**Retail Sales Prediction**

**Souvik Bhattacharyya**

**Data science Trainee,**

**AlmaBetter, Bangalore**

**Abstract:**

Rossmann operates over 3,000 drugstores in seven European countries. This paper outlines a retail sales prediction. The relative importance of consumer demographic characteristics for accurately modeling the sales of each customer type is derived and implemented in the model. The data consisted of daily sales information by store type. The project can help understand what could be the feature that can affect our different store types' sales through feature selection, feature engineering, data analysis, and prediction with machine learning algorithms that take into account previous trends to determine the correct model.

1. **Problem Statement**

Rossmann operates over 3,000 drugstores in seven European countries. Currently, Rossmann store managers are tasked with predicting their daily sales 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 the results can be quite varied.

The main objective is to build a predictive model that could help them predict sales. This would help them estimate the product demand and supply according to customers’ needs in different store types located at different locations.

**The Rossman company provides us with two datasets. Each dataset contains different features.**

* Rossmann Stores Data.csv - historical data including Sales
* store.csv - supplemental information about the stores

**Following are the features inside our two datasets provided by Rossman:**

* + Id - an Id that represents a (Store, Date) duple within the test set.
  + Store - a unique Id for each store
  + Sales - the turnover for any given day (this is what you are predicting)
  + Customers - the number of customers on a given day.
  + Open - an indicator for whether the store was open: 0 = closed, 1 = open.
  + State Holiday - 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.
  + School Holiday - indicates if the (Store, Date) was affected by the closure of public schools
  + Store Type - differentiates between 4 different store models: a, b, c, d.
  + Assortment - describes an assortment level: a = basic, b = extra, c = extended
  + CompetitionOpenSince [Month/Year] - gives the approximate year and month of the time the nearest competitor has opened
  + Promo - indicates whether a store is running a promo on that day.
  + Promo2 - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating

**Following libraries, we will use in our analysis and model building: -**

* *Pandas: - Pandas are for solving data wrangling and exploratory.*
* *NumPy: - NumPy is for numerical problem solving*
* *Matplotlib: - Matplotlib is for Data Visualization*
* *Seaborn: - Seaborn is for Data Visualization*
* *Scikit Learn: - Scikit learn for Machine learning*

1. **Steps involved:**

* **Exploratory Data Analysis**

After loading the datasets, we performed this method by comparing our target variable, which is Sales with other independent variables. This process helped us figure out various aspects, distributions, and relationships between the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

* **Null values Treatment**

Our Store.csv dataset contains a large number of null values, which might tend to disturb the performance of our model. Hence, we dropped those features that had more than 40% null values to get a better result, and for the features that had less than 40% null values, we replaced them using the mean, mode, and median of these features according to their distribution.

* **Encoding of categorical columns**

We used label encoding and dummy encoding to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine learning algorithm and need to be converted to the numerical format.

* **Feature Selection**

We check the correlation of each independent variable with the target variable, i.e., which feature is more important compared to our model and which is of less importance.

* **Standardization of features**

Our main motive in taking this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it. To scale our data, we use the log function, which has large values.

* **Fitting different models**

For modeling, we tried various regression algorithms:

* 1. Liner Regression
  2. Liner Regression with Regularization lasso
  3. Decision Tree

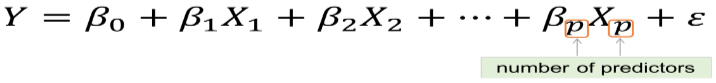
1. **Algorithms:**
2. **Liner Regression**

Linear regression analysis is used to predict the values of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.



where Y is the output or target variable  
X: the input/dependent variable  
β1: Intercept  
β2: constant of X

**Multiple Linear Regression**: It's as simple as its name: it elucidates the connection between the target variable and two or more explanatory variables. Multiple linear regression is used to do any kind of predictive analysis as there is more than one explanatory variable.



**Key Assumptions in the Regression Model:**

• The dependent/target variable is continuous

• There isn’t any relationship between the explanatory/independent variables (no multicollinearity)

• There should be a linear relationship between target/dependent and explanatory variables

• Residuals should follow a normal distribution

• Residuals should have constant variance

• Residuals should be independently distributed/no autocorrelation

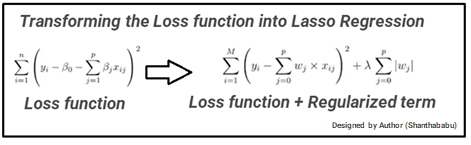
1. **Liner Regression with Regularization lasso**

During the machine learning model-building process, the regularization technique is an unavoidable and important step for improving the model's prediction and reducing errors.

Lasso Regression (L1 Regularization):

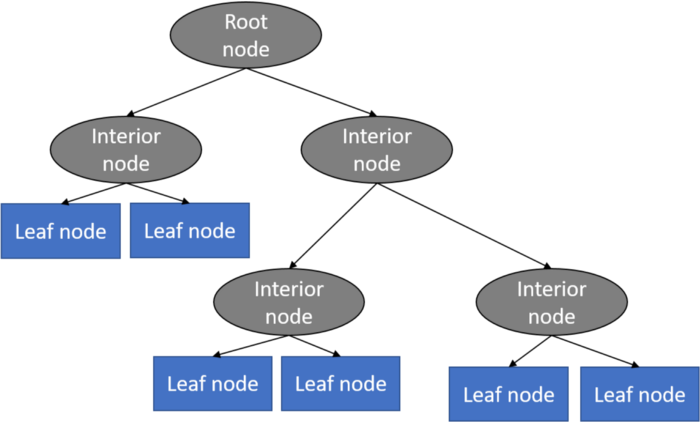
This is very similar to ridge regression, with a little difference in the penalty factor: the coefficient is magnitude instead of squared. In which many coefficients can become zero, corresponding attributes and features become zero and are dropped from the list, ultimately reducing dimensions and supporting dimensionality reduction. So, deciding that those attributes and features are not suitable as predators for predicting target value This is L1 regularization because of the addition of the **absolute value** as a **penalty equivalent**to the magnitude of the coefficients.

***Lasso Regression = Loss function + Regularized term***



1. **Decision Tree**

A Decision Tree is one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification tasks, with the latter having a more practical application.



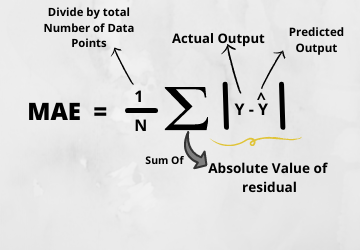
It is a tree-structured classifier with three types of nodes. The root node is the initial node that represents the entire sample and may be subdivided further. The interior nodes represent the features of a data set, and the branches represent the decision rules. Finally, the leaf nodes represent the outcome. This algorithm is very useful for solving decision-related problems

1. **Model performance:**

The model can be evaluated by various metrics, such as

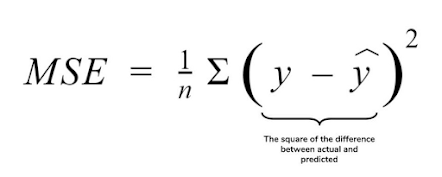
* 1. Mean Absolute Error (MAE)

MAE is a very simple metric that calculates the absolute difference between actual and predicted values.



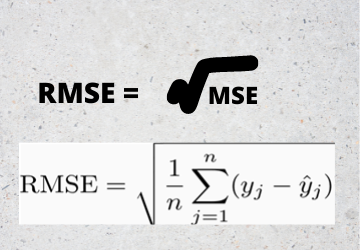
* 1. Mean Squared Error (MSE)

MSE is the most used and simplest metric, with a little bit of change in the mean absolute error. The mean squared error states that finding the squared difference between the actual and predicted value



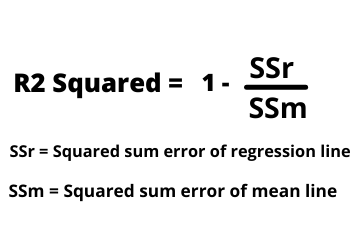
* 1. Root Mean Squared Error (RMSE).

As the name implies, RMSE is a simple square root of the mean squared error.



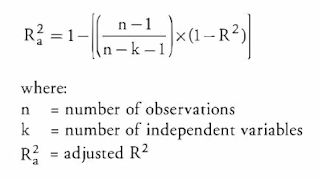
* 1. R Squared (R2)

R2 score is a metric that tells you how well your model performed rather than the loss in absolute terms.



* 1. Adjusted R Squared

The disadvantage of the R2 score is while adding new features in data the R2 score starts increasing or remains constant but it never decreases because it assumes that while adding more data variance of data increases



1. **Conclusion:**

Starting with loading the data so far, we have done EDA, null value treatment, encoding of categorical columns, feature selection, and then model building. After implementing three machine learning models in the dataset to find the six-week prediction, we can see that the decision tree regressor outperforms the other two models, which underperformed due to poor prediction accuracy. In contrast, the Decision Tree regressor has 97% accuracy, which is a decent accuracy score. So, we can deploy this model to solve business problems.