Final Report Project 5: Restaurant Revenue Prediction

PHYS 247: Applied Data Science - Spring 2020
Silvia Cabellos – scabe002@ucr.edu

Description of the project:

This project was one of the competitions posted on www.kaggle.com a few years ago and the goal is to build a model to "**predict annual restaurant sales based on objective measurements".**

"Finding a mathematical model to increase the effectiveness of investments in new restaurant sites would allow TFI to invest more in other important business areas, like sustainability, innovation, and training for new employees. Using demographic, real estate, and commercial data, this competition challenges you to predict the annual restaurant sales of 100,000 regional locations."

For the class, the purpose of this report is to provide details about the data sets, the processing and cleaning as well as the chosen method and evaluation.

Separately to this document, I will be providing:

1. Jupyter Notebook with all the python code.
2. Submission csv file, with an id and prediction of the test data.

Data Set :

The data set can be found in https://www.kaggle.com/c/restaurant-revenue-prediction/data , there is a "train.csv" and a "test.csv"
The train.csv contains 43 columns with the following data fields in 137 rows:

1. Id : Restaurant id.
2. Open Date : opening date for a restaurant
3. City : City that the restaurant is in. Note that there are unicode in the names.
4. City Group: Type of the city. Big cities, or Other.
5. Type: Type of the restaurant. FC: Food Court, IL: Inline, DT: Drive Thru, MB: Mobile
6. P1, P2 - P37: There are three categories of these obfuscated data.
    a. Demographic data are gathered from third party providers with GIS systems. These include population in any given area, age and gender distribution, development scales.
    b. Real estate data mainly relate to the m2 of the location, front facade of the location, car park availability.
    c. Commercial data mainly include the existence of points of interest including schools, banks, other QSR operators.
7. Revenue: The revenue column indicates a (transformed) revenue of the restaurant each year and is the target of predictive analysis. Please note that the values are transformed so they do not mean real dollar values.

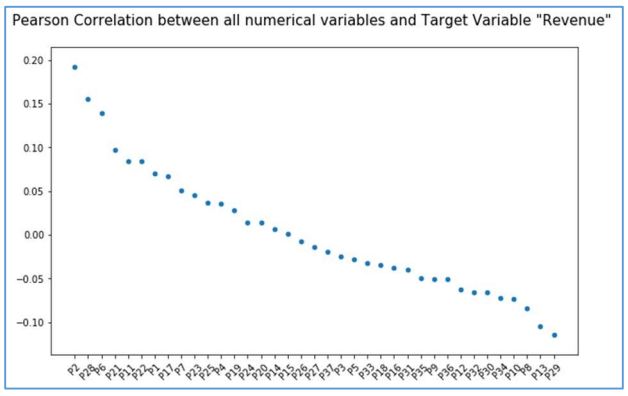
The test.csv contains 100,000 rows with the same fields minus the target value (revenue)

Data and Cleaning Processing

After importing both data for training and data for testing, I quickly realize that the sample for training the model is very small 137 rows × 43 columns and the model is going to be trained/validated using that data alone. That poses some limitations as it is more difficult to uncover patterns as well as it deteriorates the capability of making robust predictions of the unseen data.

The data set contains 43 columns out of which 4 are categorical and the rest are numerical. The target variable (“revenue”) is a continuous variable, meaning that we have to build a supervised regression model.
To get started with the data/cleaning process, and after checking for duplicates/missing numbers, we are going to look at the correlation for all the numerical values.

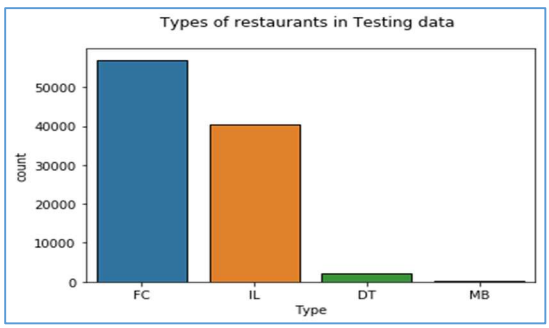
As a general guideline, we should keep those variables which show a decent or high correlation with the target variable, but we can see in the plot below, the direct (or inverse) correlation with numerical variables is low.
The top 3 variables are (P2, P28 and P6) and is still below 20%.



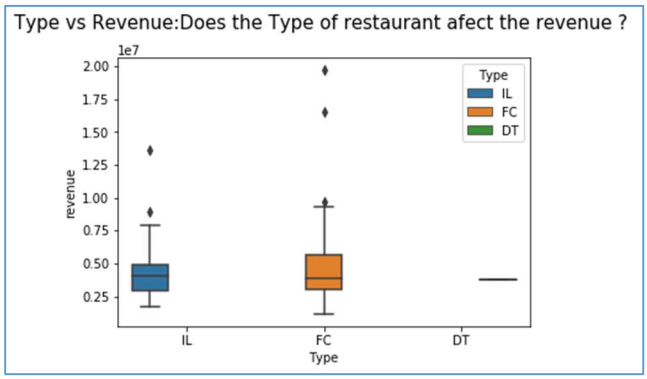
The next step in the data exploration is to look the categorical variables and their relationship with revenue. These are not obfuscated and will be easier to interpret and visualize. There are 4 categorical Variables. ‘Type’ /’Open Date’ / ‘City’ /’City Group’

Type :

The types of restaurant differ between data and testing. Types “MB” and “DT” represent a very small portion of the datasets and can be removed.

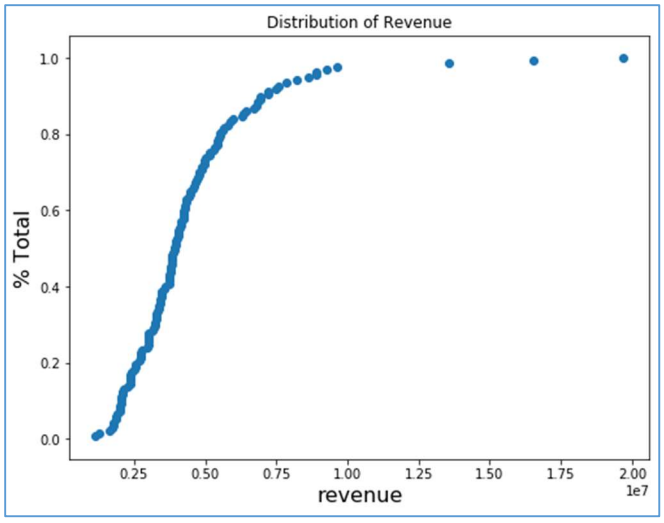


We can also look at the relationship between type and revenue visualizing a box-plot.



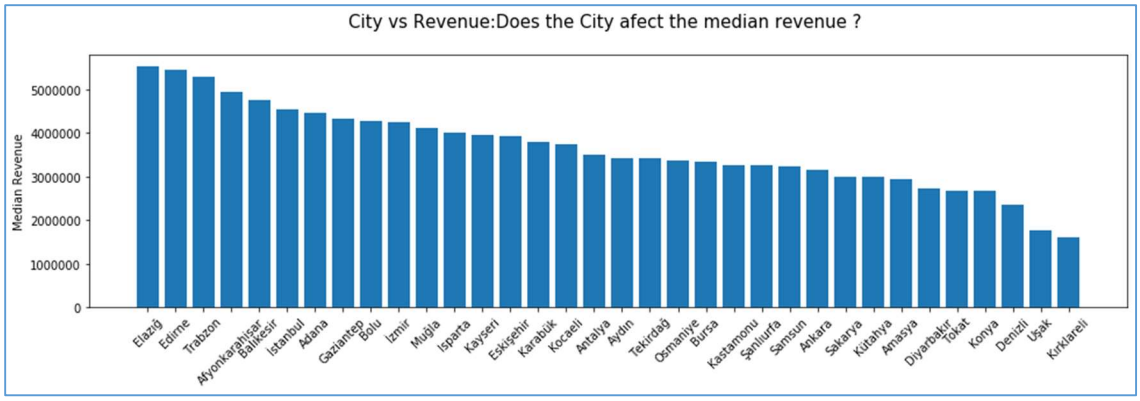
We noticed that there are a few outliers over the fence of the boxes. We can draw the revenue data points and see the spread corresponding to the distribution.
Because the error is the squared distance between the data point and the predicted value, large distances have disproportionately large errors which cause the regression analysis to converge on a solution with a poor correlation **coefficient**.

As such, outliers should ideally be removed from the data-set. Identifying outliers can be a somewhat a tricky task. - Dropping the last few records with highest revenue will give us more powerful model. We can remove those points that are over 9M in revenue.



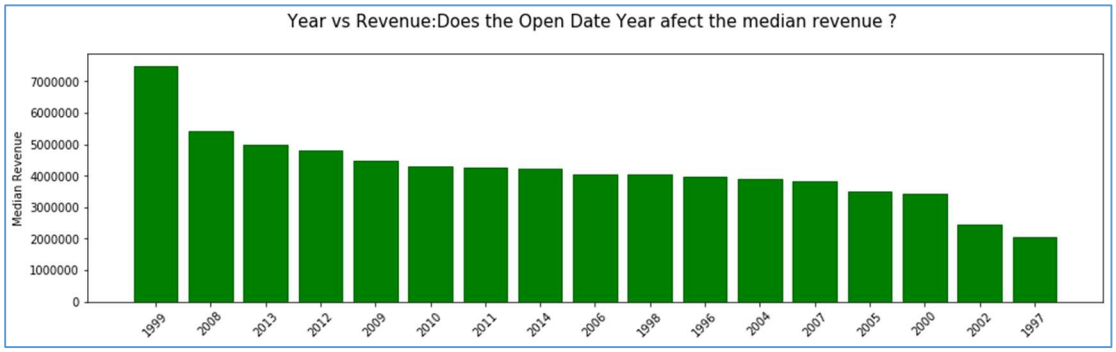
City :

To visualize how the city and the revenue are related, we can run a plot with the median revenue by city in sorted order.



Open Date.

We can also plot the relationship between the year when the restaurant opened its doors and its sales.



The first attempt of cleaning and preprocessing this “Open Date” variable was to extract the “Open year”, which is easier to categorize than the complete date, but then I noticed that both data/test sets have different number of years so I decided to create a variable with the “number of months since they opened”. I could have done “days since opened”, but then variability will be a lot higher for that feature.

```
]:  1  #Converting Open_Date into Open months count
    2
    3
    4  data['Months_Open'] = round((pd.to_datetime('2015-01-01') - pd.to_datetime(data['Open Date'])).dt.days/30,0)
    5  test['Months_Open'] = round((pd.to_datetime('2015-01-01') - pd.to_datetime(test['Open Date'])).dt.days/30,0)
```

Dropping and Encoding Categorical Variables:

Finally, I decided to drop some columns and encode the rest of the categorical variables using the method “get_dummies”. As a summary of the data cleaning and preprocessing I have done the following:

1. Import the data set
2. Check for duplicates and missing values
3. Analyze correlation of numerical variable with target variable
4. Analyze Outliers for Revenue and transform “Open Date” into “Months since Open”

5. Encode City Group, Type
6. And drop the following variables. I kept the "Id" in the test set for submission purposes.

```
X=data_clean.drop(columns=['Id','Open Date', 'City','Open Date','Open_year'])

# for the test - i keep the ID around to build the final prediction/submission
X_test_with_ID=test_clean.drop(columns=['Open Date', 'City','Open Date','Open_year'])
```

Method and Evaluation

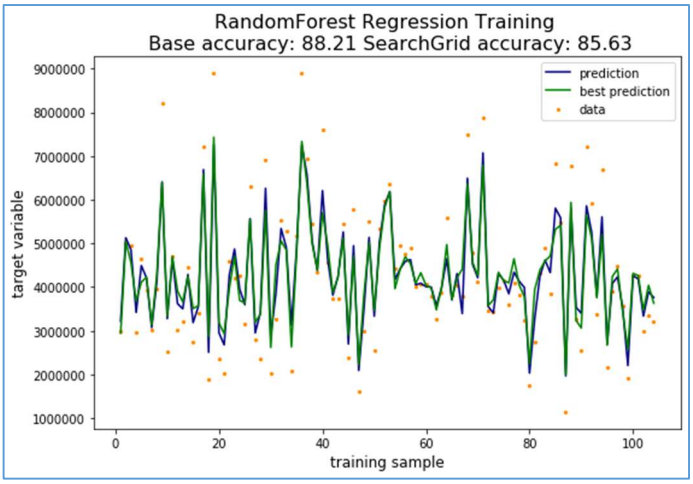
Now that the data is clean and analyzed, we are going to:

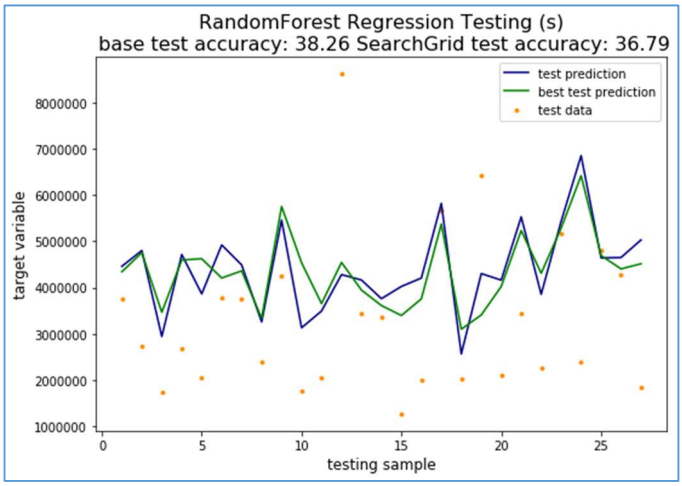
1. Split the data set into training/testing to help evaluate our model.
2. Scale the data using MinMaxScaler.
3. Train the model. Before training the model, I am going to run a RandomizedSearchCV for the best parameters for our training model and apply those and compare results and scores to a previously chosen base model.

After testing some models (Lasso, Rigde, KNeighborsRegressor), finally I built two different models: one using RandomForest Regression – and a second one for GradientBoostingRegressor (which might be preferred for small sample sizes and is robust to over-fitting).

Below are some of the evaluation metrics for both methods. I calculated the error between labels and predicted values for both training and testing samples.

1. **Random Forest Regression**

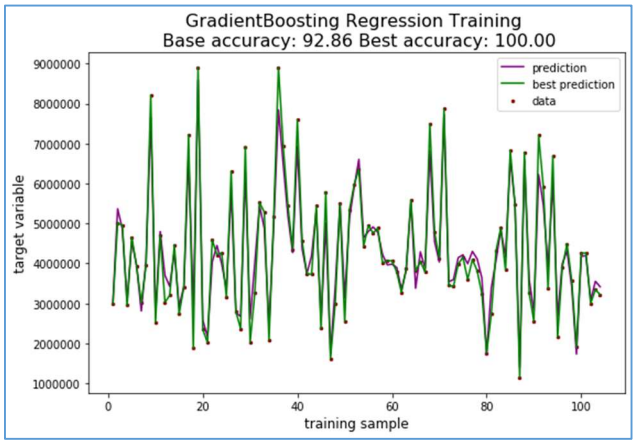


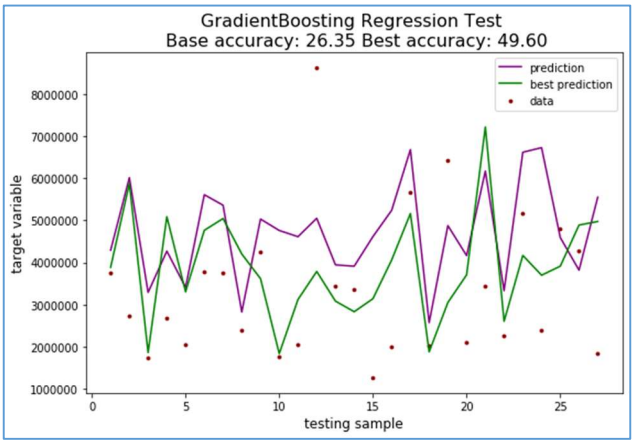



```
Model Performance
Average Error: 441375.2674 .
Accuracy = 88.21%.

Model Performance
Average Error: 538594.8563 .
Accuracy = 85.63%.

For training data Improvement of -2.93%.
```

```
Model Performance
Average Error: 1554763.0867 .
Accuracy = 38.26%.

Model Performance
Average Error: 1575796.0718 .
Accuracy = 36.79%.

For testing data Improvement of -3.84%.
```

2. Gradient Boosting Regression






```
Model Performance
Average Error: 270355.1650 .
Accuracy = 92.86%.

Model Performance
Average Error: 0.0002 .
Accuracy = 100.00%.

For training data Improvement of 7.69%.
```

```
Model Performance
Average Error: 1810073.7614 .
Accuracy = 26.35%.

Model Performance
Average Error: 1452382.1186 .
Accuracy = 49.60%.

For testing data Improvement of 88.22%.
```

As a final step, I applied both models to the testing data set and exported the predictions in a submission file.

This concludes the Final project¶

In summary, I imported the data sets, cleaned, and analyzed the variables and preprocessed the data sets, transforming the categorical values and removing outliers.

After that, I split, normalized the data, and trained different regression models and fine-tuned the hyperparameters to find the best estimators. Finally, I chose the models with best accuracies and smallest errors and applied those to the big testing sample, exporting the results to a submission csv file.

Thank you!

Silvia