**R&Py Training Project 1**

**Instructions:**

* You will complete this project using Python.
* Create you own code and compare it with output

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

* **Problem Statement: Overall Objective**

Predict the number of check-ins based on the features available in Foursquare’s dataset.

* **Requirement:**

Datasets for these projects are available at below location.

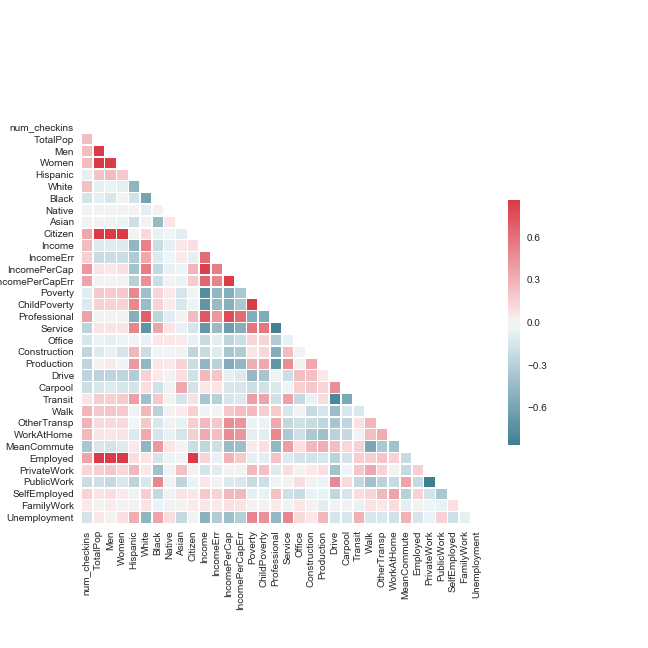
* Dataset1: <https://www.kaggle.com/chetanism/foursquare-nyc-and-tokyo-checkin-dataset>
* Dataset2: <https://www.kaggle.com/muonneutrino/new-york-city-census-data>

**How to achieve the objective: Step by Step Instructions: Procedure**

1. Import NYC check-in dataset(dataset\_TSMC2014\_NYC.csv) into python
2. Convert Coordinates in the dataset to census tracts and output the file to census\_tracts\_per\_checkin.csv
3. Add a census tract column to the census\_tracts\_per\_checkin.csv file. Make sure this column is the same as the dataset\_TSMC2014\_NYC.csv file
4. Import NYC census dataset into python(nyc\_census\_tracts.csv)
5. Create a dataset from 4 that excludes all null values of the income variable from nyc\_census\_tracts.csv and
6. Create a separate dataset consisting of the census\_tract column alone
7. Calculate the total number of checkins per census tract and include it as a separate column in 6
8. Merge datasets created in 5 and 7
9. Remove census\_tract, county and borough columns from the merged dataset in 8
10. Eliminate outliers from the num\_checkins column
11. Impute or remove missing values as you deem fit
12. Construct a correlogram of all the variables, output the correlation values to an external excel file
13. Split the dataset into dependent and independent variables, with the dependent variable being total number of check-ins
14. Do min-max scaling on the independent variables
15. Fit an extra-trees model of the data and display the relative importance of the features
16. Eliminate unimportant features
17. Fit a Regression model and further eliminate any unnecessary features
18. Split the dataset into training and testing
19. Do cross-validation
20. Fit SVR, linear SVR, gradient boosting, ada-boost, SGD-regressor, lasso and elastic net models on the dataset and measure the cross-validation accuracy scores
21. Fit the model with the best cross-validation accuracy to the test data.
22. Output the prediction results to an excel sheet.

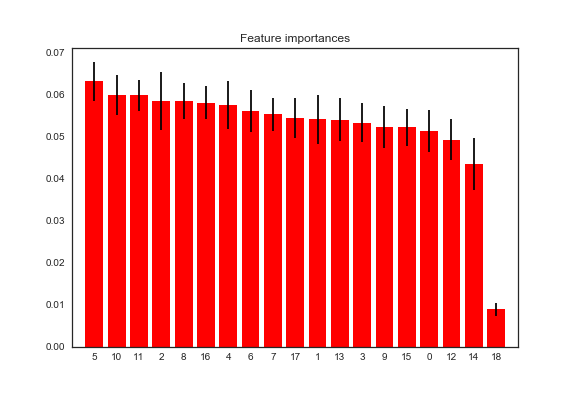
* **Final Goal (Desired Output):**

Desired Output (Numbers may vary depending on the training and testing samples):

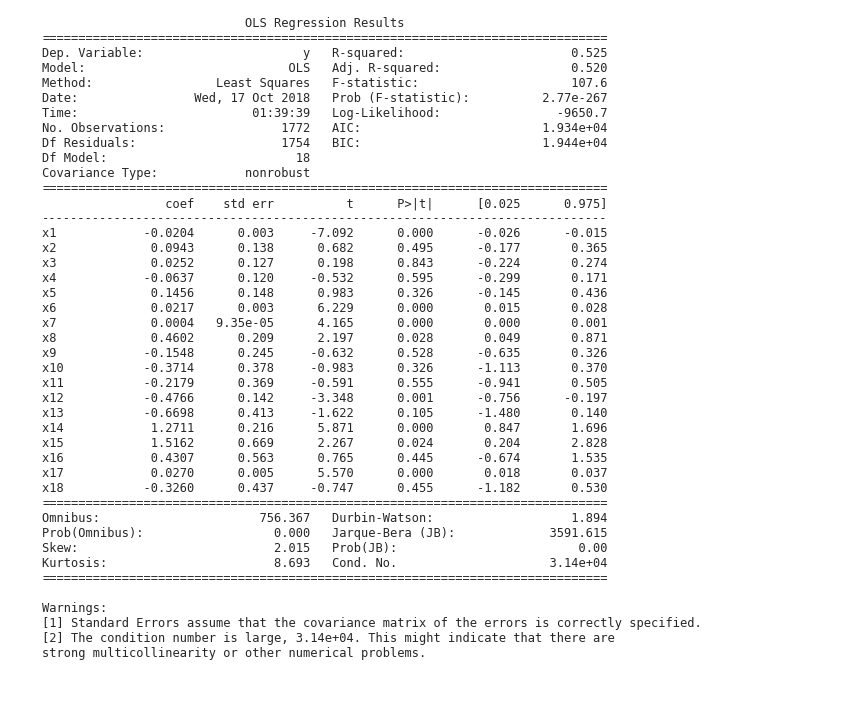


*Figure 1*. Correlation diagram representing the correlations among the variables in the dataset.

A correlation value close to one signifies a strong correlation, while a correlation value close to zero indicates negative correlation. The varying shades of red indicate a positive correlation, while the different shades of blue represent negative correlation.



*Figure 2*. The feature importance plot has the variables as abscise and the feature importance value as the ordinate. The variable 18 has the lowest feature importance score and it corresponds to the variable ‘family work’

Table 3: OLS results

*Note.* The variables that have a p-value less than the significance level are x1, x6, x7, x8, x12, x14, x15 and x17, these variables correspond to the independent variable’s population, citizen, income, poverty, drive, walk, other transportation and employment.

Table 4: *Model Performance after feature selection using both Tree Classifier and Regression*

|  |  |  |  |
| --- | --- | --- | --- |
| Model | *Train* | *Cross-Validation* | *Test* |
| SVR | 70.68 | 69.91 | 68.75 |
| Linear SVR | 67.08 | 66.35 | 68.54 |
| Gradient Boosting | 43.39 | 58.85 | 69.47 |
| Ada Boosting | 70.70 | 75.35 | 91.14 |
| SGD | 61.51 | 61.06 | 65.82 |
| Linear Regression | 56.04 | 55.86 | 66.04 |
| Lasso | 56.44 | 56.26 | 65.72 |
| enet | 59.47 | 59.09 | 65.62 |

*Note.* The model abbreviations denote support vector regression, linear support vector regression, gradient boosting, Ada boosting, stochastic gradient descent, linear regression, lasso regression, and elastic net in that order. Train refers to training accuracy, Cross-Validation refers to cross validation accuracies with five folds and Test denotes test accuracy.

* **Source Code:**

It is highly recommended to try and come up with your own code and solution of above project for best learning. However, if any questions, please use below code for your reference -

<https://github.com/datadream89/digital_estate/blob/master/Applied_Project_ANLY.ipynb>