Restaurant Revenue Prediction

Luis A. Hernandez
March 19, 2021

UC RIVERSIDE

Introduction

Problem Set Summary: Restaurant Revenue Prediction

- Based off of a publicly available dataset from Kaggle, the aim of this project is to predict annual restaurant revenue sales based on objective measurements.

- Provided datasets include:
  - train.csv
    - Dimensions: 137(rows)x43(cols)
    - ‘revenue’ feature included
  - test.csv
    - Dimensions: 100000(rows)x42(cols)
    - ‘revenue’ feature not included
    - ‘MB’ not included within ‘Type’ feature of test set.
    - Cities incomplete in training set.

UC RIVERSIDE

Introduction

Python Usage

- Primarily using python programming language, the datasets were cleaned through pandas dataframe methods.
- Once clean, the data was visualized through matplolib/seaborn to see if there were any interesting insights.
- Then, the data was analyzed to predict the target variable (revenue) using sci-kit learn.

UC RIVERSIDE

Project Layout

- I already mentioned the usage of python, but what steps will be explicitly taken:
  a. Data Exploration
     - Dataset Background.
     - Map visualization: Report location and prevalence.
     - Revenue generating cities.
     - Revenue generating restaurant types.
     - Prominent restaurant launch dates.

  b. Data Preparation
     - Normalization assumption.
     - Dummy Variables
     - Supplementary Challenge Consideration.
     - Scaling.

  c. Analysis
     - Principal component Analysis.
     - Linear Regression.

Data Exploration

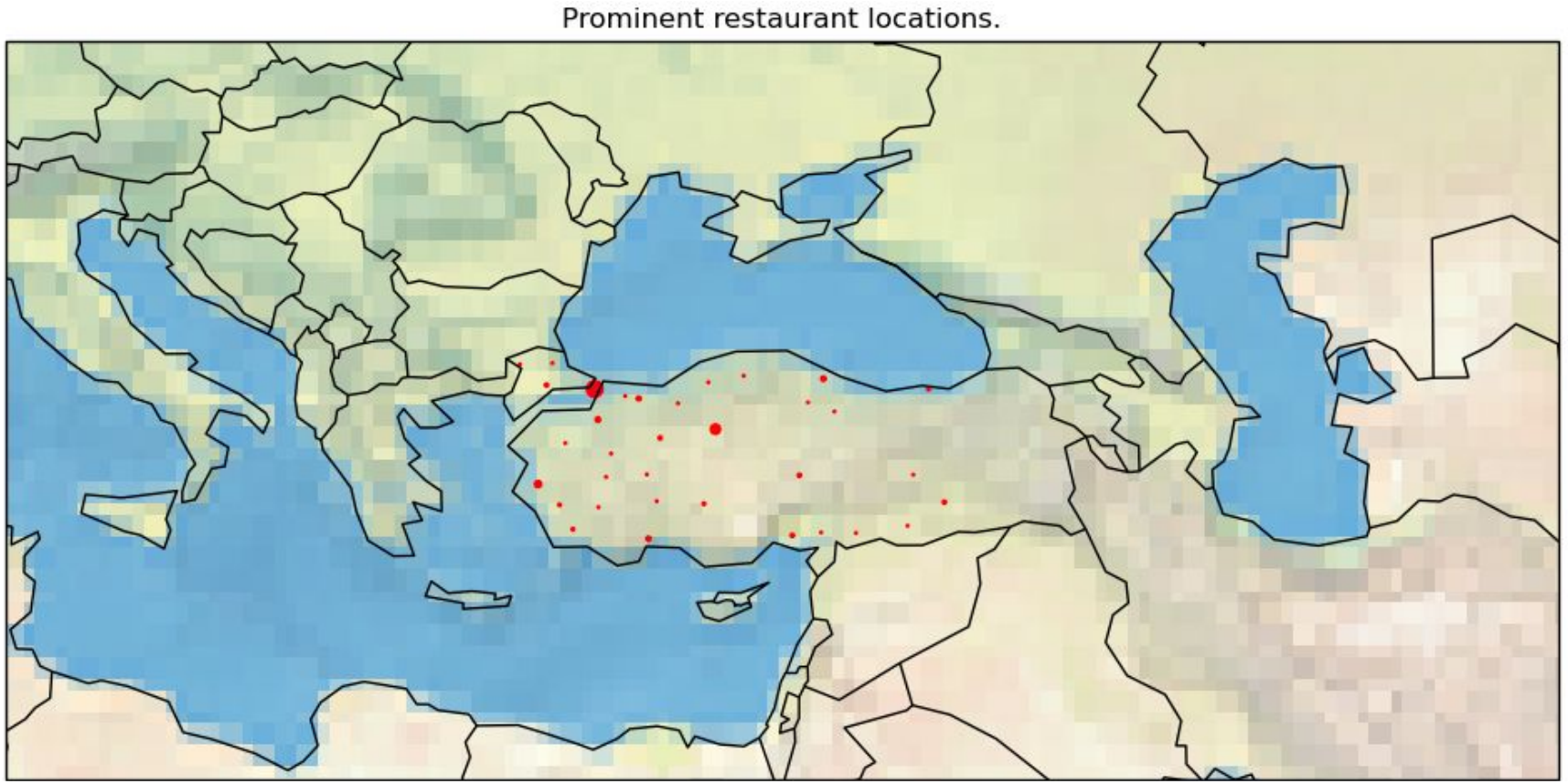
Dataset Background

- After importing the data, all of the datapoints were determined to be located in Turkey when exploring throughout the dataset ‘City’ column.

- The next thing to find out though, was in what city the restaurants were located and to see what kind of balance there was in this dataset.

Data Preparation

Location

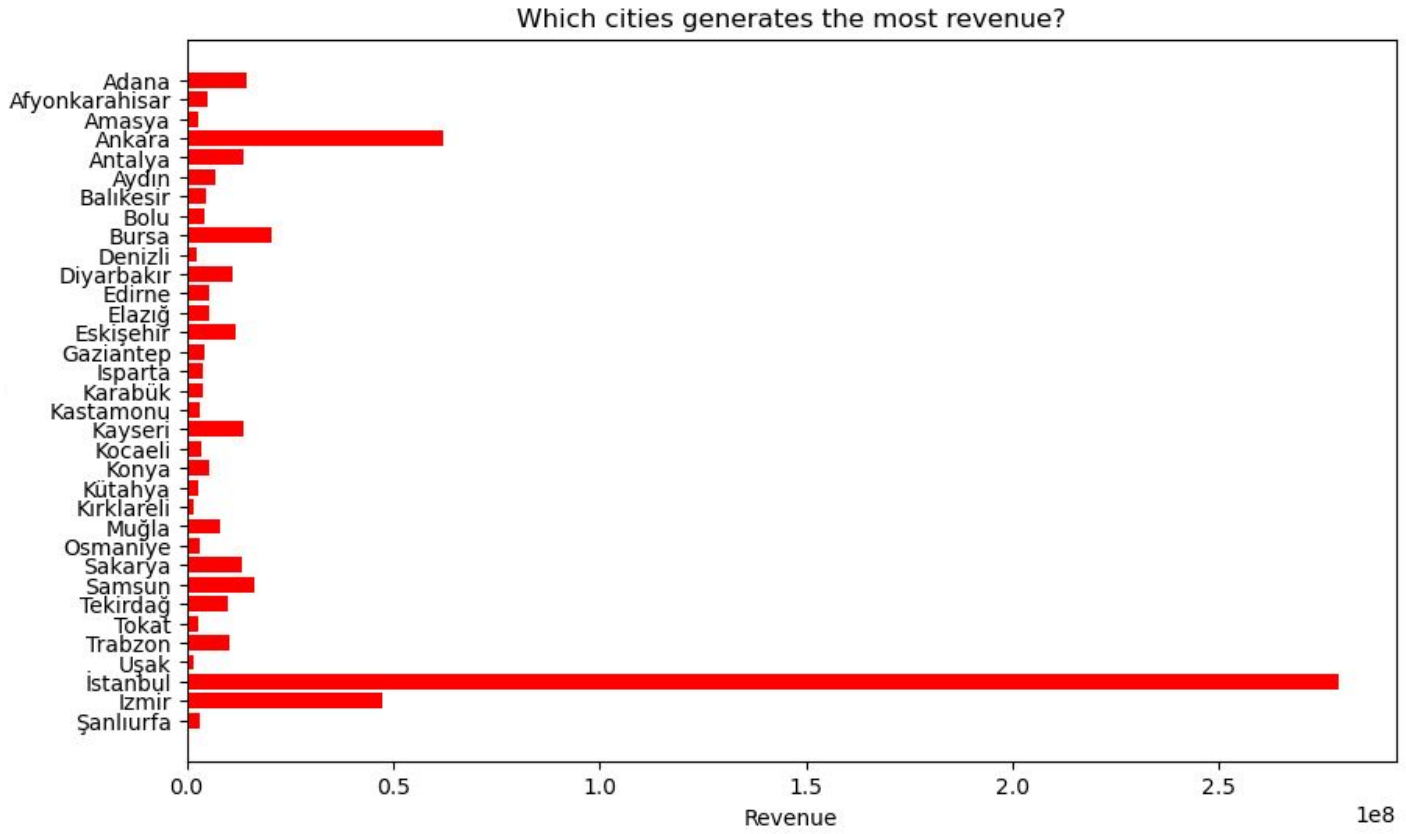


Data Exploration

Location Extended

- Seeing clearly the hierarchy of the top contributing cities in the country through a map visualization, a barchart was the next thing that came to mind to explicitly see the names of those cities.

- With Istanbul, a widely considered metropolitan area, being the top city in the training set, what subsequently made sense to me was to see what kind of behavior dominated the restaurant types.



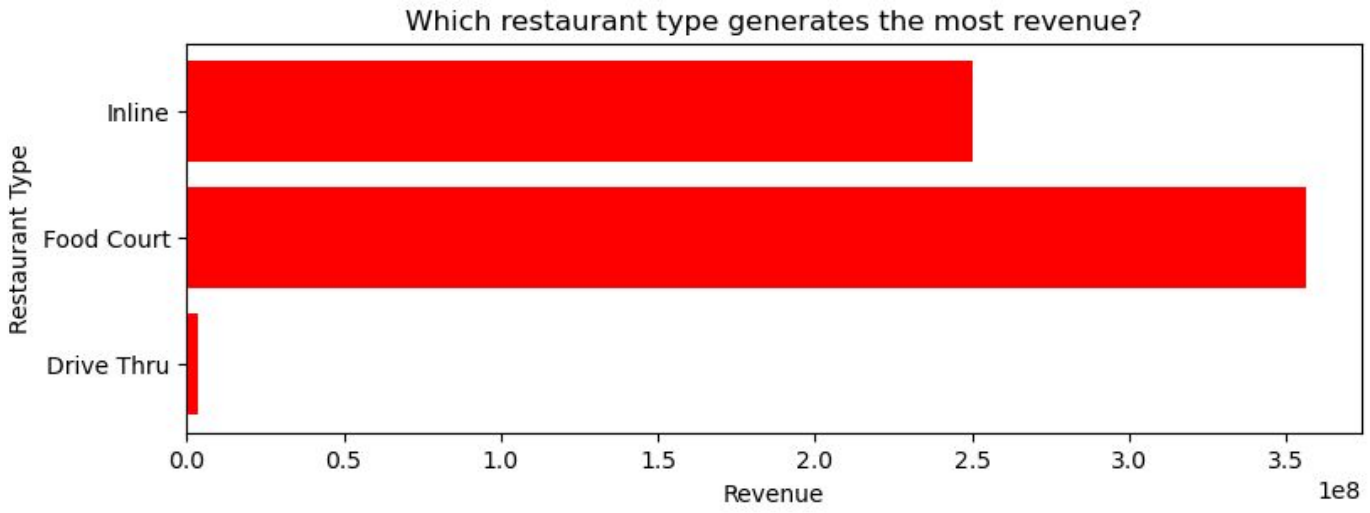
UC RIVERSIDE

Data Exploration

Restaurant Types

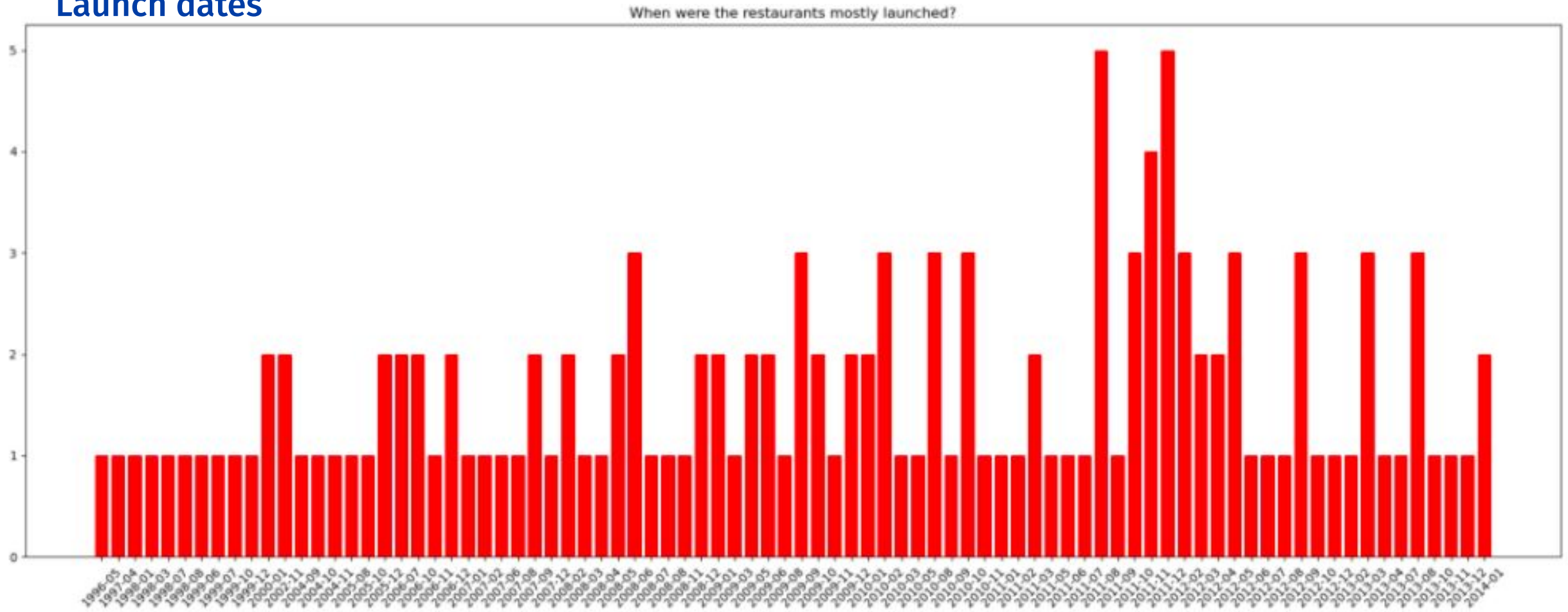
- Keeping in mind that previously it was found out that Istanbul dominated the restaurant scene, it was interesting to find out that Food Court was the corresponding restaurant type majority in the training set.

- Furthermore, if we remember back to the intro, it was mentioned that the mobile ‘MB’ type was not present in the training set and we can see that explicitly here. This will be corrected through KNN imputation to compensate for this missing value.



Data Exploration

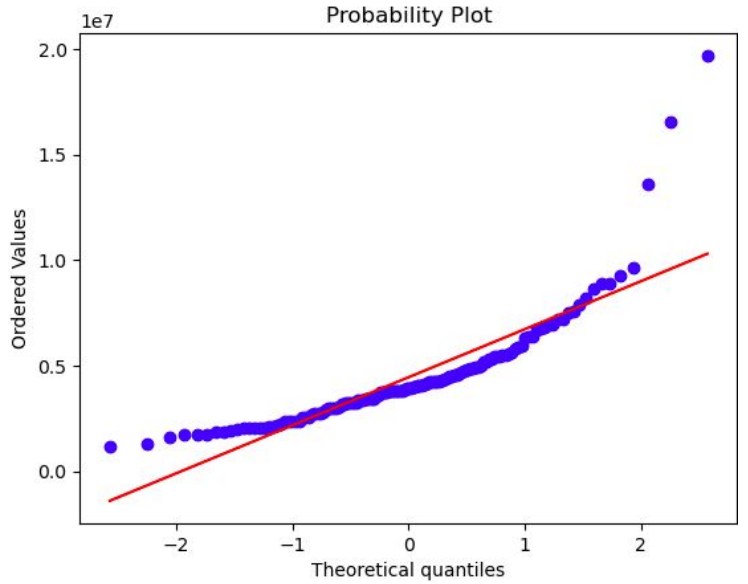
Launch dates

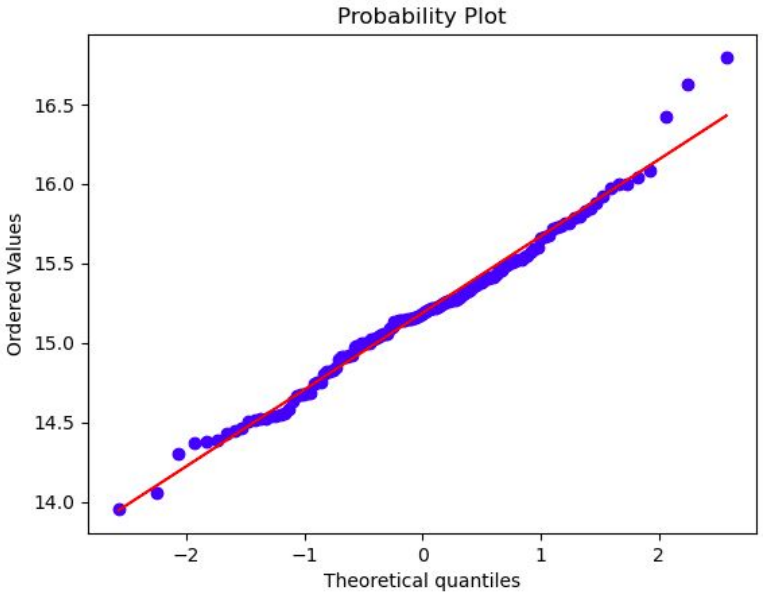


Data Preparation

Pre-Model Check: Normalization

- Being that a linear regression will be performed on this dataset to predict the revenue, a quick normality assumption check was performed.
- On the top, we can see that the data is rightly skewed with a couple of outliers.
- On the bottom is the same data with a logarithm applied to the data.
- Given that this dataset is already limited in size, I decided to keep the outliers.





UC RIVERSIDE

Data Preparation

Dummy Variables

- Considering the fact that there were a couple of categorical variables and that the models perform based on numerical variables, they had to be converted.
- The first one was the ‘City Group’ feature that was simple to convert, by considering the ‘Big City’ type as the base case (i.e. 0) and the ‘other’ type as 1.
- Next was the previously mentioned restaurant ‘Type’ feature, which itself was already a challenge as the mobile case of this feature was missing in the test set. In order to compensate for this, a KNN imputation was performed by considering the ‘MB’ case as missing/null and using the remaining features to output the most probable classification to any of the other types. Considering ‘Food Court’ as 1, ‘Inline’ as 2, and ‘Drive Thru’ as 3, the resulting ‘Type’ feature was later used for the revenue prediction model.

Data Preparation

Pending operation incomplete-Kmeans transform of ‘City’ feature.

- Included in the previously presented dummy variables is the ‘City’ categorical data, which as we can see from the image on the right is incomplete in the training dataset.

- Contrast to the previous methods performed, this one was challenging enough to remain pending for this analysis as p-value driven Kmeans was necessary for this part.

K-means clustering for city compensation ¶

```
len(res_train['City'].drop_duplicates())
```

```
34
```

```
len(res_test['City'].drop_duplicates())
```

```
57
```

UC RIVERSIDE

Data Preparation

Pre-Model Scaling

- Concentrating on the response variable-revenue-first, it was stored under three variables to evaluate how each output would influence the linear regression model, which more realistically came out to be the original figure in whole currency values.

- Next, using the StandardScaler available through sci-kit learn, the remaining variables were normalized in both the training set and test set in order to have a more accurate model performance.

- Additionally, one of those three stored response variables was normalized as well.

Analysis

Principal Component Analysis

- Once normalized, the data was ready for model implementation, but seeing that the size, specifically the amount of columns, was considerable, an additional method had to be applied in order to see if it made a difference in model performance.
- This is where Principal Component Analysis (PCA) comes in, which in this case 10 were chosen to see how they contributed.

	PCA 0	PCA 1	PCA 2	PCA 3	PCA 4	PCA 5	PCA 6	PCA 7	PCA 8	PCA 9
0	1.012842	-2.438753	0.676485	-0.478576	0.245135	-0.125838	-0.493670	2.554560	-0.299681	1.377948
1	-2.569325	0.616360	0.569430	-1.591578	-0.200607	-0.204548	1.100668	0.020088	0.194114	-0.158363
2	-2.982403	0.001187	-1.559098	0.356310	0.955777	-0.998175	0.736698	0.008614	-1.207692	-1.328838
3	10.754897	0.797182	-3.881189	3.059873	0.144958	-1.669889	1.866375	0.386360	0.098977	2.099220
4	0.043206	-2.614608	-1.376587	0.011273	-0.089899	-0.456133	0.391977	0.950938	-1.251368	-0.975633

UC RIVERSIDE

Analysis

Linear Regression

- Initially processed with PCA features, the efficiency of the model is displayed through the r^2 value in the top figure to the right.
- Seeing that it was a low value, an alternative method was considered with a combination of the PCA training set and the non-logarithm of the response variable, which is the middle figure.
- Finally, the non-logarithm and non-PCA features were considered, yielding the bottom result of the figure to the right.

```
#r^2 value for the first case.
lr_model.score(pca_train,rtrain_ylog)
```

```
0.16432385226944046
```

```
#r^2 value for new condition.
lr_mod1.score(pca_train, rtrain_y)
```

```
0.17137996630362795
```

```
#r^2 value for new condition.
lr_mod2.score(restaurant_train, rtrain_y)
```

```
0.35698460553456723
```

UC RIVERSIDE

Analysis

Linear Regression

- Seeing that the best case to consider was the latter one, the equation that represents that linear regression model is given below and the predicted values are displayed in the jupyter notebook:

$$y = 376208464.86x + -248563.0922338681x_1 + -973123.53x_2 + 321649.43x_3 + 161051.31x_4 + -224605.67x_5 + -92758.24x_6 + 47900.84x_7 + 298724.93x_8 + 32700.57x_9 + -1251762.89x_{10} + 1493858.24x_{11} + 16636.69x_{12} + -267240.66x_{13} + -296008.94x_{14} + -711795.70x_{15} + -130967.12x_{16} + -221059.06x_{17} + -388506.68x_{18} + 219037.84x_{19} + 361843.95x_{20} + -100343.4x_{21} + -331946.44x_{22} + 164907.69x_{23} + -297270.08x_{24} + 155878.28x_{25} + 529504.76x_{26} + 359115.32x_{27} + -1075796.78x_{28} + 123448.5x_{29} + 448317.28x_{30} - 104774.2x_{31} + 69890.90x_{32} + 181840.09x_{33} - 315823.94x_{34} - 129597.68x_{35} - 171514.06x_{36} + 21228.84x_{37} + 666847.09x_{38} - 10774.76x_{40} + 4726.75x_{41} + 23927.01x_{42} - 183046.81x_{43}$$

UC RIVERSIDE


Thanks!
UC RIVERSIDE