**Retail Sales Prediction**

**Prasanna R Kadlayyanavaramath**

**Data science trainee,**

**AlmaBetter, Bangalore**

**Abstract:**

#### Sales forecasting is the process of estimating future sales. Accurate sales forecasts enable companies to make informed business decisions and predict short-term and long-term performance. Companies can base their forecasts on past sales data, industry-wide comparisons, and economic trends.

#### It is easy to predict the future sales by analyzing the past results. This gives insight into how a company should manage its workforce, cash flow, and resources. In addition to helping a company allocate its internal resources effectively, predictive sales data is important for businesses when looking to acquire investment capital.

Our experiment can help understand what could be the reason for the regression of such labels by feature selection, data analysis and prediction with machine learning algorithms taking into account previous trends to determine the correct

Regression models.

***Keywords:machine learning,Regression models,feature selection,Data wrangling***

**1.Problem Statement**

In this project, We are looking to forecast the rossmann store data. Here, we are tasked with predicting their daily sales in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of sales records, we will be predicting sales based on their unique circumstances, the accuracy of results can be quite varied and the task is to forecast the “**Sales**” column.

The main objective is to build a predictive model, which could help them in predicting the “Sales” proactively. This would in turn help them to make more revenue to the stores to run quickly and efficiently.

* **Id** – an Id that represents a Store
* **Store** – a unique Id for each store
* **Sales** – the turnover for any given day (this is what we are predicting)
* **Customers** – the number of customers on a given day
* **Open** – an indicator for whether the store was open: 0 = closed, 1 = open
* **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
* **SchoolHoliday** – indicates if the (Store) was affected by the closure of public schools
* **StoreType** – differentiates between 4 different store models: a, b, c, d
* **Assortment** – describes an assortment level: a = basic, b = extra, c = extended
* **CompetitionDistance** – distance in meters to the nearest competitor store
* **CompetitionOpenSince[Month/Year]** – gives the approximate year and month of the time the nearest competitor was 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
* **Promo2Since[Year/Week]** – describes the year and calendar week when the store started participating in Promo2
* **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

**2. Introduction**

This project on sales forecasting or sales prediction has always been a very significant area to concentrate upon. An efficient and optimal way of forecasting has become essential for all the vendors in order to sustain the efficacy of the marketing organizations. Manual infestation of this task could lead to drastic errors leading to poor management of the organization, and most importantly would be time consuming, which is something not desirable in this expedited world. This is where machine learning can be exploited in a great way.

### Our goal here is to build a predictive model, which could help rossmann store to predicting their daily sales for up to six weeks in advance.

**3. Steps involved:**

## **3.1 Summary of Data**

The dataset consists of (1017209 rows × 18 columns) after merging two given datasets. In this dataset the data is provided for training tags the stores into four different categories. Sale data, the “target variable” of every store is provided along with various features like the distance of the nearest competitor, promotional activities and number of customers to name a few. Data provided is a mixed set of continuous and discrete variables.

* 1. **Exploratory Data Analysis**

After loading the dataset we performed this method by comparing our target variable that is ‘Sales’ with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

**3.3 Null values Treatment**

Our dataset contains a large number of null values which might tend to disturb our accuracy hence we dropped them at the beginning of our project inorder to get a better result.

We replaced the Nan values of CompetitionDistance  with median,

CompetitionOpenSinceMonth and CompetitionOpenSinceYear with mode.Promo2SinceWeek,Promo2SinceYear,PromoInterval are NaN wherever Promo2 is 0 or False They can be replaced with 0.

**3.4 Encoding of categorical columns**

I used One Hot 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 and needs to be converted to numerical format.

**3.5 Feature Selection**

In these steps we used algorithms like regression models and hyperparameter tuning to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

By using OLS models we come to know that the null hypothesis can be rejected with evidence which we will be using further in our model.

**3.6 Standardization of features**

The features are not uniform so, we need to transform and standardize the data. Our main motive through 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.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

* 1. **Fitting different models**

For modelling we tried various regression algorithms like:

1. **Linear Regression**
2. **Lasso and Ridge Regression**
3. **Decision Tree and Elastic Net**
4. **Random Forest**
5. **Gradient Boosting**

**3.8 Tuning the hyperparameters for better accuracy**

Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree based models

like Decision tree and other models. So, using Grid SearchCv we can tune the parameters for better accuracy.

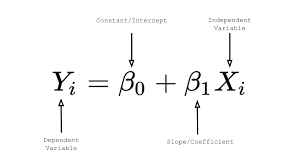
**3.9 SHAP Values for features**

I have applied SHAP value plots on the Random Forest model to determine the features that were most important while model building and the features that didn’t put much weight on the performance of our model.

**4. Algorithms:**

**4.1 Linear Regression:**

Regression models describe the relationship between variables by fitting a line to the observed data. Linear regression models use a straight line.Linear regression uses a linear approach to model the relationship between independent and dependent variables. In simple words its a best fit line drawn over the values of independent variables and dependent variable. In case of single variable, the formula is same as straight line equation having an intercept and slope.



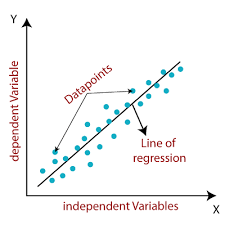
y\_pred = β0+β1x

where β0 and β1 are intercept and slope respectively.

In case of multiple features the formula translates into:

y\_pred=β0+β1x1+β2x2+β3x3+.....

where x1,x2,x3 are the features values and β0,β1,β2..... are weights assigned to each of the features. These become the parameters which the algorithm tries to learn using Gradient descent.



our r2 score value for train and test data is 0.82 that means our model is able to capture most of the data variance.

.

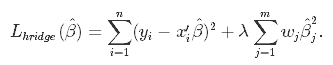
**4.2 Lasso and Ridge Regression:**

This is a regularization technique used in feature selection using a Shrinkage method also referred to as the **penalized regression method**.

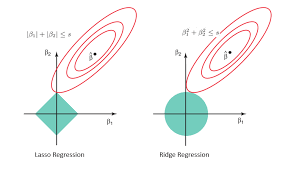
Lasso is short for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator, which is used both for regularization and model selection. If a model uses the **L1 regularization** technique, then it is called lasso regression.



Similar to the lasso regression, **ridge regression** puts a similar constraint on the coefficients by introducing a penalty factor. However, while lasso regression takes the magnitude of the coefficients, ridge regression takes the square.



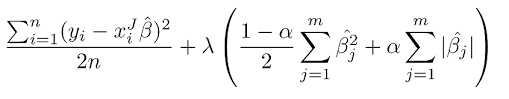
Ridge regression is also referred to as **L2 Regularization**.



The r2\_score for the Lasso(L1)train and test set is 0.71. This means our linear model is not performing well on the data and for Ridge(L2) it is 0.82 bit well on this model.

**4.3 Elastic Net Regression:**

**Elastic net** is a popular type of regularized linear regression that combines both L1 and L2 models.

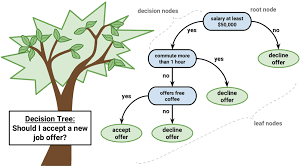


The r2\_score for the test set is 0.78. This means our linear model is performing well on the data.

**4.4 Decision Tree:**

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**.

It works by **splitting the data up in a tree-like pattern into smaller and smaller subsets**. Then, when predicting the output value of a set of features, it will predict the output based on the subset that the set of features falls into.



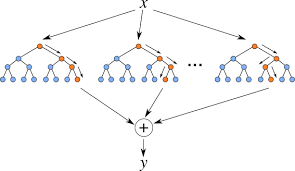
The r2\_score for our train and test set is 0.85. This means our linear model is performing well on the data.

**4.5 Random Forest:**

Random forest is a **supervised learning algorithm** that uses an ensemble learning method for **classification and regression**.

**Random forest** is a **bagging** technique and **not a boosting** technique. The trees in **random forests** run in parallel, meaning is no interaction between these trees while building the trees.

Random forest operates by constructing a multitude of decision trees at training time and outputting the class that’s the **mode** of the **classes (classification)** or **mean prediction (regression)** of the individual trees.



The r2\_score for the train and test set is 0.95. This means our linear model is extremely performing well on the data comparing with other models.

**4.6 Gradient Boosting(GBM):**

Gradient boosted trees consider the special case where the simple model is a decision tree

****

In this case, there are going to be 2 kinds of parameters P: the weights at each leaf, w, and the number of leaves T in each tree (so that in the above example, T=3 and w=[2, 0.1, -1]).

When building a decision tree, a challenge is to decide how to split a current leaf. For instance, in the above image, how could I add another layer to the (age > 15) leaf? A ‘greedy’ way to do this is to consider every possible split on the remaining features (so, gender and occupation), and calculate the new loss for each split; you could then pick the tree which most reduces your loss.

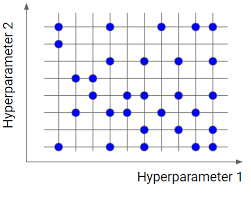
The r2\_score for the train and test set is 0.89. This means our linear model is performing well on the data.

**5. Hyper parameter tuning:**

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects performance, stability and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs.

I used Grid Search CV for hyperparameter tuning. This also results in cross validation and in our case we divided the dataset into different folds.

1. **Grid Search CV-**Grid Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model’s performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.



The r2\_score for the test set is 0.88. This means our linear model is performing well on the data.

**6. Conclusion:**

That's it! We reached the end of our exercise.

Starting with loading the data so far we have done EDA , null values treatment, encoding of categorical columns, feature selection and then model building.

1. Sales is highly correlated to number of Customers.
2. For all stores, Promotion leads to increase in Sales and Customers both.
3. During starting days of the week like Monday,Tuesday, wednesday, friday the sales got a positive growth.But due to holiday on sunday completely it leads to zero.It gives the negative growth. So, Keeping store open on starting days of the week is must to grow the business. in peak level.
4. The most selling and crowded store type is A and D.Store type b is the least one.
5. More stores are opened during School holidays than State holidays. Sales are increased during normal days than the public holidays.
6. Highest average sales were seen with Assortment levels-a which is 'basic'. We can see the drop of sales on 'b'(extra) assortment.
7. During the initial consecutive years we could see positive impact on sales due to promotions.
8. Day of the week has a negative correlation indicating low sales as the weekends, and promo, customers and open has positive correlation.
9. CompetitionDistance showing negative correlation suggests that as the distance increases sales reduce, which was also observed through the plot earlier.

Next, we implemented 7 machine learning algorithms Linear Regression,lasso,ridge,elasticnet,decission tree, Random Forest and GradientBoosting. We did hyperparameter tuning to improve our model performance. The results of our evaluation are:

1. No overfitting is seen.
2. Random forest Regressor and Gradient Boosting gridsearchcv gives the highest R2 score of 96% and 88% recpectively for Train Set and 89% for Test set.
3. Feature Importance value for Random Forest and Gradient Boost are almost same.
4. We are able to see the importance of the features and how it is effecting to dependent variable. We can ignore the 'customers' feature because, The task is not to predict the number of customers, so fitting a model based on the given variables in the test set to predict the number of customers surely makes no sense. But features like 'CompetitionDistance','Promo','StoreType','DayOfWeek' and 'Assortment' are very important features where dependent variable depends on these features.
5. We can deploy this model.

Random forest Regressor and Gradient Boosting gridsearchcv gives the highest R2 score of 96% and 88% recpectively for Train Set and 89% for Test set.So, We are taking these ML Models for future prediction of the sales.

**References-**

1. MachineLearningMastery
2. GeeksforGeeks
3. Analytics Vidhya