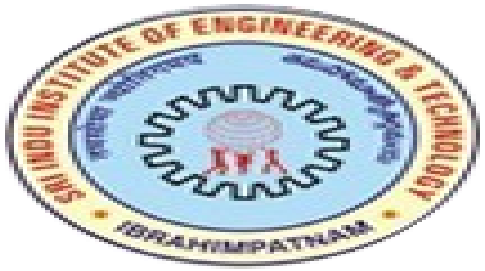


Sri Indu

Institute of Engineering & Technology

Institution, Recognized under 2(f) of UGC Act 1956.
Approved by AICTE, New Delhi. Affiliated to JNTUH, Hyderabad.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINI PROJECT ON:

BIGMART SALES PREDICTION USING MACHINE LEARNING

PRESENTED BY:

- A.R. FAISAL – 19X31A0502(TEAM LEADER)
- G. SRIMANJUNATH – 19X31A0540
- D. GANESH – 19X31A0532
- H. BABU – 19X31A0558

GUIDED BY:

MRS. SRUTHI

(ASSISTANT PROFESSOR)

CONTENTS:

ABSTRACT:

Everybody wants to know how to buy goods cheaper or how to advertise them at low cost. Here is the answer. That is Big Mart. Big Mart is Online one stop marketplace where you can buy or sell or advertise your merchandise at low cost. The goal is to make Big Mart the shopping paradise for buyers and the marketing solutions for the sellers. The ultimate goal is to prosper with customers. The project "**BIGMART SALES DATASET**" aims to build a predictive model and find out the sales of each product at a particular store. Big Mart will **use this model to understand the properties of products and stores which play a key role in increasing sales.** This can also be done based on the hypothesis that should be done before looking at the data.

KEYWORDS:

BIGMART SALES DATASET, Predictive Model

INTRODUCTION:

- With the rapid development of global malls and stores chains and the increase in the number of electronic payment customers, the competition among the rival organizations is becoming more serious day by day.
- Each organization is trying to attract more customers using personalized and short-time offers which makes the prediction of future volume of sales of every item an important asset in the planning and inventory management of every organization, transport service, etc.
- Due to the cheap availability of computing and storage, it has become possible to use sophisticated machine learning algorithms for this purpose.
- In this project, we are providing forecast for the sales data of big mart in a number of big mart stores across various location types which is based on the historical data of sales volume.

EXISTING SYSTEM:

- Predicting the sales of each product is not a simple task. Prediction of sales depends on a number factors such as product demand, product supply and many other things. So seller has to himself study all of these things and himself predict the sales.
- Unfortunately, information about the sales are not always accurate and based on that the seller wrongly predicts the sales and thereby suffers.
- Existing System includes a process where a seller decides to keep the product based on inaccurate data and thereby, the consumer also suffers with the seller. And the existing system is also slow as it is not automated and the prediction of sales is to be done by the seller himself.

DISADVANTAGES:

- Not accurate
- Not efficient
- Inflexible

PROPOSED SYSTEM:

- To use various technologies for model building like multiple linear regression analysis and random forest to forecast the sales volume.
- The aim is to build a predictive model and find out the sales of each product at a particular store.
- Using this model, Big Mart will try to understand the properties of products and stores which play a key role in increasing sales.

ADVANTAGES:

- Saves money and time
- Scalable
- Helps in gaining the trust of customers

MODULES:

➢ LINEAR REGRESSSOR

- Linear Regression is usually the first algorithm that people learn for Machine Learning and Data Science. Linear Regression is a linear model that assumes a linear relationship between the input variables (X) and the single output variable (y). In general, there are two cases:

- **Single Variable Linear Regression:** it models the relationship between a single input variable (single feature variable) and a single output variable.
- **Multi-Variable Linear Regression:** (also known as Multivariate Linear Regression), it models the relationship between multiple input variables (multiple features variables) and a single output variable.

➢ RANDOM FOREST REGRESSOR

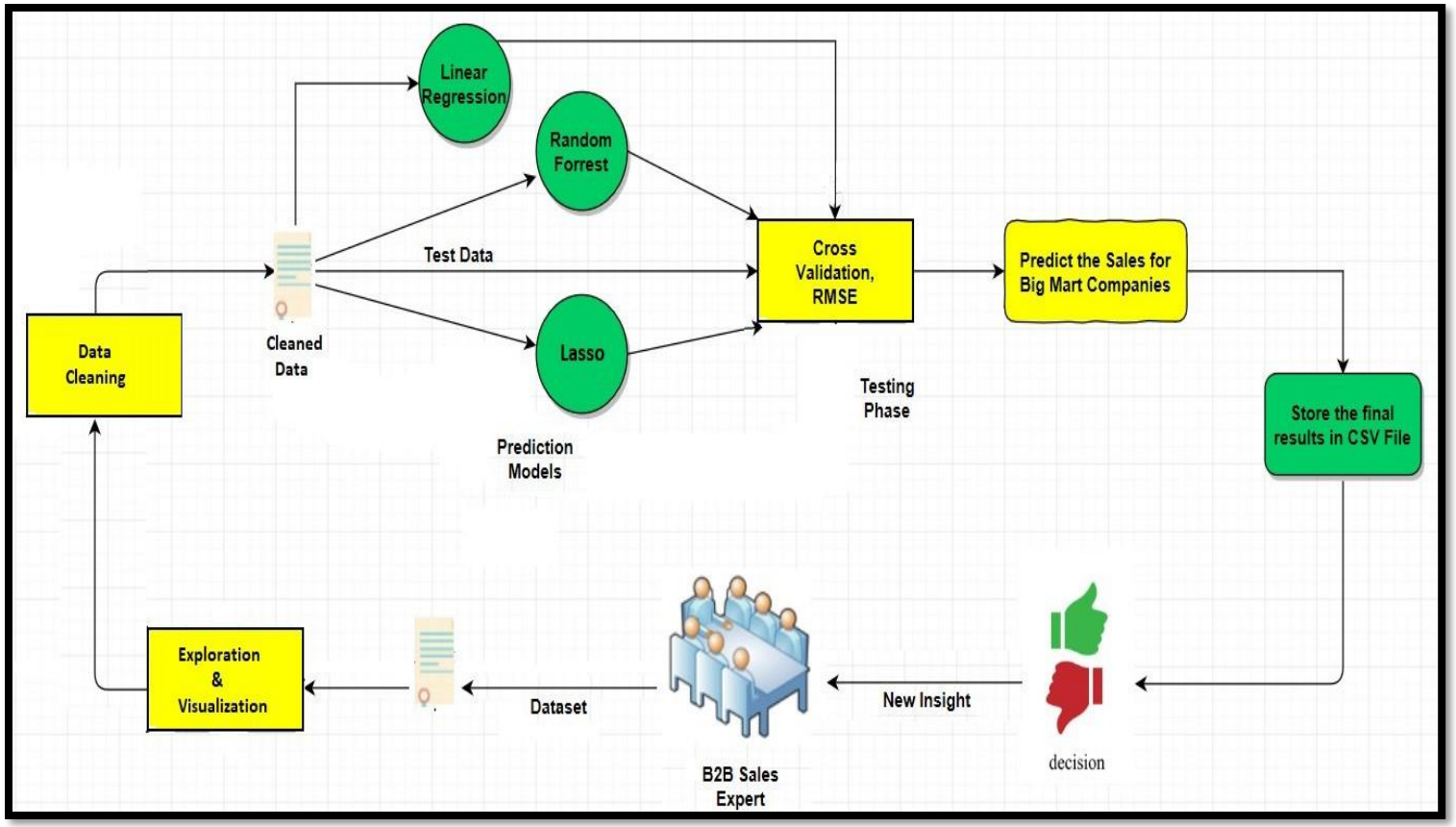
- Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.
- Random Forest Regression is very similar to Decision Tree Regression, Basically. it's a meta estimator that fits a number of Decision Trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- A Random Forest regressor may or may not perform better than the Decision Tree in regression (while it usually performs better in classification), because of the delicate overfitting-underfitting tradeoff in the nature of tree-constructing algorithms.

➢ LASSO REGRESSOR

- LASSO regression is a variation of Linear Regression that uses Shrinkage. Shrinkage is a process that data values are shrunk towards a central point as the mean. This type of regression is well-suited for models showing heavy multicollinearity (heavy correlation of features with each other) or when you want to automate certain parts of model selection, like variable selection/parameter elimination..
- The word “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. It is a statistical formula for the regularisation of data models and feature selection.
- The LASSO regression in regularization is based on simple models that posses fewer parameters. Regularization resolves the overfitting problem, which affects the accuracy level of the model. Regularization is executed by the addition of the “penalty” term to the best-fit equation produced by the trained data.

SYSTEM ARCHITECTURE:

- Data understanding and exploration
- Data Visualization
- Data Cleaning
- Predicting Models



HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4Ghz.
- Hard Disk : 20GB.
- Floppy Drive : 1.44Mb.
- Monitor : 14' Colour Monitor.
- Mouse : Optical Mouse.
- RAM : 512Mb.

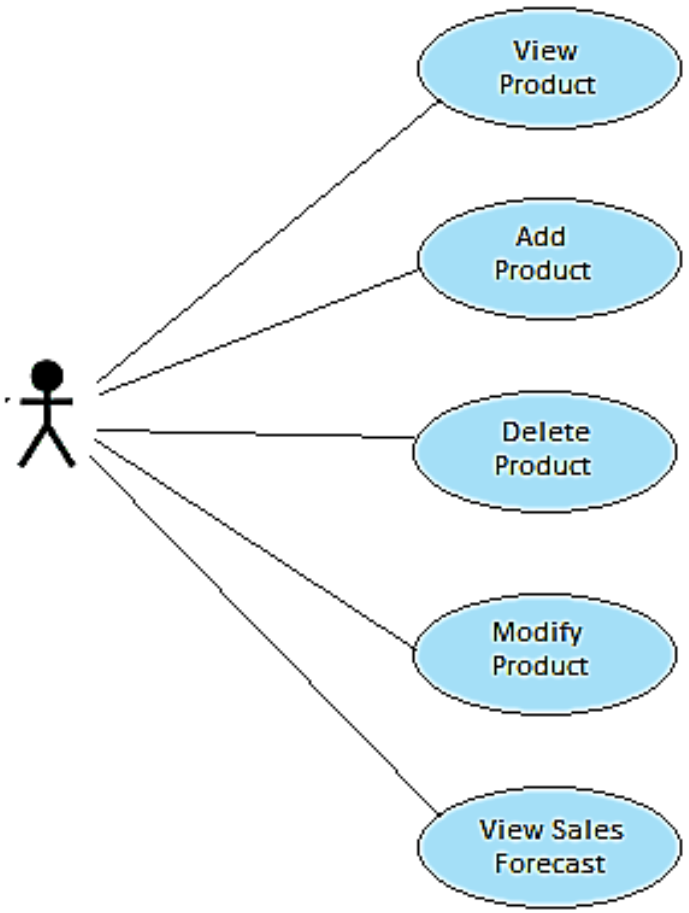
SOFTWARE REQUIREMENTS:

- Operating System : Windows 7 ultimate.
- Coding Language : Python.
- Front-End : Python.
- Designing : HTML, CSS, JavaScript.
- DataBase : MySQL.

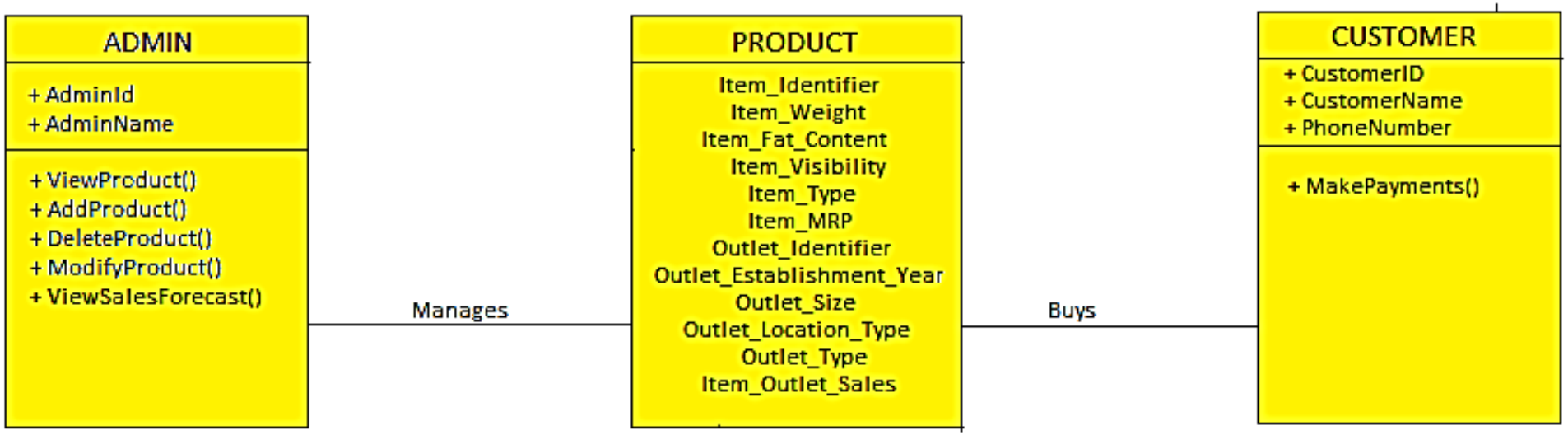
UML DIAGRAMS:

- **USE CASE DIAGRAM**

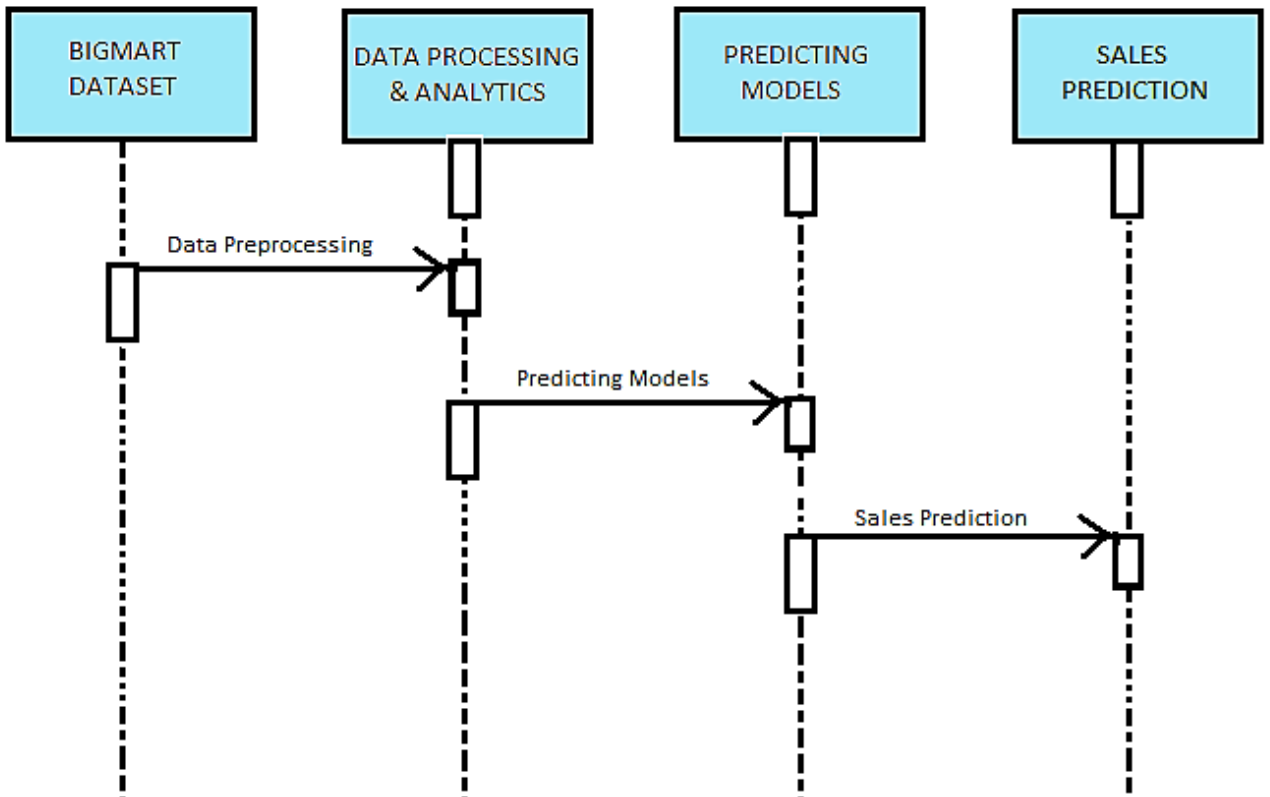
ADMIN:



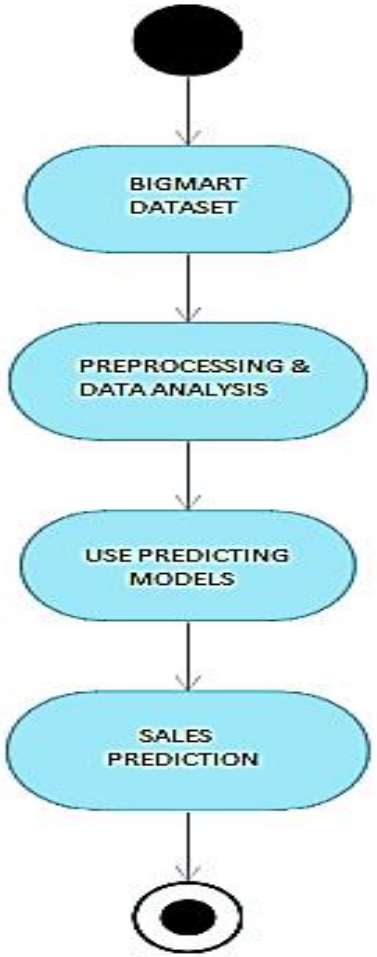
➢ CLASS DIAGRAM:



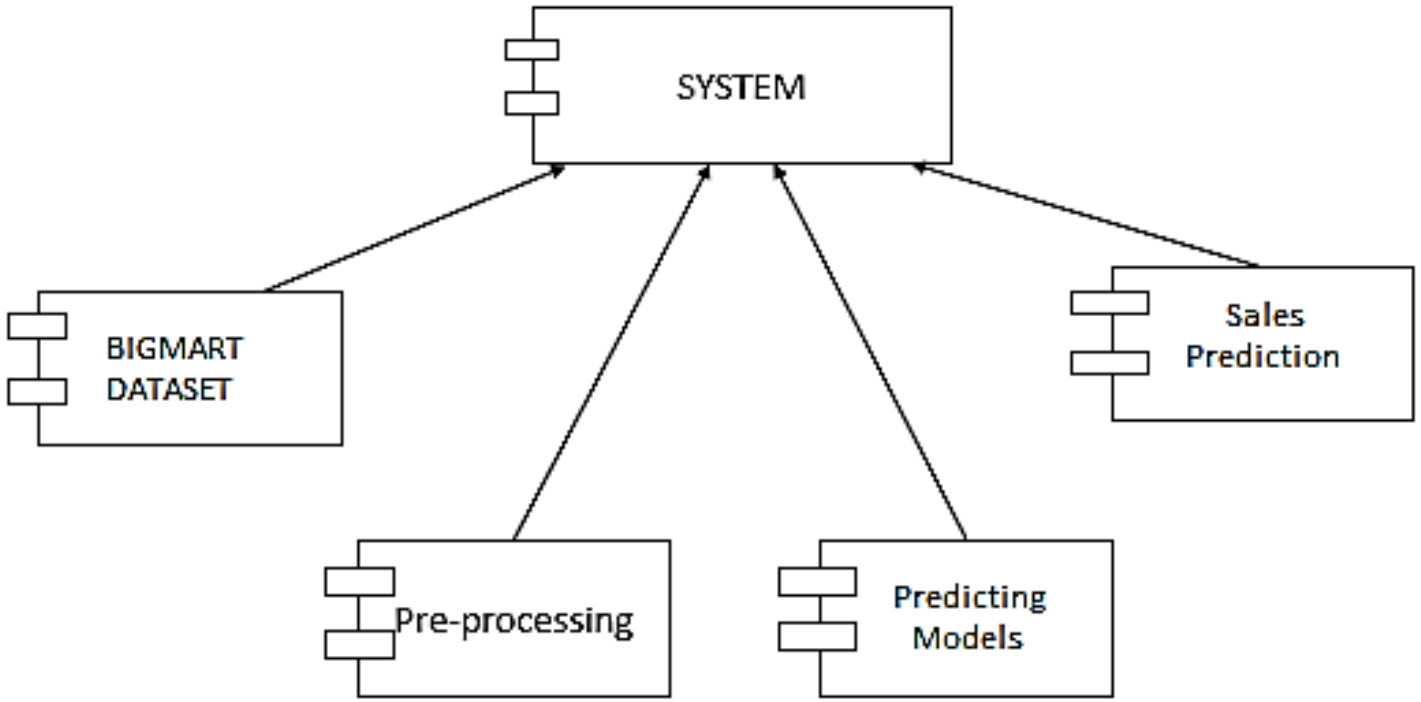
➢ SEQUENCE DIAGRAM:



➢ ACTIVITY DIAGRAM:



➢ COMPONENT DIAGRAM:



IMPLEMENTATION:

➢ Python:

- Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together.
- Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse.
- The Python interpreter and the extensive standard library are available in source(open-source) or binary form without charge for all major platforms, and can be freely distributed.
- Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms object-oriented, imperative, functional and procedural, and has a large number of comprehensive standard library.

➢ Keras:

- Keras is an open-source high-level Neural Network Library, which is written in Python is capable enough to run on Theano, TensorFlow, or CNTK. It cannot handle low-level computations, so it makes use of the Backend library to resolve it. The Backend library act as a high-level API wrapper for the low-level API, which lets it run on the TensorFLow, CNTK, or Theano.
- Keras can be developed in R as well as Python, suct that the code can be run with TensorFlow, Theano, CNTK, or MXNet as per the requirements. Keras can run on CPU, NVIDEA, AMD GPU, TPU, etc.
- It ensures that producing models with Keras is really simple as it totally supports to run with TensorFlow serving, GPU acceleration(WebKeras, Keras.js), Android(TF,TF Lite), iOS(Native CodeML) and Raspberry Pi.

➢ Tensor Flow:

- Tensorflow is a library that is used in machine learning and it is an open-source library for numerical computations.
- It is used for developing machine learning applications and this library was first created by the Google brain team and it is the most common and successfully used library that provides various tools for machine learning applications.
- TensorFlow library is used in many companies in the industries like Airbnb. This company applies machine learning using TensorFlow to detect objects and classify the

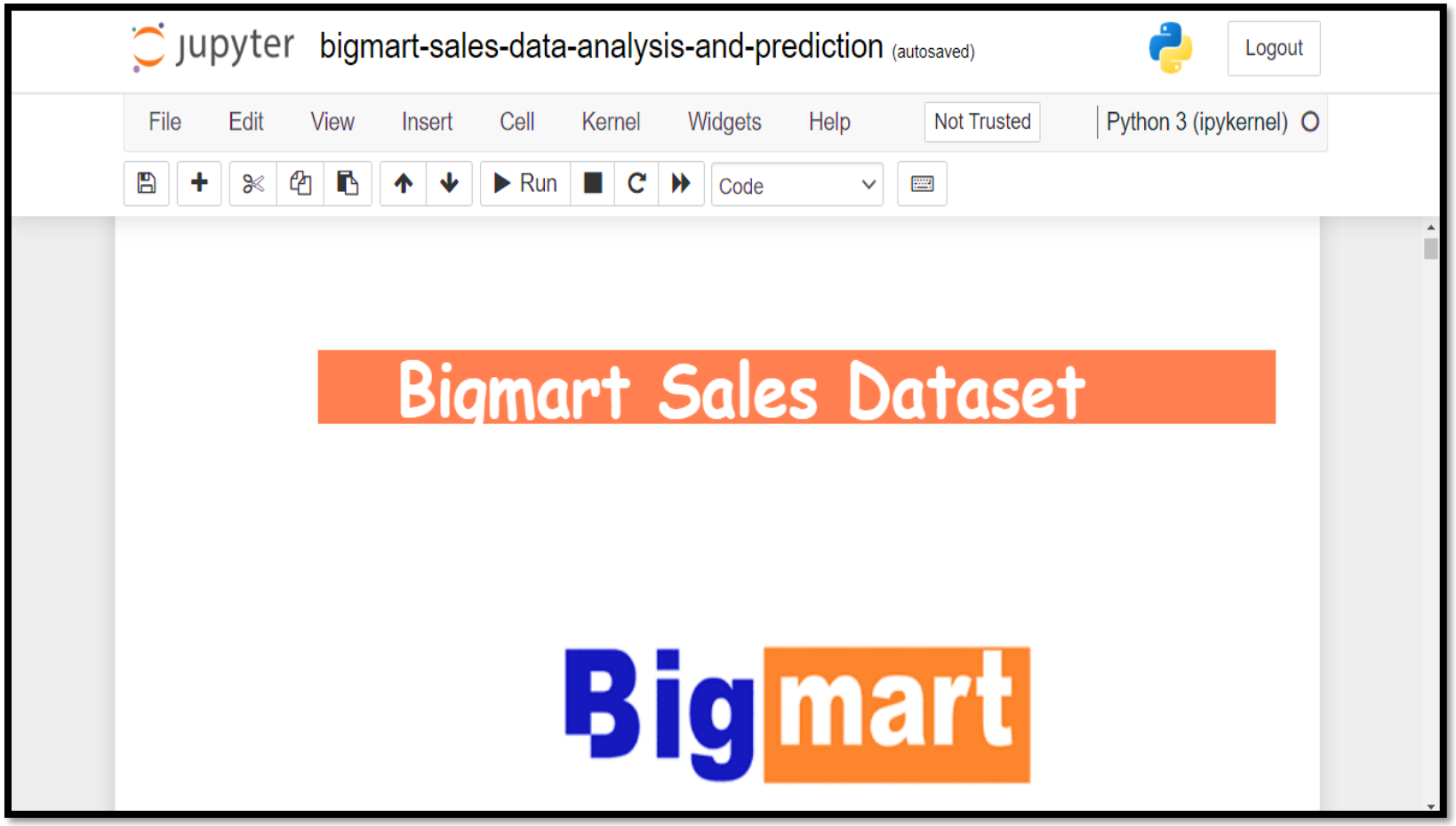
SAMPLE CODE:

```
#FOR LINEAR REGRESSION
LR = LinearRegression(normalize=True)  #model

LR.fit(X_train, y_train)  #fit

y_predict = LR.predict(X_test)  #predict

LR_MAE = round(MAE(y_test, y_predict),2)  #score variables
LR_MSE = round(MSE(y_test, y_predict),2)
LR_R_2 = round(R2(y_test, y_predict),4)
LR_CS  = round(CVS(LR, X, y, cv=5).mean(),4)

print(f" Mean Absolute Error: {LR_MAE}\n")
print(f" Mean Squared Error: {LR_MSE}\n")
print(f" R^2 Score: {LR_R_2}\n")
cross_val(LR,LinearRegression(),X,y,5)
```

```
#FOR RANDOM FOREST REGRESSION
RFR=RandomForestRegressor(n_estimators=200,max_depth=5,
min_samples_leaf=100,n_jobs=4,random_state=101)  #model

RFR.fit(X_train, y_train)  #fit

y_predict = RFR.predict(X_test) #predict

RFR_MAE = round(MAE(y_test, y_predict),2) #score variables
RFR_MSE = round(MSE(y_test, y_predict),2)
RFR_R_2 = round(R2(y_test, y_predict),4)
RFR_CS  = round(CVS(RFR, X, y, cv=5).mean(),4)

print(f" Mean Absolute Error: {RFR_MAE}\n")
print(f" Mean Squared Error: {RFR_MSE}\n")
print(f" R^2 Score: {RFR_R_2}\n")
cross_val(RFR,RandomForestRegressor(),X,y,5)
```

```
#FOR LASSO REGRESSION
LS = Lasso(alpha = 0.05)  #model

LS.fit(X_train,y_train)  #fit

y_predict = LS.predict(X_test) #predict

LS_MAE = round(MAE(y_test, y_predict),2)  #score variables
LS_MSE = round(MSE(y_test, y_predict),2)
LS_R_2 = round(R2(y_test, y_predict),4)
LS_CS  = round(CVS(LS, X, y, cv=5).mean(),4)

print(f" Mean Absolute Error: {LS_MAE}\n")
print(f" Mean Squared Error: {LS_MSE}\n")
print(f" R^2 Score: {LS_R_2}\n")
cross_val(LS,Lasso(alpha = 0.05),X,y,5)
```

OUPUT SCREENS:



preprocessing of the training dataset

In [6]:
```
#column information
tr_df.info(verbose=True, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8523 entries, 0 to 8522
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            8523 non-null   object 
 1   Item_Weight                7060 non-null   float64
 2   Item_Fat_Content           8523 non-null   object 
 3   Item_Visibility            8523 non-null   float64
 4   Item_Type                  8523 non-null   object 
 5   Item_MRP                   8523 non-null   float64
 6   Outlet_Identifier          8523 non-null   object 
 7   Outlet_Establishment_Year  8523 non-null   int64  
 8   Outlet_Size                6113 non-null   object 
 9   Outlet_Location_Type       8523 non-null   object 
 10  Outlet_Type                8523 non-null   object 
 11  Item_Outlet_Sales          8523 non-null   float64
dtypes: float64(4), int64(1), object(7)
memory usage: 799.2+ KB
```

In [7]:
```
#summary statistics test
te_df.describe()
```

Out[7]:

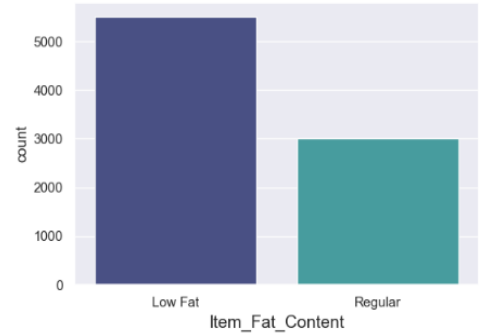
	Item_Weight	Item_Visibility	Item_MRP	Outlet_Establishment_Year
count	4705.000000	5681.000000	5681.000000	5681.000000
mean	12.695633	0.065684	141.023273	1997.828903
std	4.664849	0.051252	61.809091	8.372256
min	4.555000	0.000000	31.990000	1985.000000
25%	8.645000	0.027047	94.412000	1987.000000
50%	12.500000	0.054154	141.415400	1999.000000

Data Visualization 📊

Univariate Plots

For starters we will create countplots for the categorical columns:

In [20]:
```
#categorical columns:
['Item_Identifier', 'Item_Fat_Content', 'Item_Type', 'Outlet_Identifier',
 'Outlet_Size', 'Outlet_Location_Type', 'Outlet_Type']

plt.figure(figsize=(6,4))
sns.countplot(x='Item_Fat_Content' , data=tr_df ,palette='mako')
plt.xlabel('Item_Fat_Content', fontsize=14)
plt.show()
```



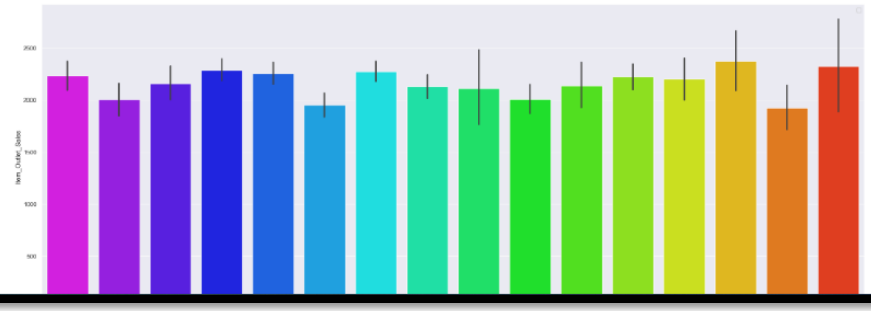
multivariate plots

I want to check the following relationships with `Item_Outlet_Sales`:

- Sales per item type
- Sales per outlet
- Sales per outlet type
- Sales per outlet size
- Sales per location type

In [28]:
```
plt.figure(figsize=(27,10))
sns.barplot('Item_Type' ,'Item_Outlet_Sales', data=tr_df ,palette='gist_rainbow_r')
plt.xlabel('Item_Type', fontsize=14)
plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label start with an underscore are ignored when legend() is called with no argument.
```



Corraltion Matrix

In [33]:
```
#plotting the correlation matrix
sns.heatmap(tr_df.corr() ,cmap='rocket')
```

Out[33]: <AxesSubplot:>

jupyter bigmart-sales-data-analysis-and-prediction (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [47]:
```
MAE= [LR_MAE,RFR_MAE,LS_MAE]
MSE= [LR_MSE,RFR_MSE,LS_MSE]
R_2= [LR_R_2,RFR_R_2,LS_R_2]
Cross_score= [LR_CS,RFR_CS,LS_CS]

Models = pd.DataFrame({
    'models': ["Linear Regression","Random Forest Regressor","Lasso Regressor"],
    'MAE': MAE, 'MSE': MSE, 'R^2':R_2, 'Cross Validation Score':Cross_score})
Models.sort_values(by='MAE', ascending=True)
```

Out[47]:

	models	MAE	MSE	R^2	Cross Validation Score
2	Lasso Regressor	838.07	1285554.86	0.5594	0.5581
0	Linear Regression	838.20	1285809.57	0.5593	0.5580
1	Random Forest Regressor	1030.27	1964025.66	0.3268	0.5920

RESULT:

	models	MAE	MSE	R^2	Cross Validation Score
2	Lasso Regressor	838.07	1285554.86	0.5594	0.5581
0	Linear Regression	838.20	1285809.57	0.5593	0.5580
1	Random Forest Regressor	1030.27	1964025.66	0.3268	0.5920

- Here, we compare different models of regressions, i.e., Lasso regressor, Linear regressor and the Random Forest Regressor. From the above table, we see that Mean Absolute Error(MAE) and the Mean Squared Error(MSE) is the Most in the Random Forest Regressor when compared to the Linear Regressor and the Lasso Regressor.
- Here the R^2 (coefficient of determination) which is the regression score function is lower of the random forest regressor when compared to the Lasso regressor and the Linear Regressor.
- Linear Regression and Lasso Regressor have the best perfomance in most categories. The performance of the Random Forest is not optimal even though his cross validation is the highest.

DISCUSSIONS:

- Here, from the result, we can see that the Linear Regressor and the Lasso Regressor can be to used in our Big MART sales application for the prediction of the sales. Further we noted that by using the Linear Regressor and the Lasso Regressor, the accuracy of the prediction of sales also increased when compared to the Random Forest Regressor.
- By using this data prediction models, we are successful in predicting the sales of the products in the particular stores which helps both the Stores manager and the Customer.

CONCLUSION:

- Here, we describe the basics of machine learning and the associated data processing and modeling algorithms, followed by their application for the task of sales prediction in Big Mart shopping centers at different locations.
- On implementation, the prediction results show the correlation among different attributes considered and how a particular location of medium size recorded the highest sales, suggesting that other shopping locations should follow similar patterns for improved sales.
- Multiple instances parameters and various factors can be used to make this sales prediction more innovative and successful. Accuracy, which plays a key role in prediction-based systems, can be significantly increased as the number of parameters used are increased.
- Linear Regression and Lasso Regressor have the best perfomance in most categories when compared to the Random Forest Regressor.

SCOPE OF FUTURE WORK:

- The project can be further collaborated in a web-based application or in any device supported with an in-built intelligence by virtue of Internet of Things (IoT), to be more feasible for use.
- Various stakeholders concerned with sales information can also provide more inputs to help in hypothesis generation and more instances can be taken into consideration such that more precise results that are closer to real world situations are generated.
- Moreover, the flexibility of the proposed approach can be increased with variants at a very appropriate stage of regression model-building.
- There is a further need of experiments for proper measurements of both accuracy and resource efficiency to assess and optimize correctly.

Thank You