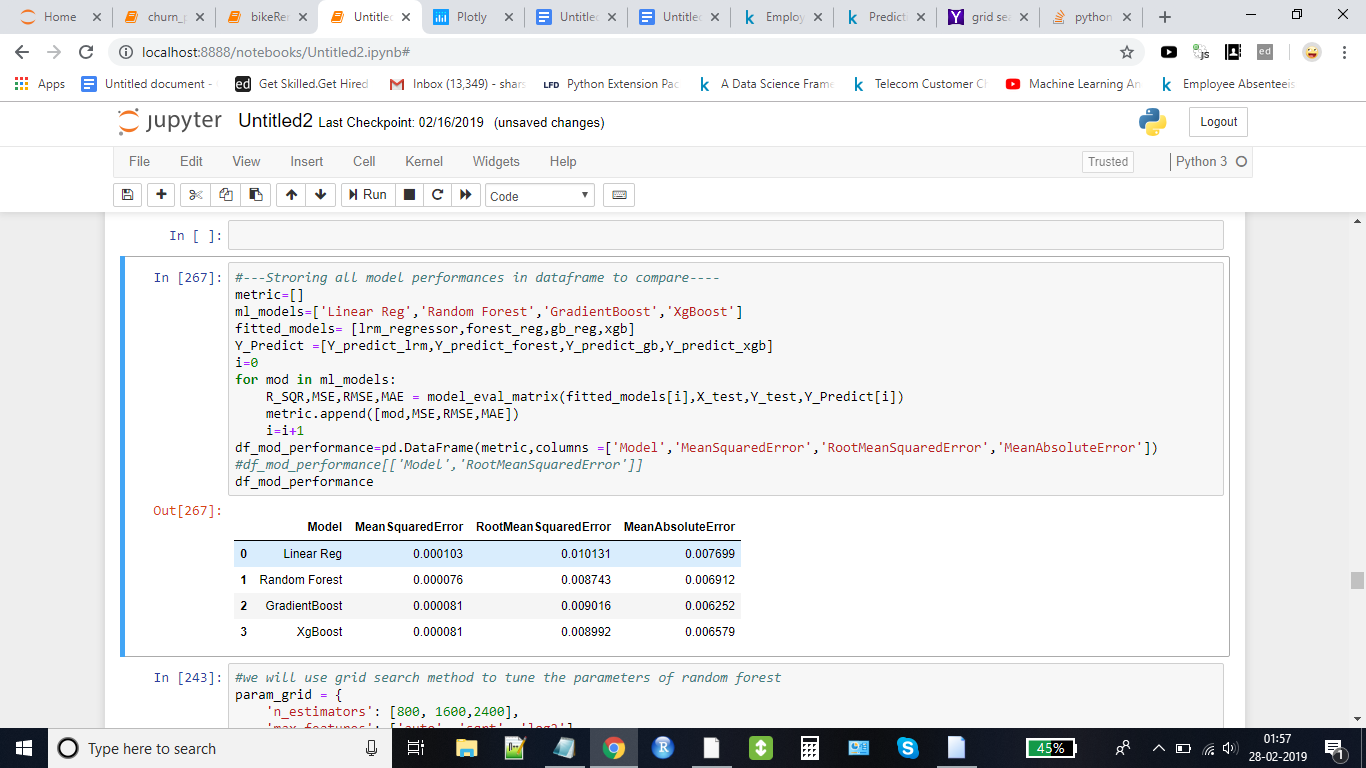
gb\_reg = GradientBoostingRegressor(loss='ls', learning\_rate=0.2, n\_estimators=2000, subsample=1.0, criterion='friedman\_mse', min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_depth=3, min\_impurity\_decrease=0.0, min\_impurity\_split=None, init=None, random\_state=1, max\_features=None, alpha=0.9, verbose=0, max\_leaf\_nodes=15, warm\_start=False, presort='auto')

**6.4 XGBoost Model**

xgb = xgboost.XGBRegressor(n\_estimators=100, learning\_rate=0.08, gamma=0, subsample=0.75,

colsample\_bytree=1, max\_depth=7)

**7. Model Performance Comparison**



It is cleared that random forest is predicting target values with least root mean squared error and other metrics. Now we will try Grid Search to fine tune the parameters of random forest model.

param\_grid = {

'n\_estimators': [800, 1600,2400],

'max\_features': ['auto', 'sqrt', 'log2']

}

CV\_rfc = GridSearchCV(estimator=forest\_reg, param\_grid=param\_grid, cv= 5)

**Best parameters - {'max\_features': 'sqrt', 'n\_estimators': 2400}**

**Root mean squared error for new random forest model 0.008639010819575816**

Previous RMSE for random forest was **0.008743**

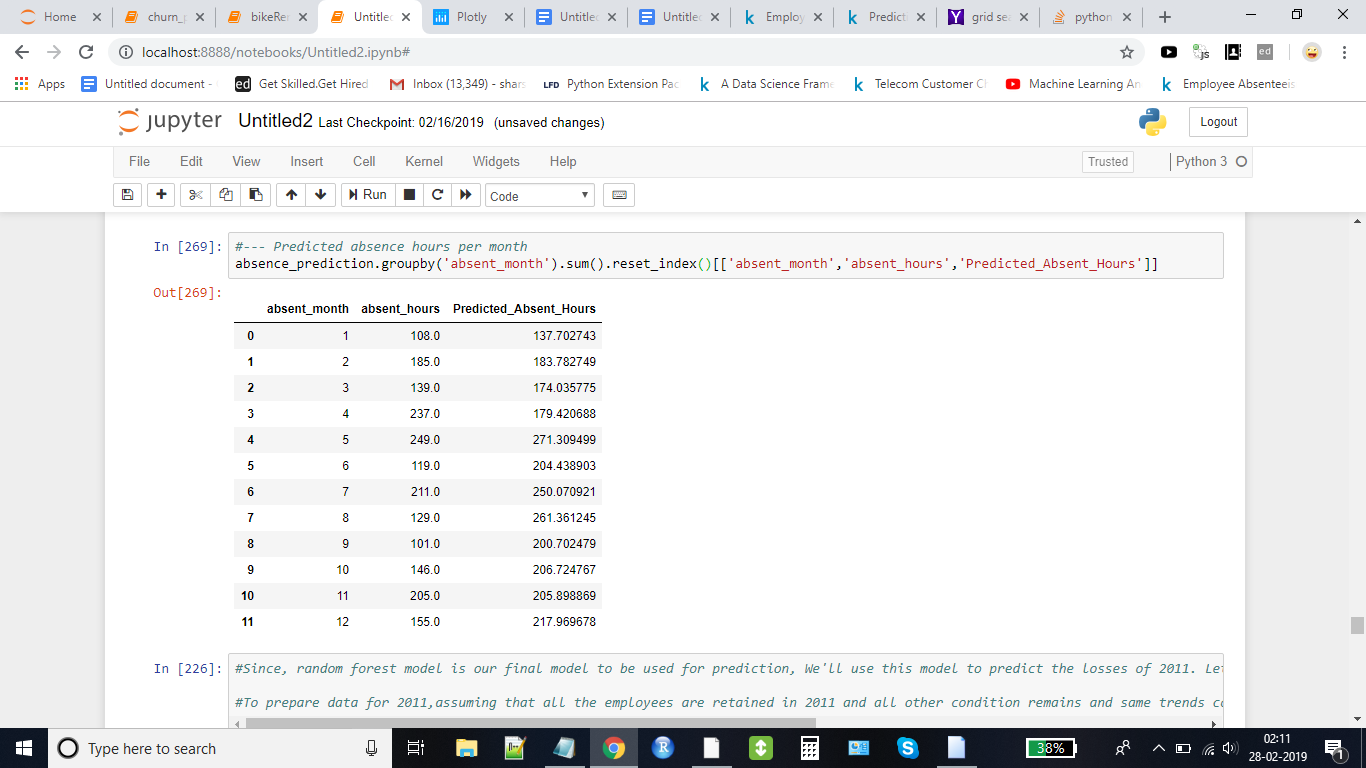
**So it has clearly performed better after tuning the parameters**

**8. Model Predictions**

**Predictions of our model with respect to data provided.**

***Predicted absence hours of 2010 - 2493.418314528776***

***Actual absence hours of 2010 - 1984.0***

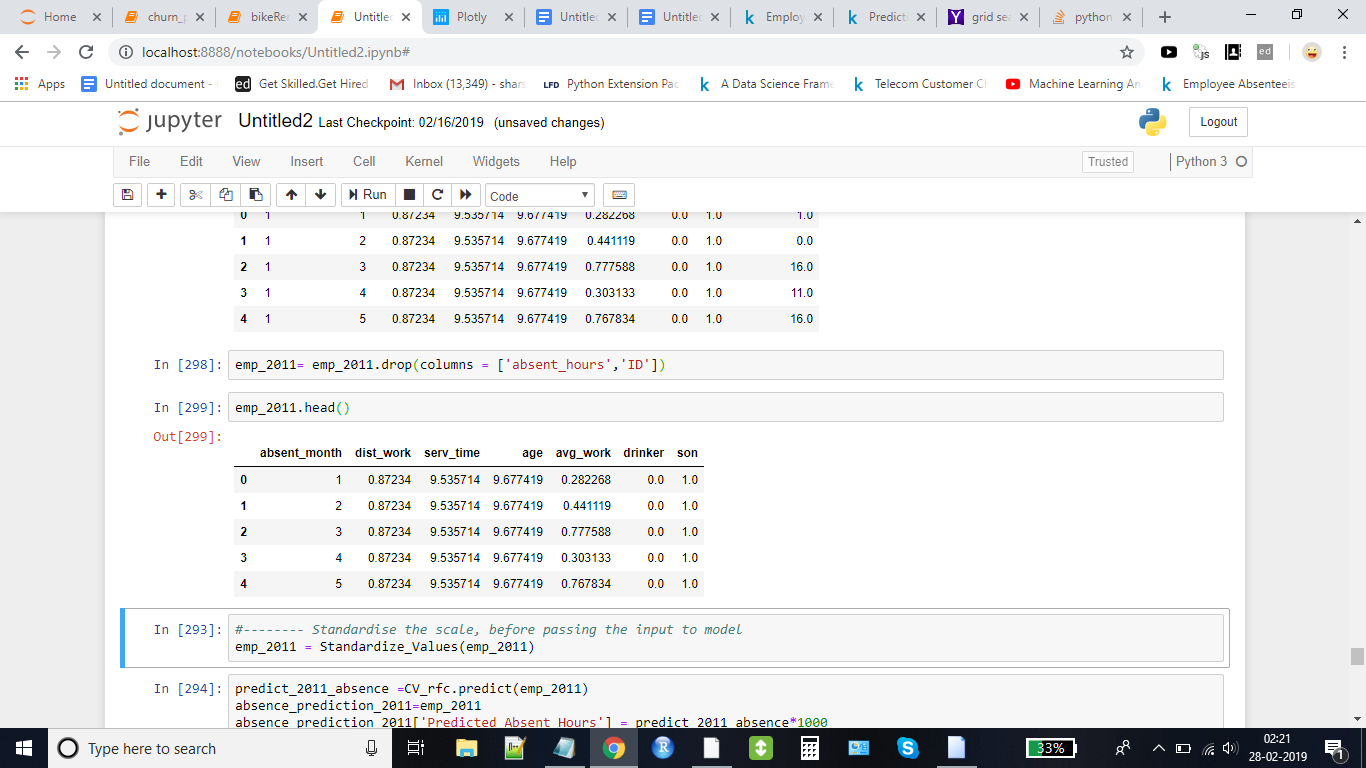
***“ Predicted and actual absence hours per month “***

Since, random forest model is our final model to be used for prediction, We'll use this model to predict the losses of 2011. We will now prepare data for 2011

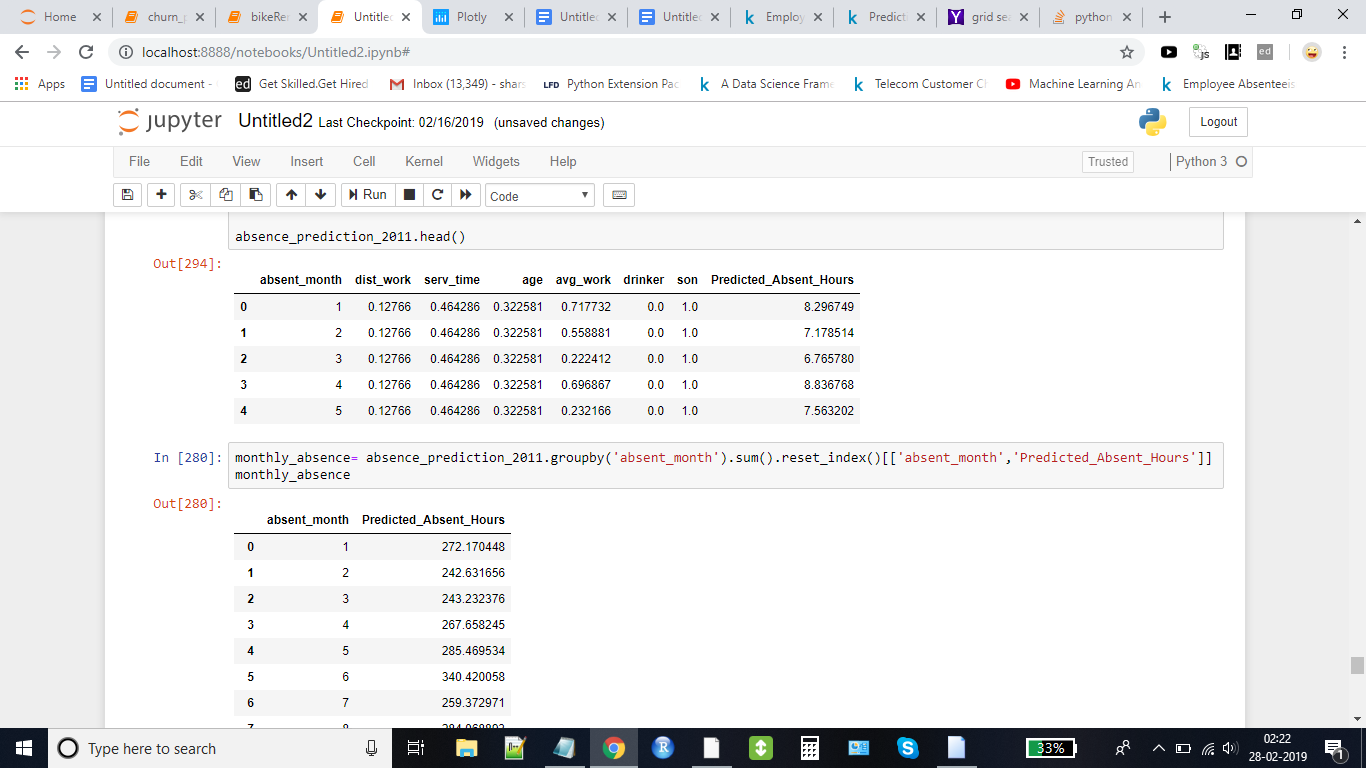
***To prepare data for 2011,assuming that all the employees are retained in 2011 and all other condition remains and same trends continues, we need to add +1 to 'Service\_Time' and 'Age'(keeping all other features same)***

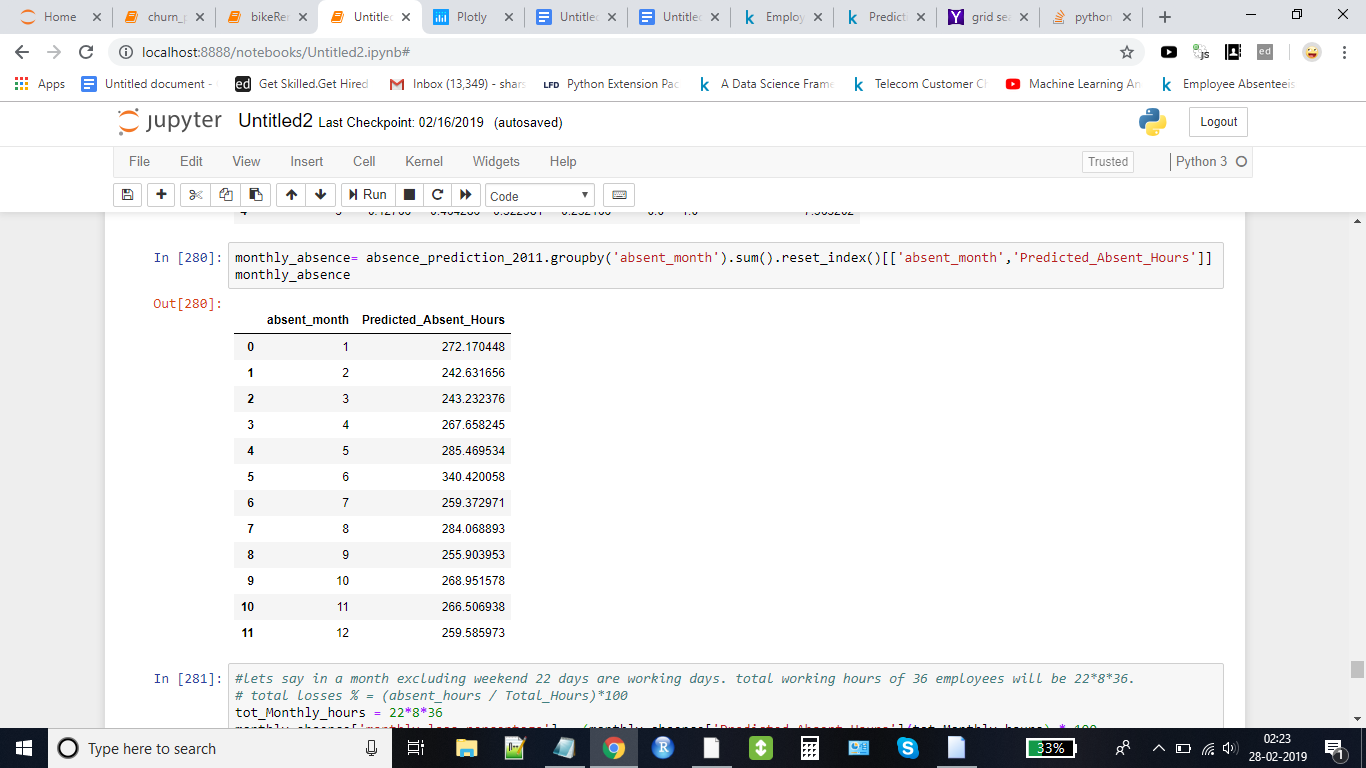
**Predictions for 2011**

**New Prepared Data for 2011**



**Predicted absent\_hours for 2011 data**

**Predicted absent\_hours per month in 2011**



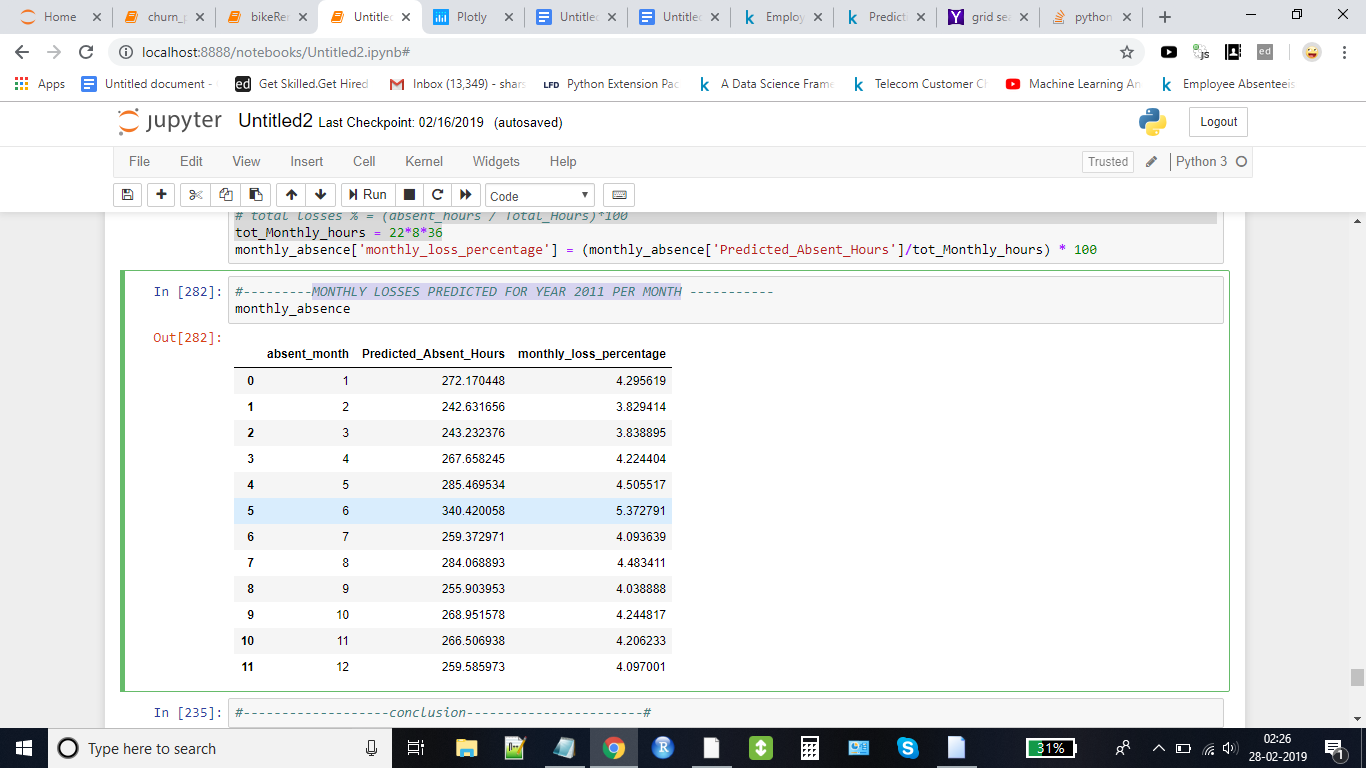
**MONTHLY LOSSES PREDICTED FOR YEAR 2011 PER MONTH**

Let's say in a month excluding weekend 22 days are working days.

Total working hours of 36 employees will be 22\*8\*36.

total losses % = (absent\_hours / Total\_Hours)\*100

tot\_Monthly\_hours = 22\*8\*36



**THE END**