**1. Resturant Revenue Prediction(Kaggle Dataset)**

Tab Food Investment(TFI) is a quick service restaurants across the globe with some of the world's most well-known brands such as Burger King, Sbarro, Popeyes, Usta Donerci, and Arby’s. With over 1200 outlets, they currently employ over 20,000 people in Europe and Asia and make significant daily investments in developing new restaurant sites.With investments in opening new outlets, the objective of the current exercise is to find 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.

**Data description:**

TFI has provided a dataset with 137 restaurants in the training set, and a test set of 100000 restaurants. The data columns include the open date, location, city type, and three categories of obfuscated data: Demographic data, Real estate data, and Commercial data. The revenue column indicates a (transformed) revenue of the restaurant in a given year and is the target of predictive analysis.

**Data Sets:**

Id : Restaurant id.   
Open Date : opening date for a restaurant  
City : City that the restaurant is in. Note that there are unicode in the names.   
City Group: Type of the city. Big cities, or Other.   
Type: Type of the restaurant. FC: Food Court, IL: Inline, DT: Drive Thru, MB: Mobile  
P1, P2 - P37: There are three categories of these obfuscated data.

Demographic data are gathered from third party providers with GIS systems. These include population in any given area, age and gender distribution, development scales. Real estate data mainly relate to the m2 of the location, front facade of the location, car park availability. Commercial data mainly include the existence of points of interest including schools, banks, other QSR operators.

Revenue: The revenue column indicates a (transformed) revenue of the restaurant in a given year and is the target of predictive analysis.

Please note that the values are transformed so they don't mean real dollar values.

**Experiments conducted:**

**Steps in Data Processing:**

1.Type Column: Converted categories into two (instead of four) types (Food Court and Others)  
2. Calculated Age of current resturants (from OpenDate Column)  
3. Data Standardisation for numerical columns using StandardScaler  
4. Dummified the categorical variables  
5. Data was split into 80:20 for training and validation   
6. PCA for all variables between P1, P2-P37

**Models Applied:**

Linear Regression(with and without PCA),   
Elastic Net,   
Gradient Boosting,  
XGBoost with Grid SearchCV,   
Random Forest

**Error Metrics for evaluation** - RMSE

**Model Results:**

Random Forest with best parameter perform better than any other applied algorithm with 1745455   
RMSE value.

**2. Kaggle Dataset: Whats Cooking**

Some of our strongest geographic and cultural associations are tied to a region's local foods. This playground competitions asks you to predict the category of a dish's cuisine given a list of its ingredients.

Submissions are evaluated on the categorization accuracy (the percent of dishes that you correctly classify).Submission should predict the cuisine for each recipe in the test set.

**Data description:**

In the dataset, recipe id, the type of cuisine, and the list of ingredients of each recipe (of variable length) is included. The data is stored in JSON format. In the test file test.json, the format of a recipe is the same as train.json, only the cuisine type is removed, asit is the target variable to be predicted.

**Data Fields:**

train.json - the training set containing recipes id, type of cuisine, and list of ingredients  
test.json - the test set containing recipes id, and list of ingredients  
sample\_submission.csv - a sample submission file in the correct format

**Steps in Data Processing Conducted:**

1. Removing space within words from the ingredients column   
2. Removing the list of lists within the 'ingredients' column  
3. Creating an addditional column to remove the list of lists, hypen and then concatinating to orginal dataframe - now the new column is a cleaned column for applying TFIDF vectorization   
4. Data was split into 80:20 for training and validation

**Models Applied:**

Support Vector Machine,   
Random Forest  
Stochastic Gradient Descent

**Error Metrics for evaluation** – Accuracy

Model Results:

LinearSVC performs better results than other algorithms used with 78 percent accuracy on test data.

**3. Analytics Vidhya Hackathon**

Sales of a given product at a retail store can depend both on store attributes as well as product attributes. Build a data science solution to predict the same for 'Big Mart'!

**Columns provided in Train/Test Datafile**

Item\_Identifier Item\_Weight   
Item\_Fat\_Content   
Item\_Visibility Item\_Type   
Item\_MRP   
Outlet\_Identifier   
Outlet\_Establishment\_Year   
Outlet\_Size   
Outlet\_Location\_Type   
Outlet\_Type   
Item\_Outlet\_Sales (Target Column)

Train Rows - 8524   
Test Rows (for prediction) - 5682

**Experiments conducted:**

**Steps in Data Processing:**

1. Rename the categories in 'item\_identifier' column to more intuitive categories

2. Calculating the 'Age' of the outlet from 'Outlet\_Establishment\_Year' column

3. Co-relation of the numerical variables (most independent variables are low or moderately co-related with 'Sales Revenue'

4. Transforming the Y variable for removing skewness using log, squareroot and boxcox transfomation

5. Data Standardisation for numerical columns using StandardScaler

6. Dummified the categorical variables

7. Data was split into 80:20 for training and validation

**Models Applied:**

Support Vector Machine (tuning with best params),

Transformed Target Regression (boxcox),

XGBoost with Transfomed Target Regressor (boxcox)

**Error Metrics for evaluation - RMSE**

**Model Results:**

Transfomed Target Regressor performed better on the Leaderscoreboard

compared to other two models

Github Link: <https://github.com/bhar1122/MLModelling>