Rossman store sales prediction
Project Description

Business Problem

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied.

Data Description

We have been provided data with the following data fields.

1. Id - an Id that represents a (Store, Date) duple within the test set
2. Store - a unique Id for each store
3. Sales - the turnover for any given day (this is what you are predicting)
4. Customers - the number of customers on a given day
5. Open - an indicator for whether the store was open: 0 = closed, 1 = open
6. StateHoliday - indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None
7. SchoolHoliday - indicates if the (Store, Date) was affected by the closure of public schools
8. StoreType - differentiates between 4 different store models: a, b, c, d
9. Assortment - describes an assortment level: a = basic, b = extra, c = extended
10. CompetitionDistance - distance in meters to the nearest competitor store
11. CompetitionOpenSince[Month/Year] - gives the approximate year and month of the time the nearest competitor was opened
12. Promo - indicates whether a store is running a promo on that day
13. Promo2 - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
14. Promo2Since[Year/Week] - describes the year and calendar week when the store started participating in Promo2
15. PromoInterval - describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store

Src : https://www.kaggle.com/c/rossmann-store-sales/data

Aim

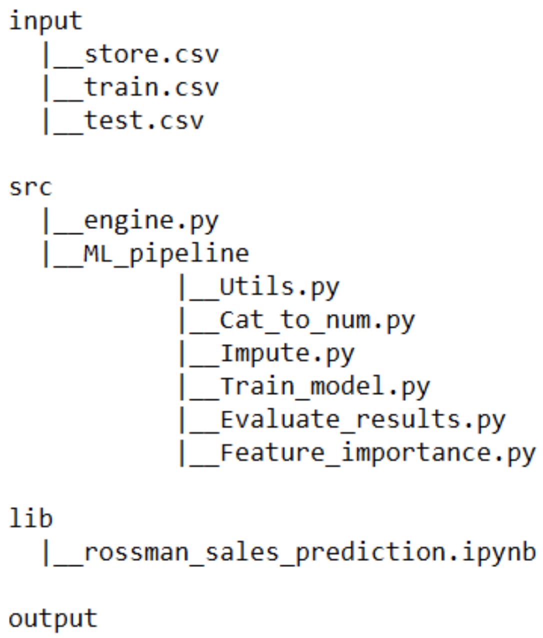
We are provided with historical sales data for 1,115 Rossmann stores. The task is to forecast the "Sales" column.

Approach

1. **Exploratory Data Analysis**
   Exploratory data analysis is the process of analysing the dataset to understand its characteristics. In this step, we will figure out the following.
   a. Identifying the number of unique users
   b. Platforms used by the users the most
   c. Correlations
   d. Checking for null / inconsistent values and various other insights are drawn.
2. **Imputation**
   This step involves the process of filling the missing values in appropriate ways so that the data is not lost.
3. **Outliers Detection and removal**
   Outliers are nothing but points that are abnormally distant from the other points. These kinds of outliers present in the data are detected and eliminated.
4. **Further Exploratory Data Analysis**
   Further EDA is performed to understand the data more and find out a few exceptional cases.
5. **Label Encoding and One hot encoding**
   Machine learning algorithms for regression can understand the input only in the form of numbers and hence it is highly essential to convert the non - numeric data that we have to numeric data by providing them labels.
6. **Model Building and evaluation**
   Various regression algorithms are applied on the dataset and the model that suits best for the dataset is selected. The models that we apply for this dataset are
   a. Linear Regression
   b. SGD Regression
   c. Random Forest Regressor
   d. Decision Tree Regressor
7. **Feature Importance Analysis**
   Once we have the right model selected (which in our case is Random forest regressor as this gives us the best train and test accuracy) it is also essential to understand what are the features that contribute the most or least towards the result.

Modularized code

The ipython notebook is modularized into different functions so that the user can use those functions instantly whenever needed. The modularized code folder is structured in the following way.



```
input
  |__store.csv
  |__train.csv
  |__test.csv

src
  |__engine.py
  |__ML_pipeline
           |__Utils.py
           |__Cat_to_num.py
           |__Impute.py
           |__Train_model.py
           |__Evaluate_results.py
           |__Feature_importance.py

lib
  |__rossman_sales_prediction.ipynb

output
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. lib

1. The input folder contains all the data that we have for analysis. In our case, it will contain a three csv files which are
   a. store.csv
   b. test.csv
   c. train.csv
2. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further contains the following.
   a. ML_pipeline
   b. engine.py

The ML_pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file

3. The output folder contains all the models that we trained for this data saved as .pkl files. These models can be easily loaded and used for future use and the user need not have to train all the models from the beginning.
4. The lib folder is a reference folder. It contains the original ipython notebook that we saw in the videos.

Project Takeaways

1. Understanding the problem statement
2. Data exploration
3. Data Visualization
4. Handling missing values
5. Handling Outliers
6. Exploring exceptional cases
7. Converting categorical to numeric forms
8. Creating heatmaps
9. Feature selection & its importance
10. Implementation using linear regression
11. Implementation using stochastic gradient descent
12. Implementation using random forest
13. Implementation using decision trees
14. Understanding feature importance