A

Project Report On

“Predictive Analysis for Big Mart Sales Using Machine Learning Algorithms”

Submitted to

Osmania University, Hyderabad

In partial fulfillment of the requirements for the award of degree

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

By

SYED SUBAAN SHAREEF	**245219733023**
MOHD SAIFUDDIN	**245219733019**
KHAN MOHMMED SOHEL	**245219733022**

Under the guidance of

Mrs. B. Shobini M.tech

Head of the Department, C.S.E, SWITS

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SWATHI INSTITUTE OF TECHNOLOGY AND SCIENCES

(Affiliated to Osmania University, Hyderabad)

2022-23

SWATHI INSTITUTE OF TECHNOLOGY AND SCIENCES

(Affiliated to Osmania University, Hyderabad)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

` This is to certify that the project entitled **"Predictive Analysis for Big Mart Sales Using Machine Learning Algorithms"** submitted by **SYED SUBAAN SHAREEF (245219733023), MOHD SAIFUDDIN (245219733019), KHAN MOHMMED SOHEL (245219733022)**, Department of Computer Science and Engineering, Swathi Institute of Technology & Sciences, affiliated to Osmania University in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering, with Computer Science and Engineering as specialization is a record of the bonafide work carried out during the academic 2022- 2023.

The results embodied in this project work have not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of the guide **Signature of External**

B. Shobini (H.O.D)

DECLARATION

We hereby declare that the project work entitled **"Predictive Analysis for Big Mart Sales Using Machine Learning Algorithms"** submitted to the Department of Computer Science and Engineering of **SWATHI INSTITUTE OF TECHNOLOGY AND SCIENCES,** affiliated to Osmania University, Hyderabad in the partial fulfillment of the requirement for award of the degree in BACHELOR OF ENGINEERING in CSE is a bonafide work done by the undersigned.

SYED SUBAAN SHAREEF – 245219733023
MOHD SAIFUDDIN – 245219733019
KHAN MOHAMMED SOHEL – 245219733022

ACKNOWLEDGMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is great pleasure to convey our profound sense of gratitude to the Principal **Dr. K. Regin Bose**, Swathi Institute of Technology & Sciences, for being kind enough for arranging necessary facilities for executing the project in the college.

We deem it a pleasure to acknowledge our sense of gratitude to our internal project guide **Mrs. B. SHOBINI**, CSE under whom we have carried out the project work. Her incisive and objective guidance and timely advice encouraged us with constant flow of energy to continue the work.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard

With Regards,

SYED SUBAAN SHAREEF (245219733023)
MOHD SAIFUDDIN (245219733019)
KHAN MOHAMMED SOHEL (245219733022)

INDEX

LIST OF FIGURES

ABSTRACT

Currently, supermarket run-centred, known as Big Marts, maintain track of each individual item's sales data in order to forecast future consumer demand and adjust inventory management. Mining the data warehouse's data storage is a popular way to find anomalies and general trends. The generated data can be utilize by retailers like Big Mart to anticipate future sales volume using a variety of machine learning approaches. For projecting the sales of a business such as Big -Mart, a predictive model was created utilizing Xgboost, Linear regression, Polynomial regression, and Ridge regression approaches, and it was revealed that the model beats existing models.

The main objective of this project was to develop an accurate sales prediction model for Big Mart supermarkets. To achieve this, various machine learning algorithms were employed, including Xgboost, Linear regression, Polynomial regression, and Ridge regression. These algorithms were selected due to their proven effectiveness in handling large datasets and their ability to capture complex relationships within the data.

The project utilized a comprehensive dataset containing historical sales data, including information such as product attributes, store locations, and time periods. This dataset was extensively preprocessed and cleaned to ensure data quality and consistency. Feature engineering techniques were applied to extract relevant features and create a robust set of predictors for the sales prediction model.

The machine learning models were trained using a portion of the dataset, and their performance was evaluated using appropriate evaluation metrics such as root mean squared error (RMSE) and mean absolute error (MAE). The models were fine-tuned and optimized to improve their predictive accuracy, with hyperparameter tuning techniques being applied to find the optimal configuration for each algorithm.

The results demonstrated that the developed sales prediction model outperformed existing models. The Xgboost algorithm exhibited the best performance, consistently delivering the lowest RMSE and MAE values. The model's ability to accurately forecast future sales volumes provides Big Mart with valuable insights for inventory management, supply chain optimization, and overall business planning.

In conclusion, the development of an effective sales prediction model using machine learning algorithms has demonstrated its potential to revolutionize the retail industry, particularly in the context of supermarket chains like Big Mart. By leveraging the power of data analysis and prediction, retailers can gain a competitive advantage, improve decision-making processes, and effectively respond to changing market dynamics.

Chapter 1 – INTRODUCTION

Cos of the rapid development of global malls and internet shopping, everyday competition between numerous shopping centres as well as large marts is growing more intense and violent. Each market tries to attract a large number of clients by offering tailored and limited-time discounts, so that the volume of sales for each item can be estimated for the organization's stock control, transportation, and logistical services. The present machine learning algorithm is very powerful, and it provides ways for predicting or forecasting sales in any type of company, which is extremely useful in overcoming low-cost prediction approaches. Better forecasting is always beneficial, both in building and upgrading marketing plans for the marketplace, which is very useful.

In addition to sales forecasting, machine learning algorithms can also be utilized in various other aspects of the retail industry. One such area is customer segmentation, where algorithms can analyze large amounts of data to identify different customer groups based on their preferences, demographics, and behavior patterns. This segmentation allows retailers to tailor their marketing strategies and product offerings to specific customer segments, thereby improving customer satisfaction and increasing sales.

Moreover, machine learning algorithms can be employed in recommendation systems to provide personalized product recommendations to customers. By analyzing a customer's past purchase history, browsing behavior, and preferences, these algorithms can suggest relevant products that the customer is likely to be interested in. This not only enhances the customer's shopping experience but also boosts sales by encouraging additional purchases.

Furthermore, machine learning can assist retailers in optimizing their pricing strategies. Algorithms can analyze market trends, competitor pricing, and customer demand to determine the optimal price points for different products. This dynamic pricing approach allows retailers to adjust prices in real-time, maximizing profitability while remaining competitive in the market.

Machine learning algorithms can also be employed in fraud detection and prevention in the retail industry. By analyzing transaction data and identifying patterns indicative of fraudulent activities.

Overall, the rapid development of machine learning algorithms has revolutionized the retail industry by providing valuable insights, improving decision-making processes, and optimizing various aspects of the retail business. As competition intensifies, retailers that embrace and leverage these advanced technologies are more likely to stay ahead in the market and thrive in the ever-evolving retail landscape.

Chapter 2 - LITERATURE REVIEWS

Literature survey 1:-

Title:- "A comparative study of linear and nonlinear models for aggregate retails sales forecasting"

Year:- 2003

Authors:- Ching-WuChua,Guoqiang PeterZhangb

Abstract:-

The purpose of this paper is to compare the accuracy of various linear and nonlinear models for forecasting aggregate retail sales. Because of the strong seasonal fluctuations observed in the retail sales, several traditional seasonal forecasting methods such as the time series approach and the regression approach with seasonal dummy variables and trigonometric functions are employed. The nonlinear versions of these methods are implemented via neural networks that are generalized nonlinear functional approximators. Issues of seasonal time series modeling such as deseasonalization are also investigated. Using multiple cross-validation samples, we find that the nonlinear models are able to outperform their linear counterparts in out-of-sample forecasting, and prior seasonal adjustment of the data can significantly improve forecasting performance of the neural network model. The overall best model is the neural network built on deseasonalized time series data. While seasonal dummy variables can be useful in developing effective regression models for predicting retail sales, the performance of dummy regression models may not be robust. Furthermore, trigonometric models are not useful in aggregate retail sales forecasting.

Literature survey 2:-

Title:- "Sustainable development and management in consumer electronics using soft computation"

Year:- 2019

Authors:- Dr. Haoxiang Wang

Abstract:-

Combination of Green supply chain management, Green product deletion decision and green cradle-to-cradle performance evaluation with Adaptive-Neuro-Fuzzy Inference System (ANFIS) to create a green system. Several factors like design process, client specification, computational intelligence and soft computing are analysed and emphasis is given on solving problems of real domain.

<u>Literature survey 3:-</u>

Title:- “Data Mining based Prediction of Demand in Indian Market for Refurbished Electronics”

Year:- 2017

Authors:- Dr. Suma V

Abstract:-

There has been an increasing demand in the e-commerce market for refurbished products across India during the last decade. Despite these demands, there has been very little research done in this domain. The real-world business environment, market factors and varying customer behavior of the online market are often ignored in the conventional statistical models evaluated by existing research work. In this paper, we do an extensive analysis of the Indian e-commerce market using data-mining approach for prediction of demand of refurbished electronics. The impact of the real-world factors on the demand and the variables are also analyzed. Real-world datasets from three random e-commerce websites are considered for analysis. Data accumulation, processing and validation is carried out by means of efficient algorithms. Based on the results of this analysis, it is evident that highly accurate prediction can be made with the proposed approach despite the impacts of varying customer behavior and market factors. The results of analysis are represented graphically and can be used for further analysis of the market and launch of new products.

Chapter 3 - SYSTEM ANALYSIS

Existing system

A significant amount of effort has been put in to date the territory of deal forecasting. This section provides a quick overview of the major work in the topic of big mart transactions. A variety of different measurable approaches, such as regression, (ARIMA) Auto-Regressive Integrated Moving Average, (ARMA) Auto-Regressive Moving Average, have been used to produce a few deal forecasting standards.

The Radial "Base Function Neural Network (RBFN)" is necessary to have an exceptional potential for anticipating agreements, and Fluffy Neural Networks were constructed with the goal of boosting prophetic effectiveness.

Drawbacks

- Less amount of accuracy score
- Small level data-set
- Applicable on small level prediction work

Proposed system

The suggested model focuses on the application of several algorithms to the dataset. We calculate Accuracy, MAE, MSE, RMSE, and finally arrive with the best yield algorithm. The Algorithms listed below are employed.

- Linear Regression, Polynomial Regression Algorithm,decision Tree, Random forest,XGBoost Regression

Advantages

- Increasing the accuracy score
- Time consumption is less
- Comparison of different algorithms can be observed
- Large amount of feature we are taking for the training and testing.

System Requirements

Hardware:

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB

Software:

- Anaconda Navigator
- in Python Language
- Jupyter Notebook

Feasibility study

Feasibility study in the sense it's a practical approach of implementing the proposed model of system . Here for a machine learning projects .we generally collect the input from online websites and filter the input data and visualize them in graphical format and then the data is divided for training and testing . That training is testing data is given to the algorithms to predict the data .

1. First, we take dataset.

2. Filter dataset according to requirements and create a new dataset which has attribute according to analysis to be done

3. Perform Pre-Processing on the dataset

4. Split the data into training and testing

5. Train the model with training data then analyze testing dataset over classification algorithm

6. Finally you will get results as accuracy metrics.

Chapter 4 – SYSTEM ARCHITECTURE

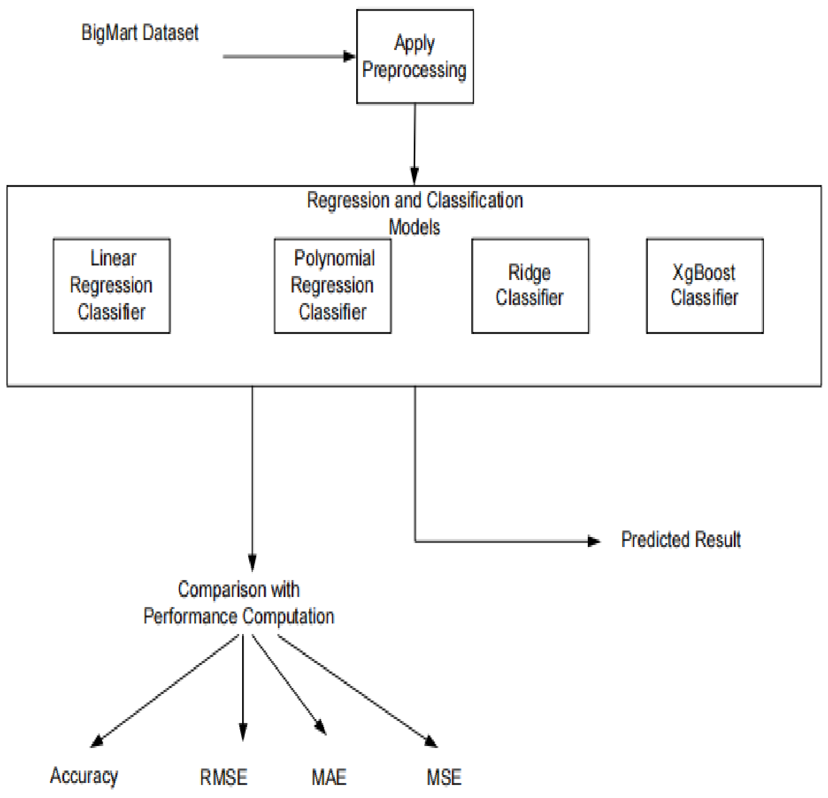

BigMart Dataset
Apply Preprocessing
Regression and Classification Models
Linear Regression Classifier
Polynomial Regression Classifier
Ridge Classifier
XgBoost Classifier
Predicted Result
Comparison with Performance Computation
Accuracy
RMSE
MAE
MSE

Fig 4.1

Modules

1. **DATA COLLECTION**
2. **DATA PRE-PROCESSING**
3. **FEATURE EXTRATION**
4. **EVALUATION MODE**

DATA COLLECTION

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for problem. In this step the data set is converted into the understandable format which can be fed into machine learning models.

Data used in this paper is a set of cervical cancer data with 15 features . This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

DATA PRE-PROCESSING

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.

Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

FEATURE EXTRATION

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

EVALUATION MODEL

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance.

Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

RMSE stands for Root Mean Square Error, and it is a commonly used evaluation metric in machine learning to measure the performance of regression models. RMSE calculates the average deviation between the predicted value and the actual values, giving more weight too larger errors.

MAE stands for Mean Absolute Error, and it is another commonly used evaluation metric in machine learning for regression models. MAE measures the average absolute difference between the predicted values and the actual values.

MSE stands for Mean Squared Error, and it is another commonly used evaluation metric in machine learning for regression models. MSE measures the average squared difference between the predicted values and the actual values.

Chapter 5 – FLOW OF OPERATIONS

UML Diagrams

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- actors
- business processes
- (logical) components
- activities
- programming language statements
- database schemas, and
- Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies· UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

Sequence Diagram:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

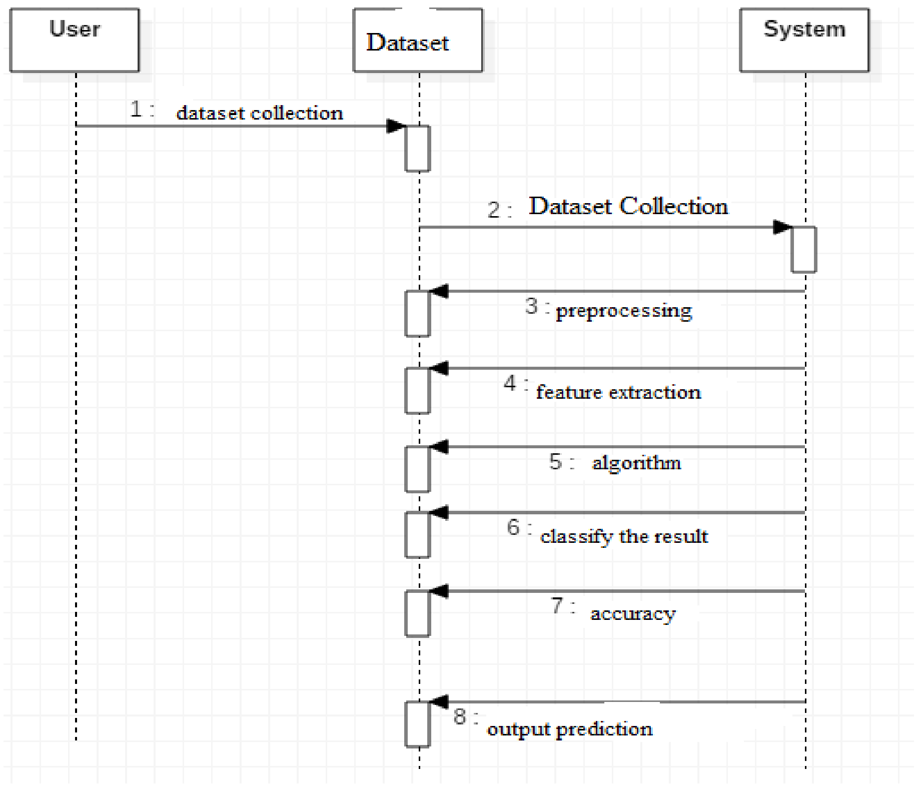


Fig 5.1

Activity Diagrams:

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity

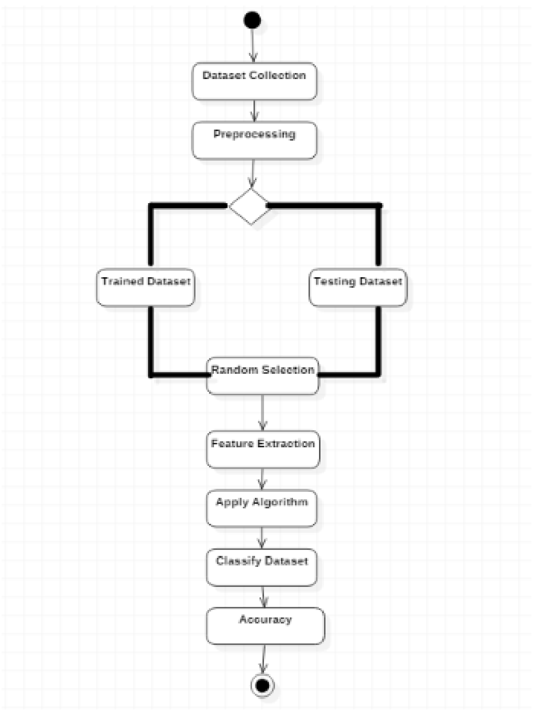


Fig 5.2

diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Usecase diagram:

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

OMG is continuously putting effort to make a truly industry standard.

UML stands for Unified Modeling Language.

UML is a pictorial language used to make software blue prints

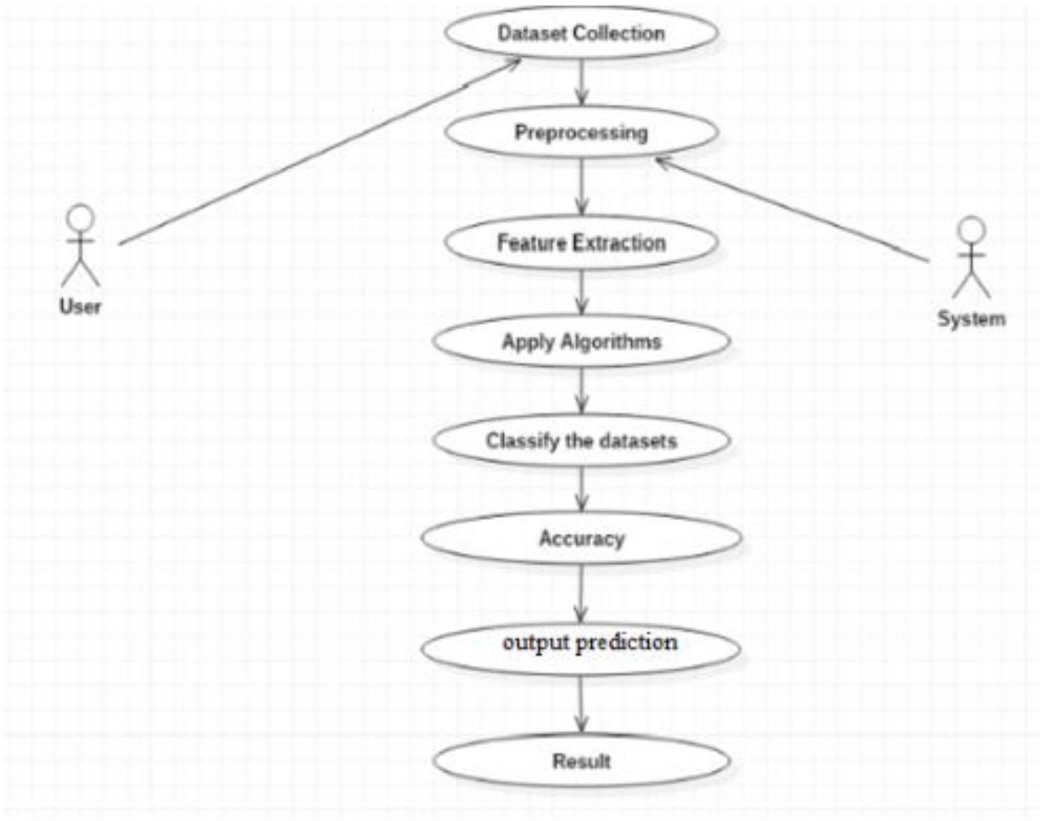


Fig 5.3

Class diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

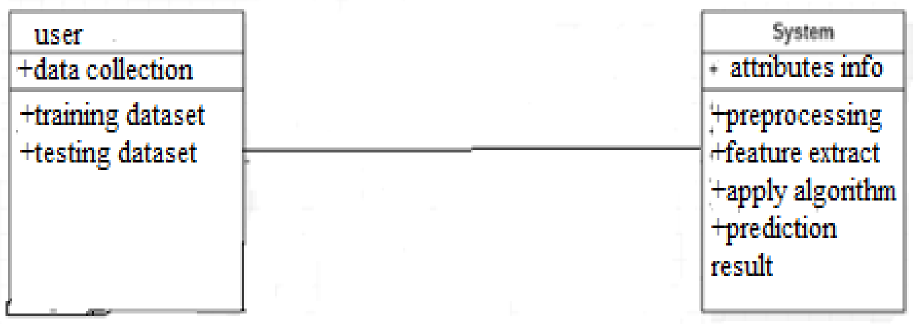


Fig 5.4

In the diagram, classes are represented with boxes that contain three compartments:

The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

Chapter 6 – SOFTWARE ENVIRONMENT

What is Python: -

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

Advantages of Python:

Let's see how Python dominates over other languages.

Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don't have to write the complete code for that manually.

Extensible

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

Improved Productivity

The language's simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

Simple and Easy

When working with Java, you may have to create a class to print 'Hello World'. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

Object-Oriented

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

Free and Open-Source

Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

<u>Advantages of Python Over Other Languages:</u>

Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category. Python is for Everyone
Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language..

<u>History of Python:</u>

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the

CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

Domain Technology – (Machine Learning)

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

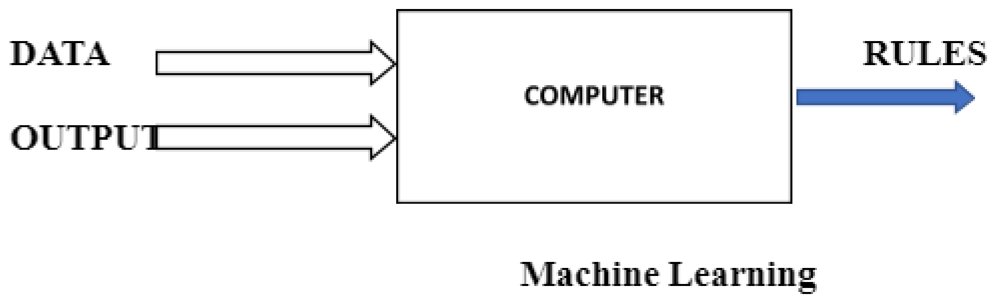


Fig 6.1

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

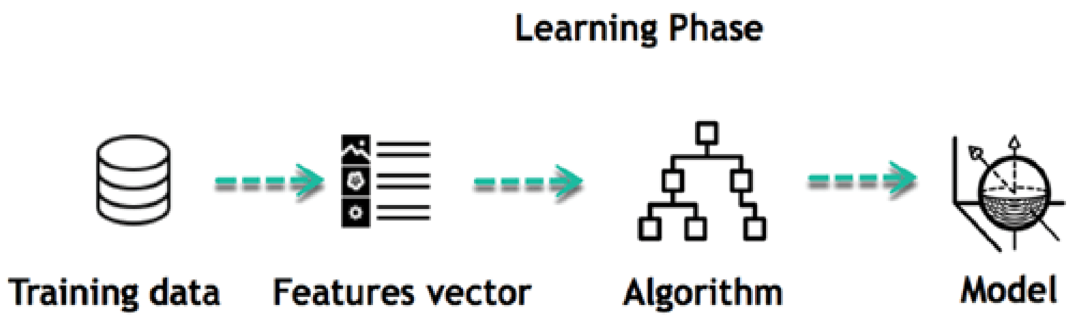


Fig 6.2

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant.

Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

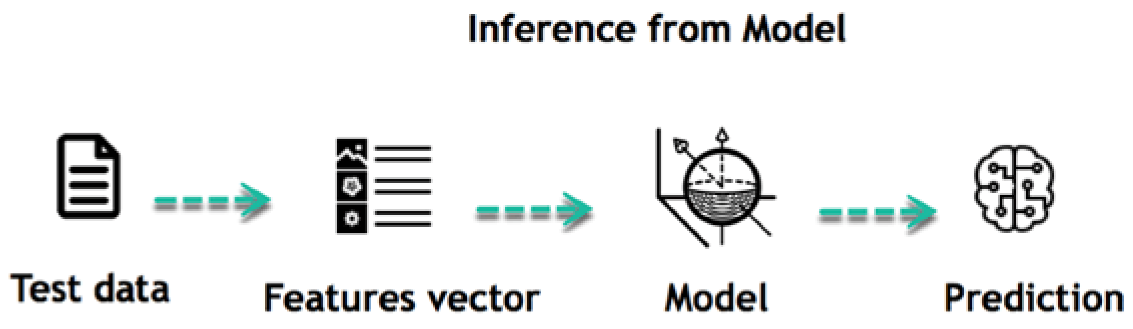


Fig 6.3

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

Algorithm Name	**Description**	**Type**
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression

Logistic regression	Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification

AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

- Classification task
- Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

<u>Unsupervised learning</u>

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

Application of Machine learning

Augmentation:

Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Automation:

Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

Finance Industry

Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

Government organization

The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

Healthcare industry

Healthcare was one of the first industry to use machine learning with image detection.

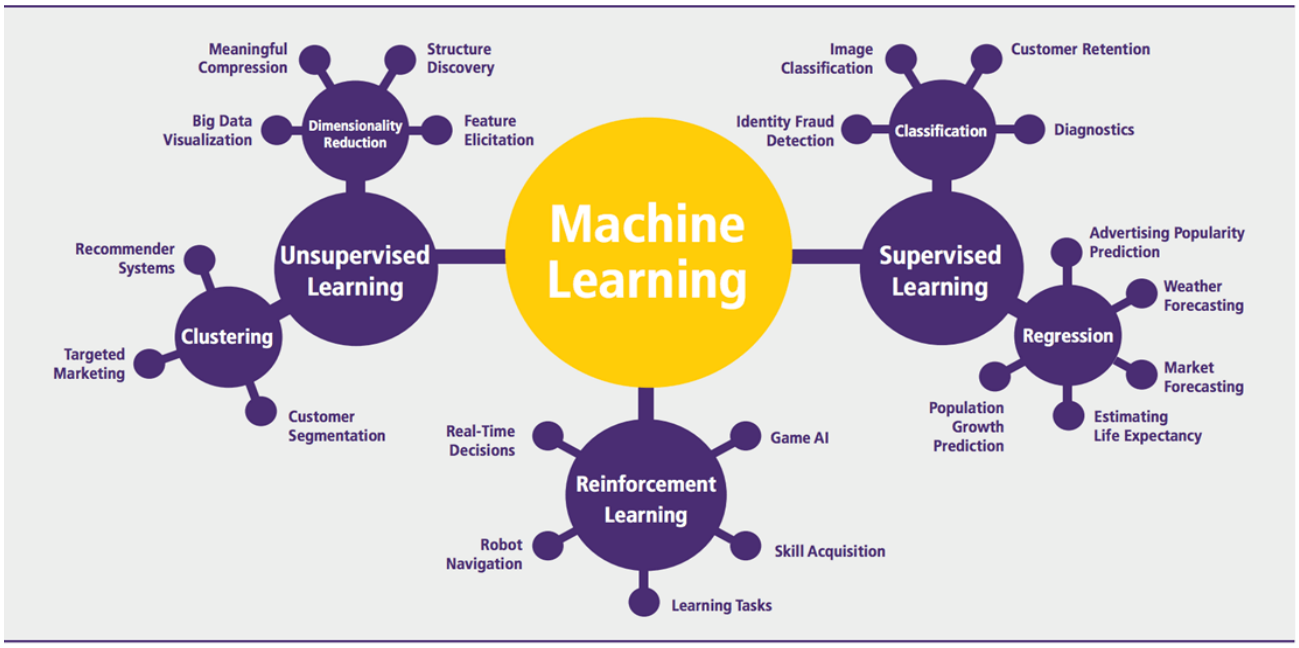


Fig 6.4

Marketing

Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

Example of Machine Learning Google Car

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

- Q-learning
- Deep Q network
- State-Action-Reward-State-Action (SARSA)
- Deep Deterministic Policy Gradient (DDPG)

Applications/ Examples of deep learning applications

AI in Finance: The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

AI in HR: Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

TensorFlow

the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations. To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.Google wants to use machine learning to take advantage of their massive datasets to give users the best experience.

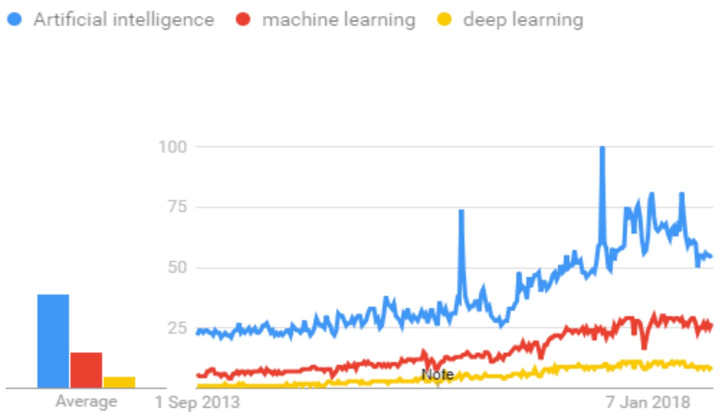


Fig 6.5

Modules Used in Project:

Modules in Python are files that contain Python code and define reusable functions, classes, and variables. They allow you to organize your code into separate files, making it easier to manage and maintain.

To use a module in Python, you need to import it into your program using the import statement. Here's an example:

```
import math

print(math.sqrt(16))   # Output: 4.0
```

Python comes with a standard library that includes many useful modules for various tasks. Some commonly used modules include **os**, **datetime**, **random**, **json**, and **csv**. These modules provide functions and classes for working with files, dates and times, generating random numbers, and handling data in different formats.

Seaborn:

Seaborn is a Python data visualization library that provides a high-level interface for creating aesthetically pleasing and informative statistical graphics. It is built on top of Matplotlib and offers a simplified approach to visualizing data.

Pandas:

Pandas is a powerful and popular Python library used for data manipulation and analysis. It provides data structures and functions for efficiently working with structured data, such as tables and time series.

Numpy:

NumPy is a fundamental Python library for numerical computing. It provides high-performance multidimensional array objects, along with a wide range of mathematical functions, to efficiently work with large datasets and perform complex mathematical operations.

Scipy:

Scipy is a powerful Python library used for scientific and technical computing. It builds on top of NumPy and provides additional functionality for tasks such as optimization, interpolation, integration, signal processing, linear algebra, and more.

Matplotlib:

Matplotlib is a Python module that provides a comprehensive library for creating static, animated, and interactive visualizations in Python. It is one of the most popular visualization libraries used by data scientists, analysts, and researchers worldwide

Matplotlib allows you to create a wide range of plots such as line, bar, scatter, histogram, pie, and more. It provides a variety of customization options, such as controlling axis limits, setting colors and styles, adding legends and labels, and more. You can also use Matplotlib to create subplots and combine multiple plots into a single figure.

Matplotlib is built on top of NumPy arrays and provides a simple interface that makes it easy to work with data in Python. It can be used in conjunction with other Python libraries such as Pandas and Seaborn to create complex visualizations and data analysis workflows.

Pickle

The **pickle** module in Python provides a way to serialize and deserialize Python objects. Serialization refers to the process of converting an object into a byte stream, which can be stored in a file or transmitted over a network. Deserialization, on the other hand, is the process of reconstructing the object from the serialized byte stream.

Chapter 7 – PYTHON INSTALLATION

Install Python Step by Step in mac and Windows:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

Note: The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your System Requirements. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a Windows 64-bit operating system. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. Download the Python Cheatsheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

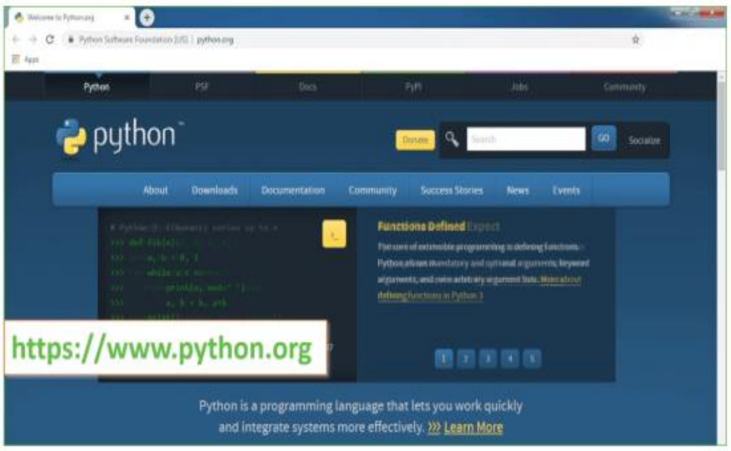
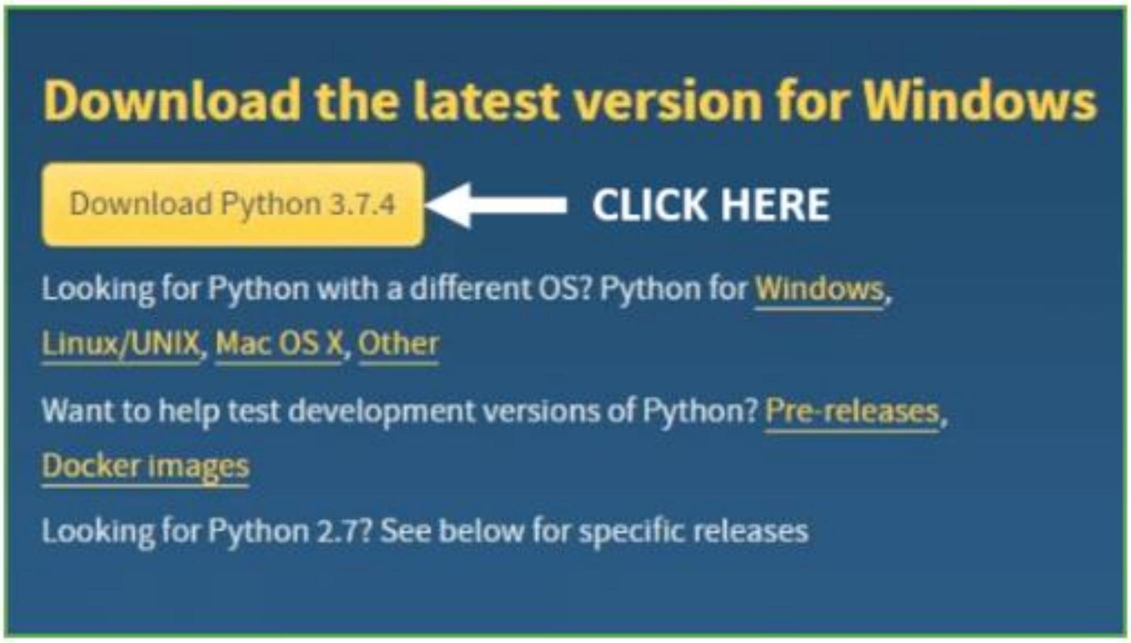

python
About
Downloads
Documentation
Community
Success Stories
News
Events
Functions Defined
https://www.python.org
Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Fig 7.1

Now, check for the latest and the correct version for your operating system.

Step 2: Click on the Download Tab.



Download the latest version for Windows

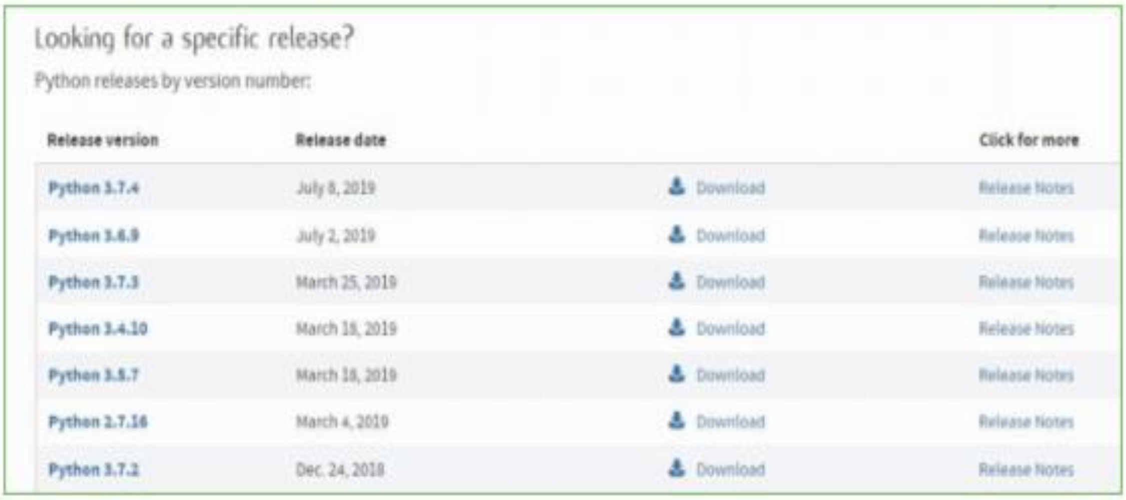
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Fig 7.2

Step 3: You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4.



Looking for a specific release?

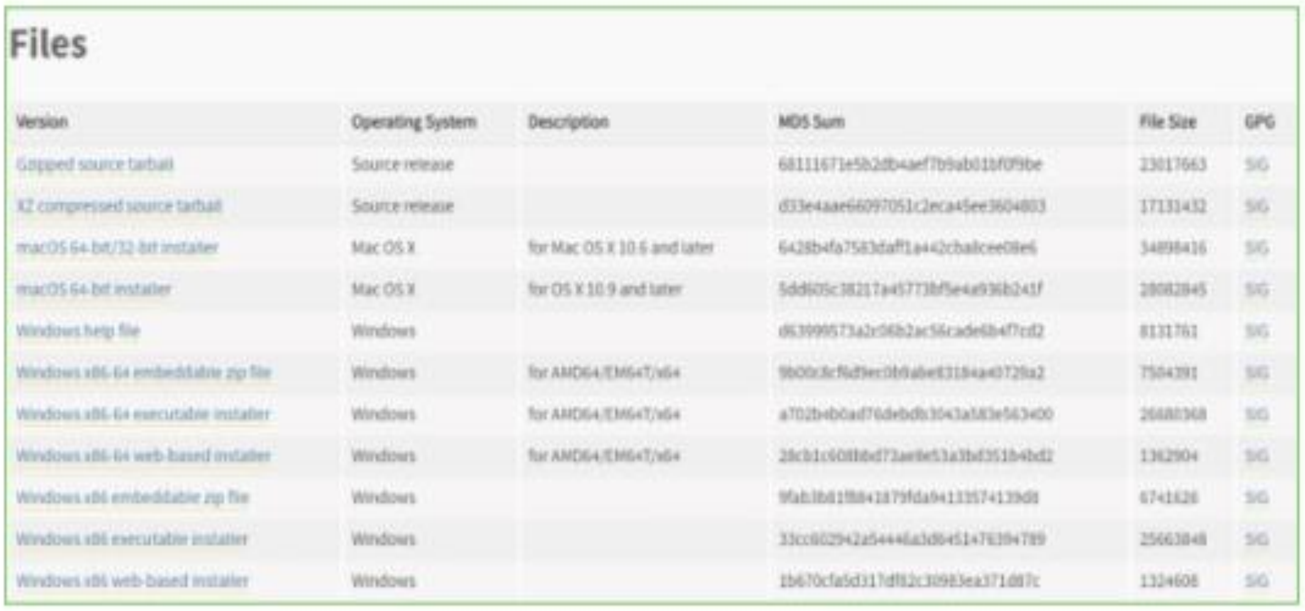
Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	Download	Release Notes
Python 3.6.9	July 2, 2019	Download	Release Notes
Python 3.7.3	March 25, 2019	Download	Release Notes
Python 3.4.10	March 18, 2019	Download	Release Notes
Python 3.5.7	March 18, 2019	Download	Release Notes
Python 2.7.16	March 4, 2019	Download	Release Notes
Python 3.7.2	Dec. 24, 2018	Download	Release Notes

Fig 7.3

Step 4: Scroll down the page until you find the Files option.

Step 5: Here you see a different version of python along with the operating system.



Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a442cba8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d63999573a2c56b2ac56cade6b4f7cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a54446a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

Fig 7.4

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

INSTALLATION OF PYTHON:

Step 1: Go to Download and Open the downloaded python version to carry out the installation process

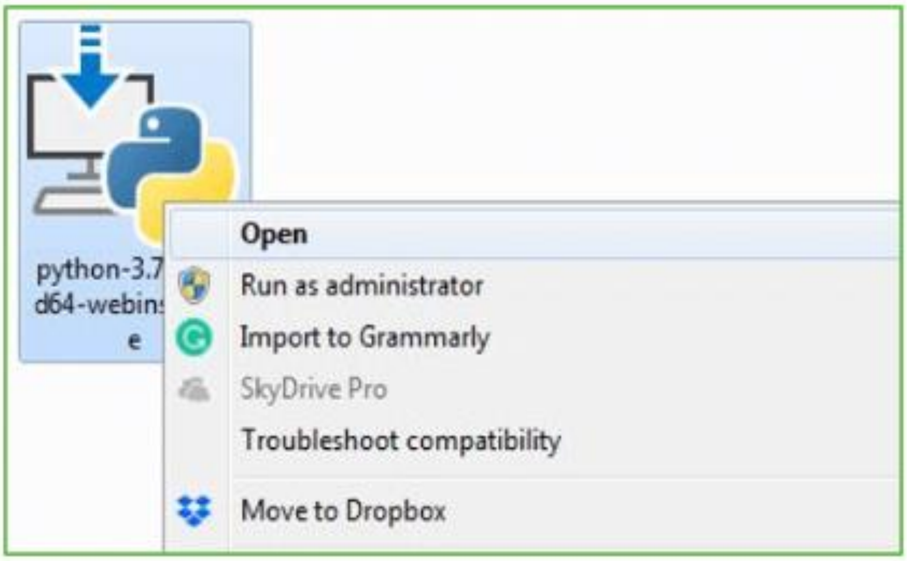
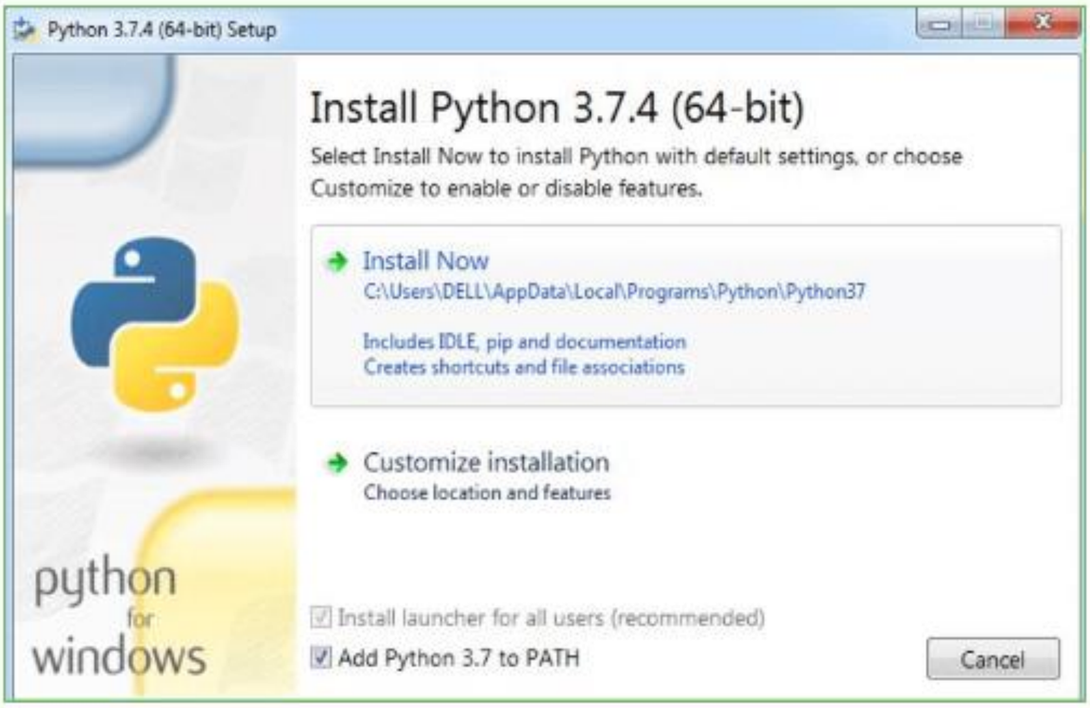


Fig 7.5

Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.


Python 3.7.4 (64-bit) Setup

Install Python 3.7.4 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\DELL\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

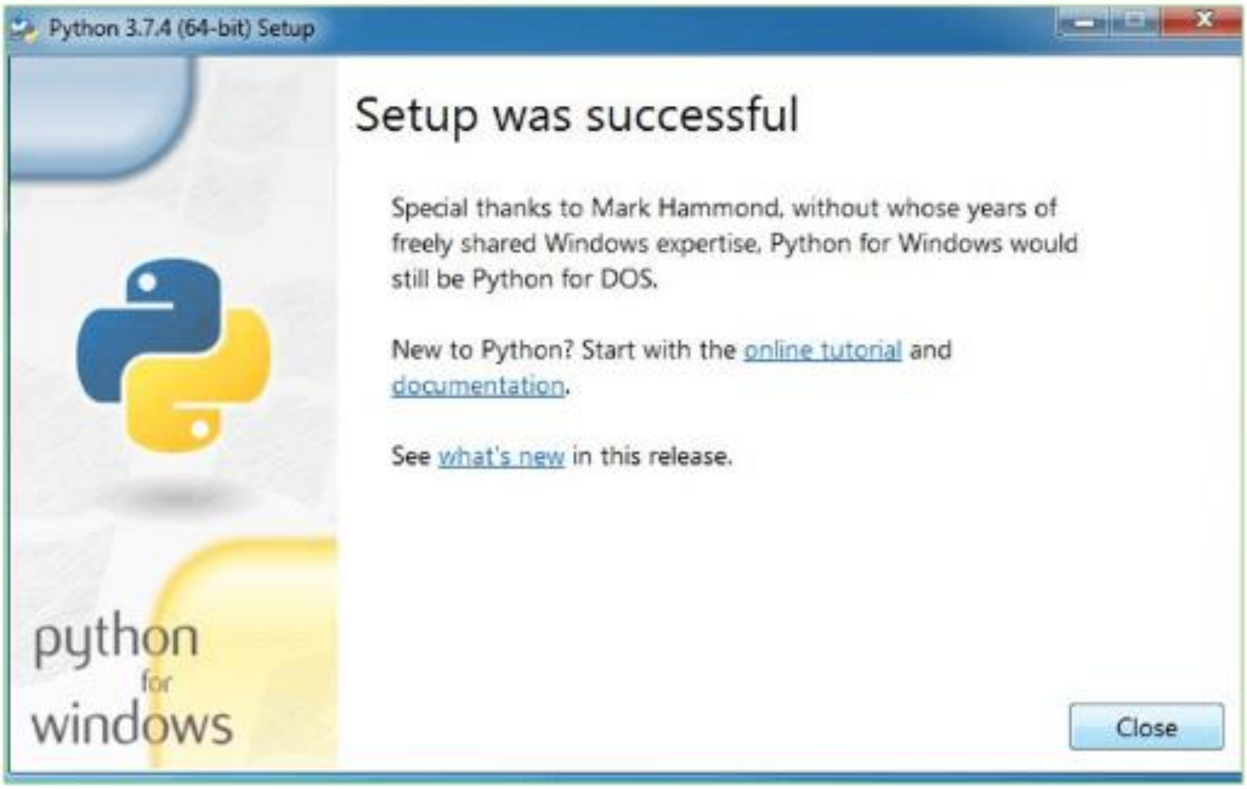
Customize installation
Choose location and features

[✓] Install launcher for all users (recommended)
[✓] Add Python 3.7 to PATH

Cancel

Fig 7.6

Step 3: Click on Install NOW After the installation is successful. Click on Close.


Python 3.7.4 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the online tutorial and documentation.

See what's new in this release.

Fig 7.7

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note: The installation process might take a couple of minutes.

Verify the python installation

Step 1: Click on Start

Step 2: In the Windows Run Command, type "cmd".

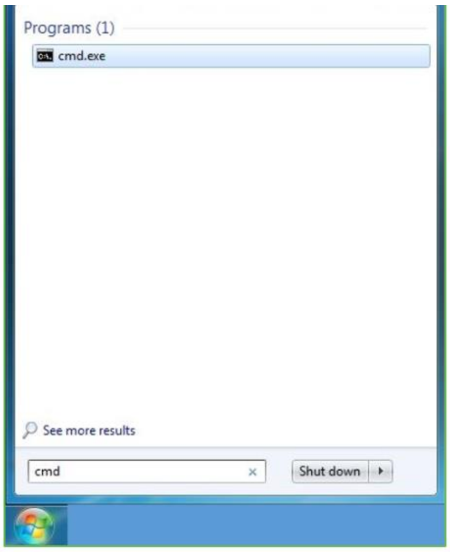
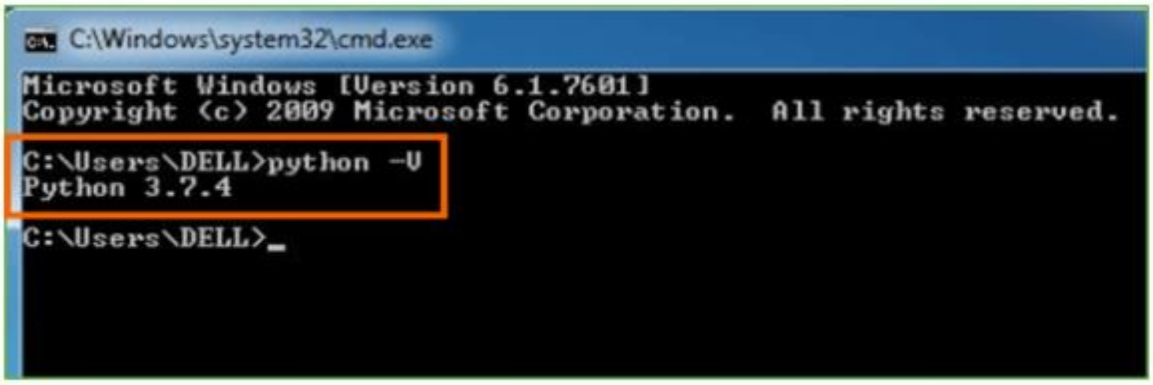


Fig 7.8

Step 3: Open the Command prompt option.

Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

Fig 7.9

Step 5: You will get the answer as 3.7.4

Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type "python idle".

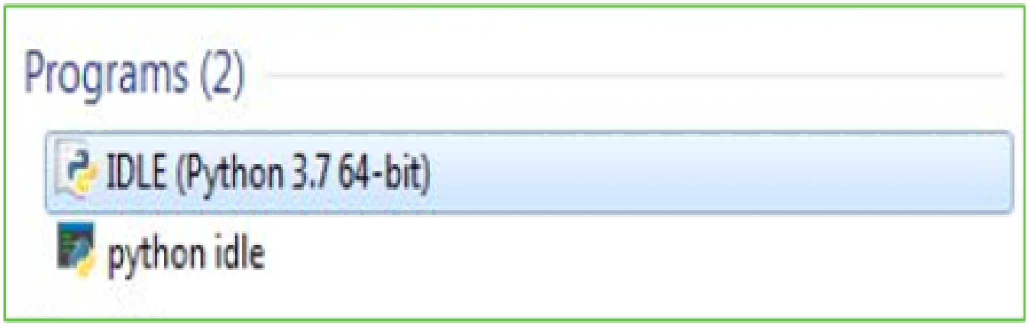
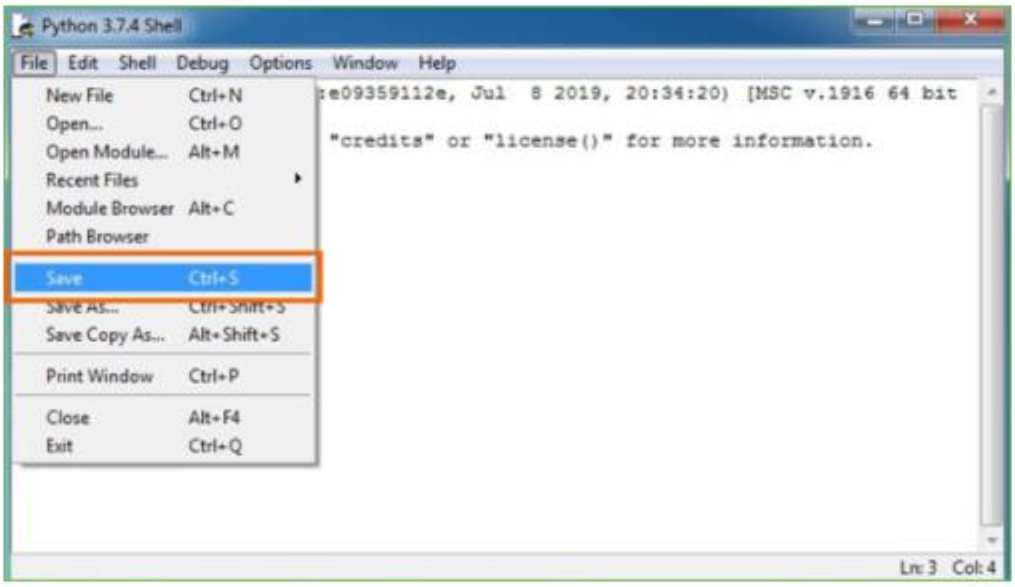


Fig 7.10

Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program

Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



```
:e09359112e, Jul  8 2019, 20:34:20) [MSC v.1916 64 bit
 "credits" or "license()" for more information.
```

Fig 7.11

Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

Step 6: Now for e.g. **enter print**

Chapter 8 – SOURCE CODE

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor,
BaggingRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score
from xgboost import XGBRegressor

<a href="#toc" role="button" aria-pressed="true" >↑Back to Table of Contents ↑</a>

<a id='3'></a>
# Exploaring the data

train = pd.read_csv('Train.csv') # read train data
train.head()

test = pd.read_csv('Test.csv') # read test data
test.head()

train.shape # in train dataset we have 8523 rows and 12 columns

test.shape # in test dataset we have 5681 rows and 11 columns

train.info()

test.info()

train.nunique()

test.nunique()

Categorical Features:

Item_Fat_Content<br>
Item_Type<br>
Outlet_Identifier<br>
Outlet_Size<br>
```

Outlet_Location_Type<br>
Outlet_Type

```
train.columns[train.isna().any()]

test.columns[test.isna().any()]

train.isnull().sum()

test.isnull().sum()

train.describe()

test.describe()
```

<a href="#toc" role="button" aria-pressed="true" >⬆️Back to Table of Contents ⬆️</a>

<a id='4'></a>
# Data Cleaning

<a id='4.1'></a>
handling Null Values

```
plt.figure(figsize=(18,10))

plt.subplot(2,2,1)
plt.title('Item_Weight in Train dataset')
sns.boxplot(x='Item_Weight', data=train)

plt.subplot(2,2,2)
plt.title('Item_Weight in Test dataset')
sns.boxplot(x='Item_Weight', data=test)

plt.show()
```

from box plot Item Weight dosen't appear to have outliers in both datasets

```
# Use the mean to replace the null values in Item_Weight feature
train.Item_Weight = train.Item_Weight.fillna(train.Item_Weight.mean())
test.Item_Weight = test.Item_Weight.fillna(test.Item_Weight.mean())

train.Outlet_Size.unique()

train.Outlet_Size.value_counts()

test.Outlet_Size.value_counts()
```

here we can replace the null values with the mode because it is a categorical column

```
train.Outlet_Size = train.Outlet_Size.fillna(train.Outlet_Size.mode()[0])
test.Outlet_Size = test.Outlet_Size.fillna(test.Outlet_Size.mode()[0])
```

```
train.isnull().sum()
```

```
test.isnull().sum()
```

<a href="#toc" role="button" aria-pressed="true" >⬆Back to Table of Contents ⬆</a>

<a id='5'></a>
# EDA

```
train.nunique()
```

```
train.Item_Fat_Content.unique()
```

```
train.Item_Fat_Content.value_counts()
```

We see there are some irregularities in the column and it is needed to fix them!<br>
for example there is low fat one with upper case and other with lower case and another abbreviated (LF) <br>
also Regular and reg <br>
so we have 2 types of item fat content Low fat and regular

```
train.Item_Fat_Content = train.Item_Fat_Content.replace(to_replace=['low fat', 'LF', 'reg'],
value=['Low Fat', 'Low Fat', 'Regular'])
```

```
train.Item_Fat_Content.value_counts()
```

```
test.Item_Fat_Content.unique()
```

```
test.Item_Fat_Content.value_counts()
```

same problem in the testing data

```
test.Item_Fat_Content = test.Item_Fat_Content.replace(to_replace=['low fat', 'LF', 'reg'],
value=['Low Fat', 'Low Fat', 'Regular'])
```

```
test.Item_Fat_Content.value_counts()
```

```
test.Item_Fat_Content.unique()
```

```
test.Item_Fat_Content.value_counts()
```

```
plt.figure(figsize=(18,12))

plt.subplot(2,2,1)
plt.title("Item_Fat_Content Count in the Training Data")
```

```
sns.countplot(x='Item_Fat_Content', data=train, palette='ocean')

plt.subplot(2,2,2)
low_fat = train.Item_Fat_Content[train.Item_Fat_Content == 'Low Fat'].count()
reg = train.Item_Fat_Content[train.Item_Fat_Content == 'Regular'].count()
plt.title("Item Content Fat Distrbution in the Training Data")
plt.pie([low_fat, reg], labels=['Low Fat', 'Regular'], explode=[0.01,0.01], autopct="%.2f%%",
colors=['lightgreen', 'yellow'], shadow=True)
plt.legend()

plt.show()
```

Low Fat has the majority in the data 64.73% but Regular 35.27%

```
plt.figure(figsize=(25,8))
plt.title("Item Type count in the Training data", {"fontsize" : 25})
order = train.groupby('Item_Type').count().sort_values(by='Item_Outlet_Sales',
ascending=False).index
sns.countplot(x='Item_Type', data=train, order=order);
```

Fruits and Vegetables are largely sold as people tend to use them on daily purpose.<br>
Snack Foods too have good sales.

```
train.Outlet_Size.value_counts()

plt.figure(figsize=(18,12))

plt.subplot(2,2,1)
plt.title("The size of the store in terms of ground area covered count")
sns.countplot(x='Outlet_Size', data=train, palette='spring')

plt.subplot(2,2,2)
medium = train.Outlet_Size[train.Outlet_Size == 'Medium'].count()
small = train.Outlet_Size[train.Outlet_Size == 'Small'].count()
high = train.Outlet_Size[train.Outlet_Size == 'High'].count()
plt.title("The size of the store in terms of ground area covered Distrbution")
plt.pie([medium, small, high], labels=['Medium', 'Small', 'High'], autopct="%.2f%%",
shadow=True, explode=[0.01,0.01,0.01])
my_circle = plt.Circle( (0,0), 0.4, color='white')
plt.gcf().gca().add_artist(my_circle)
plt.legend()

plt.show()
```

Medium Size has the majoirty in the data 61.05%

```
train.Outlet_Location_Type.unique()

train.Outlet_Location_Type.value_counts()
```

```
plt.figure(figsize=(18,12))

plt.subplot(2,2,1)
plt.title("The type of city in which the store is located")
sns.countplot(x='Outlet_Location_Type', data=train, palette='twilight')

plt.subplot(2,2,2)
tier_1 = train.Outlet_Location_Type[train.Outlet_Location_Type == 'Tier 1'].count()
tier_2 = train.Outlet_Location_Type[train.Outlet_Location_Type == 'Tier 2'].count()
tier_3 = train.Outlet_Location_Type[train.Outlet_Location_Type == 'Tier 3'].count()
plt.title("The type of city in which the store is located Distrbution")
plt.pie([tier_1, tier_2, tier_3], labels=['tier_1', 'tier_2', 'tier_3'], autopct="%.2f%%",
shadow=True, explode=[0.05,0.05,0.05])
my_circle = plt.Circle( (0,0), 0.4, color='white')
plt.gcf().gca().add_artist(my_circle)
plt.legend()

plt.show()
```

The Outlets are maximum in number in Tier 3 Cities.

```
train.Outlet_Type.unique()

train.Outlet_Type.value_counts()

plt.figure(figsize=(18,12))

plt.subplot(2,2,1)
plt.title("Whether the outlet is just a grocery store or some sort of supermarket count")
sns.countplot(x='Outlet_Type', data=train, palette='autumn')

plt.subplot(2,2,2)
sup_1 = train.Outlet_Type[train.Outlet_Type == 'Supermarket Type1'].count()
goc = train.Outlet_Type[train.Outlet_Type == 'Grocery Store'].count()
sup_3 = train.Outlet_Type[train.Outlet_Type == 'Supermarket Type3'].count()
sup_2 = train.Outlet_Type[train.Outlet_Type == 'Supermarket Type2'].count()
plt.title("Whether the outlet is just a grocery store or some sort of supermarket Distrbution")
plt.pie([sup_1, goc, sup_3, sup_2], labels=['Supermarket Type1', 'Grocery Store', 'Supermarket
Type3', 'Supermarket Type2'], autopct="%.2f%%", shadow=True, explode=[0.01,0.01,0.01,
0.01])
my_circle = plt.Circle( (0,0), 0.4, color='white')
plt.gcf().gca().add_artist(my_circle)
plt.legend(bbox_to_anchor=(1, 0))

plt.show()
```

The Outlets are more of Supermarket Type1.

```
train.Outlet_Establishment_Year.unique()

train['Age'] = 2021 - train.Outlet_Establishment_Year
test['Age'] = 2021 - test.Outlet_Establishment_Year

plt.figure(figsize=(12,6))

sns.countplot(x='Age', data=train, palette='autumn');
```

The Outlets are more of established and running from 35 years.

```
train.nunique()

FEATURES = [col for col in train.columns if col not in ['Item_Outlet_Sales',
'Outlet_Identifier']]
cat_features = [col for col in FEATURES if train[col].nunique() < 10]
cont_features = [col for col in FEATURES if train[col].nunique() >= 10]

cat_features

cont_features

ncols = 3
nrows = int(len(cat_features) / ncols + (len(FEATURES) % ncols > 0))-1

fig, axes = plt.subplots(nrows, ncols, figsize=(22, 10), facecolor='#EAEAF2')

for r in range(nrows):
    for c in range(ncols):
        col = cat_features[r*ncols+c]
        sns.barplot(y=train['Item_Outlet_Sales'], x=train[col], ax=axes[r, c], palette='autumn',
label='Train data')
        axes[r, c].set_ylabel('')
        axes[r, c].set_xlabel(col, fontsize=12, fontweight='bold')
        axes[r, c].tick_params(labelsize=10, width=0.5)
        axes[r, c].xaxis.offsetText.set_fontsize(6)
        axes[r, c].yaxis.offsetText.set_fontsize(6)
plt.show()
```

- The Item Outles sales are high for both Low Fat and Regular Item types.
- The Outlet Sales is maximum for Medium and High sized Outlets.
- The Outlets we have is Medium and may be with High size Outlets can improve the Outlet Sales.
- The Outlet Sales tend to be high for Tier3 and Tier 2 location types but we have only Tier3 locations maximum Outlets.
- Supermarket Type3 sales tends to be high which in Grocery store is very low

- It is quiet evident that Outlets established 36 years before is having good Sales margin.<br>
- We also have a outlet which was established before 23 years has the lowest sales margin, so established years wouldn't improve the Sales unless the products are sold according to customer's interest.

```
plt.figure(figsize=(25,8))

order=train.groupby('Item_Type').mean().sort_values(by='Item_Outlet_Sales',
ascending=False).index
sns.barplot(x='Item_Type',y='Item_Outlet_Sales',data=train,palette='spring', order=order);
```

The products available were Fruits-Veggies and Snack Foods but the sales of Seafood and Starchy Foods seems higher and hence the sales can be improved with having stock of products that are most bought by customers

```
train.nunique()
```

Continous Features:

- Item_Weight
- Item_Visibility
- Item_MRP

```
FEATURES = [col for col in train.columns if col not in ['Item_Outlet_Sales',
'Outlet_Identifier', 'Item_Identifier']]
cat_features = [col for col in FEATURES if train[col].nunique() < 25]
cont_features = [col for col in FEATURES if train[col].nunique() >= 25]

cont_features

# distrbution of the continous features
plt.figure(figsize=(10,20))
plt.subplot(4,1,1)
plt.title("Item_Weight Distrbution")
sns.kdeplot(x='Item_Weight', data=train)

plt.subplot(4,1,2)
plt.title("Item_Visibility Distrbution")
sns.kdeplot(x='Item_Visibility', data=train);

plt.subplot(4,1,3)
plt.title("Item_MRP Distrbution")
sns.kdeplot(x='Item_MRP', data=train);

plt.subplot(4,1,4)
plt.title("Item_Outlet_Sales")
sns.kdeplot(x='Item_Visibility', data=train);

plt.figure(figsize=(18,15))
plt.subplot(3,1,1)
sns.lineplot(x='Item_Weight', y='Item_Outlet_Sales', data=train);

plt.subplot(3,1,2)
sns.lineplot(x='Item_Visibility', y='Item_Outlet_Sales', data=train);

plt.subplot(3,1,3)
sns.lineplot(x='Item_MRP', y='Item_Outlet_Sales', data=train);
```

Item_Visibility has a minimum value of zero. This makes no practical sense because when a product is being sold in a store, the visibility cannot be 0.
Items MRP ranging from 200-250 dollars is having high Sales.

```
plt.figure(figsize=(25,5))
sns.barplot('Item_Type','Item_Outlet_Sales',hue='Item_Fat_Content',data=train,palette='mako')
plt.legend();
```

Seafood Regular is highest in price

```
plt.figure(figsize=(10,5))
sns.barplot('Outlet_Location_Type','Item_Outlet_Sales',hue='Outlet_Type',data=train,palette='magma')
plt.legend()
plt.show()

train.nunique()

plt.figure(figsize=(10,5))
sns.barplot('Item_Fat_Content','Item_Outlet_Sales',hue='Outlet_Size',data=train,palette='autumn')
plt.legend()
plt.show()

plt.figure(figsize=(10,5))
sns.barplot('Item_Fat_Content','Item_Outlet_Sales',hue='Outlet_Type',data=train,palette='summer')
plt.legend()
plt.show()
```

The Tier-3 location type has all types of Outlet type and has high sales margin.

<a href="#toc" role="button" aria-pressed="true" >⬆Back to Table of Contents ⬆</a>

<a id='6'></a>
# Date PreProcessing

```
train.nunique()
train.dtypes

test.dtypes

train.columns[train.isna().any()]

<a id='6.1'></a>
***Label Encoder***

#apply Label Encoder to convert Categorical ordered features to numeric
encode = LabelEncoder()
list_transform = ['Item_Fat_Content','Outlet_Location_Type','Outlet_Size','Outlet_Type']

for i in list_transform:
    train[i] = encode.fit_transform(train[i])

for i in list_transform:
    test[i] = encode.fit_transform(test[i])

train.dtypes

test.dtypes

#drop non needed features
train = train.drop(['Item_Identifier','Outlet_Identifier','Outlet_Establishment_Year'],axis=1)
test= test.drop(['Item_Identifier','Outlet_Identifier','Outlet_Establishment_Year'],axis=1)

train.corr()

plt.figure(figsize=(20,10))
sns.heatmap(train.corr(), annot=True);

<a id='6.2'></a>
***one hot encoder***

#apply one hot encoder to convert non-order categorical featuers
dummies1 = pd.get_dummies(train.Item_Type)
dummies2 = pd.get_dummies(test.Item_Type)

train.head()

test.head()

train = pd.concat([train, dummies1], axis=1)
test = pd.concat([test, dummies2], axis=1)

train.head()
```

```
test.head()

#drop the main feature and one dummy feature
train.drop(['Item_Type', 'Starchy Foods'], axis=1, inplace=True)
test.drop(['Item_Type', 'Starchy Foods'], axis=1, inplace=True)

cont_features

train.head()

test.head()

X = train[cont_features]
X.head()

test2 = test[cont_features]
test2.head()

<a href="#toc" role="button" aria-pressed="true" >⬆Back to Table of Contents ⬆</a>

<a id='7'></a>
# Feature Scaling

scaler = StandardScaler()
X = scaler.fit_transform(X)
test2 = scaler.fit_transform(test2)

X = pd.DataFrame(X, columns=cont_features)
test2 = pd.DataFrame(test2, columns=cont_features)

X.head()

test2.head()

FEATURES = [col for col in train.columns if col not in ['Item_Outlet_Sales']]
FEATURES

cat_features = [col for col in FEATURES if train[col].nunique() < 25]
cat_features

<a href="#toc" role="button" aria-pressed="true" >⬆Back to Table of Contents ⬆</a>

<a id= '8'></a>
# Assigning Feature and Target Variables

X = pd.concat([X, train[cat_features]], axis=1)
X.head()

y = train.Item_Outlet_Sales
```

```
y.head()

test2 = pd.concat([test2, test[cat_features]], axis=1)
test2.head()

train.shape

X.shape

test2.shape

<a href="#toc" role="button" aria-pressed="true" >⬆Back to Table of Contents ⬆</a>

<a id='9'></a>
# Spliting the data into Training and Testing data

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.25, random_state = 0)

<a href="#toc" role="button" aria-pressed="true" >⬆Back to Table of Contents ⬆</a>

<a id='10'></a>
# Modeling

def kfolds(model, model_name):
   model = cross_val_score(model, X,y, cv=10)
   model_score = np.average(model)
   print(f"{model_name} score on cross validation: {model_score * 100}%")

def train(model, model_name):
   model.fit(X_train, y_train)
   model_train_score = model.score(X_train, y_train)
   model_test_score = model.score(X_test, y_test)
   print(f"{model_name} model score on Training data: {model_train_score *
100}%\n{model_name} model score on Testing data: {model_test_score * 100}%")

def r2(model, model_name):
   score = r2_score(y_test, model.predict(X_test))
   print(f"R2 Score for {model_name} is {score * 100}%")

<a id='10.1'></a>
## Random Forest Regressor Model

rf_model = RandomForestRegressor()
kfolds(rf_model, "Random Forest")
train(rf_model, "Random Forest")

r2(rf_model, "Random Forest")

<a id='10.2'></a>
```

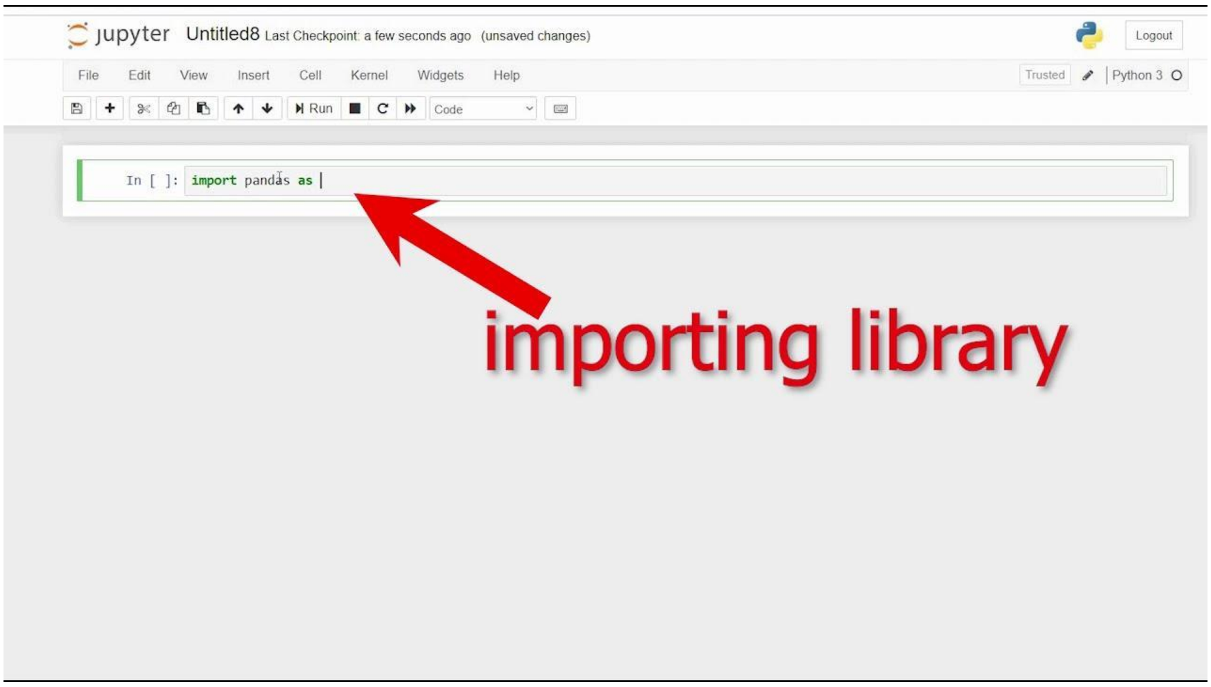
```
## Gradient Boosting Regressor Model

gbr = GradientBoostingRegressor()
kfolds(gbr, "Gradient Boosting")
train(gbr, "Gradient Boosting")

r2(gbr, "Gradient Boosting")

<a id='10.3'></a>
## Bagging Regressor Model

br = BaggingRegressor()
kfolds(br, "Bagging")
train(br, "Bagging")

r2(br, "Bagging")

<a id='10.4'></a>
## Linear Regression model

lr = LinearRegression()
kfolds(lr, "Linear Regression")
train(lr, "Linear Regression")

r2(lr, "Linear Regression")

<a id='10.5'></a>
## SVR Model

svr = SVR(gamma='auto', kernel='poly', C=15)
kfolds(svr, "SVR")
train(svr, "SVR")

r2(svr, "SVM")

<a id='10.6'></a>
## Decision Tree Regressor Model

dtr = DecisionTreeRegressor()
kfolds(dtr, "Decision Tree")
train(dtr, "Decision Tree")

r2(dtr, "Decision Tree")

<a id="10.7"></a>
## XGBoost Regressor

xgboost = XGBRegressor()
kfolds(xgboost, "XGBoost")
```

```
train(xgboost, "XGBoost")

<a href="#toc" role="button" aria-pressed="true" >↑Back to Table of Contents ↑</a>

<a id='10'></a>
# Sumbission File

**Gradient Boosting is the best**

gbr.fit(X,y)

gbr.score(X,y)

y_pred = gbr.predict(test2)

sumbission = pd.read_csv('Test.csv')

sumbission.head()

sumbission['Item_Sales_Outlet'] = y_pred

sumbission.head()

sumbission.shape

sumbission.to_csv("sumbission.csv")

Sumbission
```

These R programming is done in Jupyter Notebook. For that we have to download Anaconda Navigator. With the help of navigator we open the jupyter notebook.

Chapter 9 – RESULT

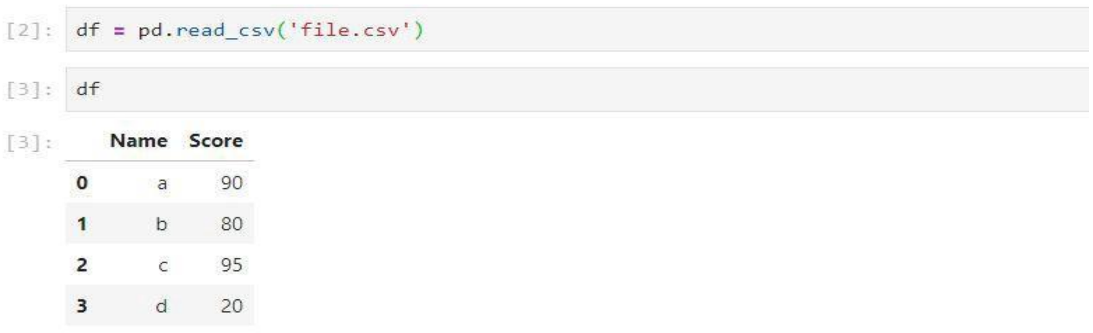
i. Importing packages


jupyter Untitled8 Last Checkpoint: a few seconds ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
importing library

```
In [ ]: import pandas as |
```

Fig 9.1

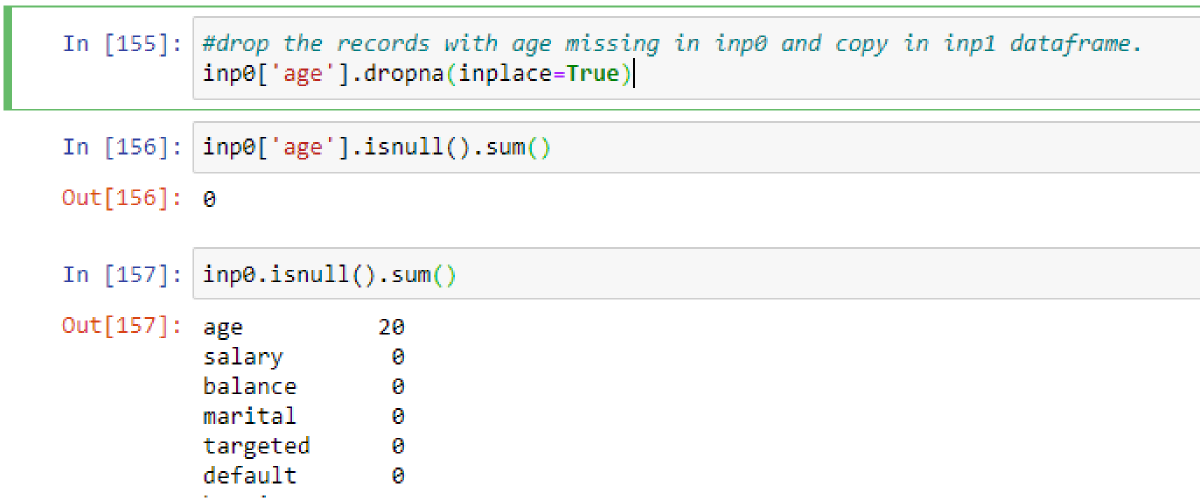
ii. Data Collection



```
[2]: df = pd.read_csv('file.csv')

[3]: df

[3]:    Name  Score
     0     a     90
     1     b     80
     2     c     95
     3     d     20
```

Fig 9.2

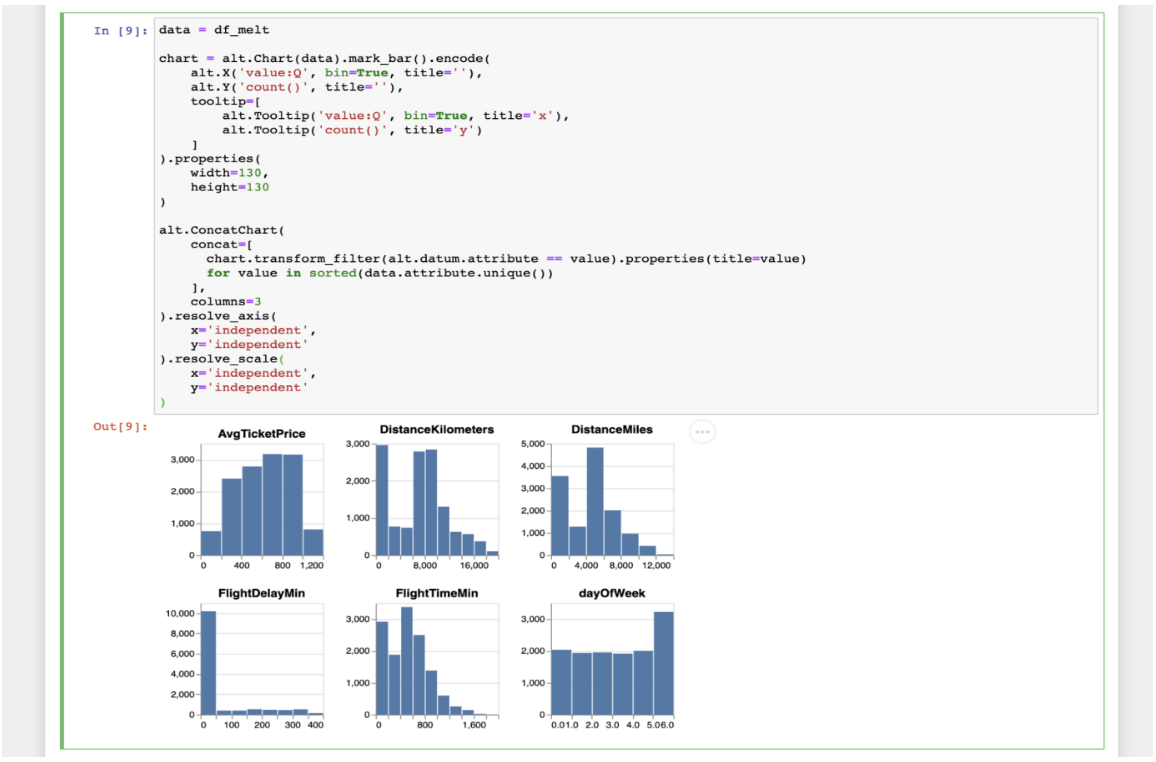
iii. Data Preproccessing



```
In [155]: #drop the records with age missing in inp0 and copy in inp1 dataframe.
          inp0['age'].dropna(inplace=True)

In [156]: inp0['age'].isnull().sum()

Out[156]: 0

In [157]: inp0.isnull().sum()

Out[157]: age          20
          salary        0
          balance       0
          marital       0
          targeted      0
          default       0
```

Fig 9.3

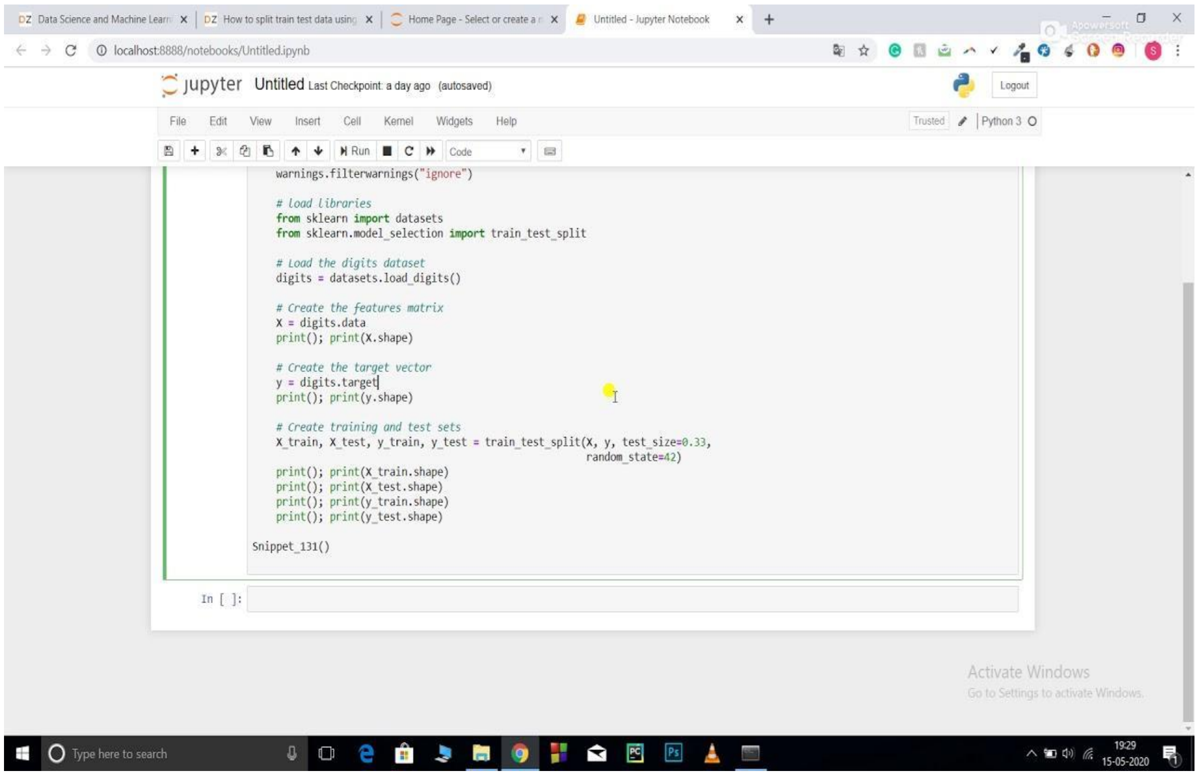
iv. Feature Extraction



```
In [9]: data = df_melt

        chart = alt.Chart(data).mark_bar().encode(
            alt.X('value:Q', bin=True, title=''),
            alt.Y('count()', title=''),
            tooltip=[
                alt.Tooltip('value:Q', bin=True, title='x'),
                alt.Tooltip('count()', title='y')
            ]
        ).properties(
            width=130,
            height=130
        )

        alt.ConcatChart(
            concat=[
              chart.transform_filter(alt.datum.attribute == value).properties(title=value)
              for value in sorted(data.attribute.unique())
            ],
            columns=3
        ).resolve_axis(
            x='independent',
            y='independent'
        ).resolve_scale(
            x='independent',
            y='independent'
        )

Out[9]:
```

Fig 9.4

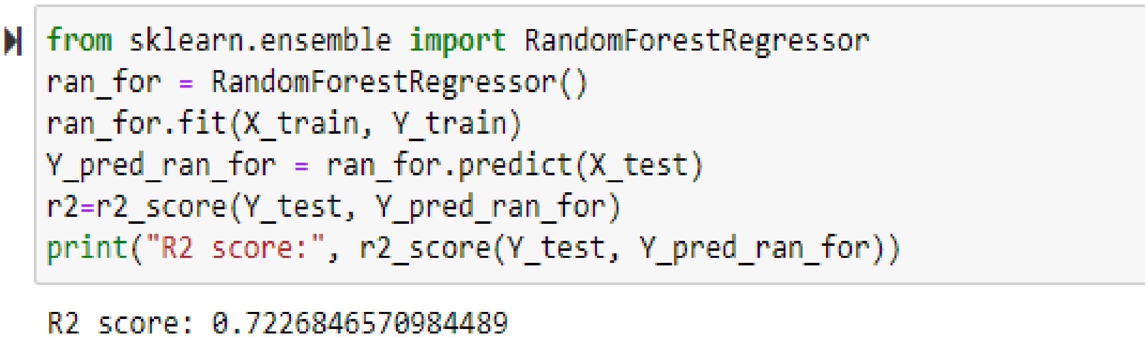
v. Training and Testing


jupyter Untitled Last Checkpoint: a day ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
warnings.filterwarnings("ignore")

# Load libraries
from sklearn import datasets
from sklearn.model_selection import train_test_split

# Load the digits dataset
digits = datasets.load_digits()

# Create the features matrix
X = digits.data
print(); print(X.shape)

# Create the target vector
y = digits.target
print(); print(y.shape)

# Create training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
                                                    random_state=42)

print(); print(X_train.shape)
print(); print(X_test.shape)
print(); print(y_train.shape)
print(); print(y_test.shape)

Snippet_131()
```

Fig 9.5

vi. Evaluation Model



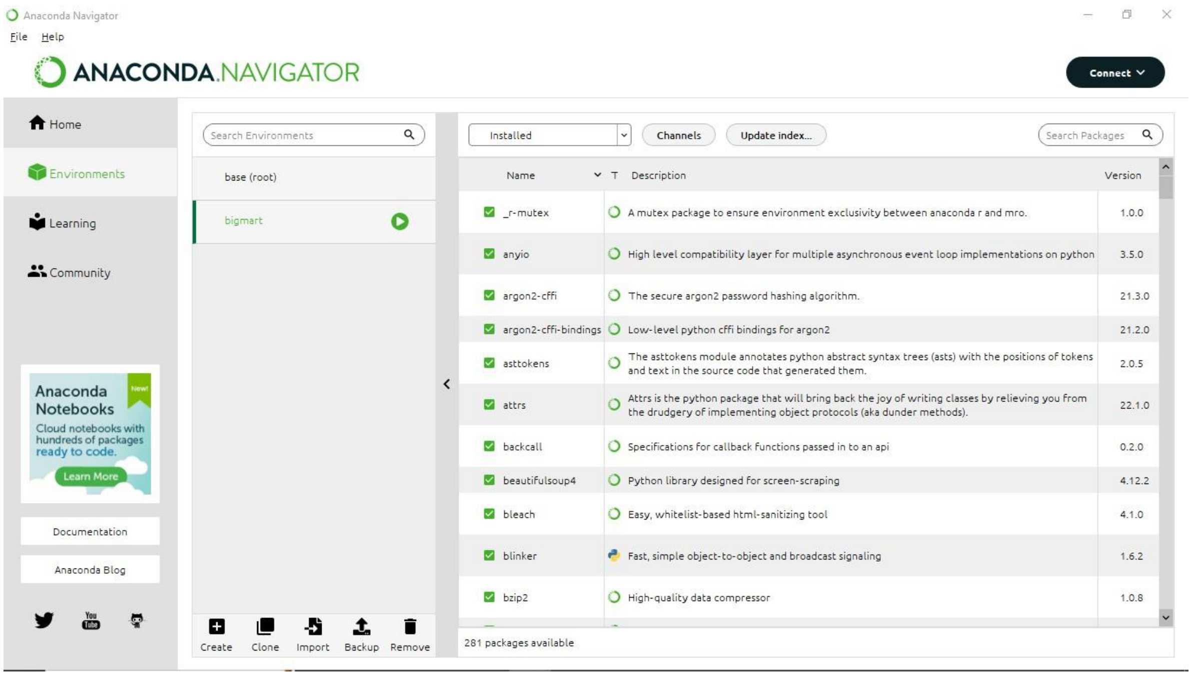
```
from sklearn.ensemble import RandomForestRegressor
ran_for = RandomForestRegressor()
ran_for.fit(X_train, Y_train)
Y_pred_ran_for = ran_for.predict(X_test)
r2=r2_score(Y_test, Y_pred_ran_for)
print("R2 score:", r2_score(Y_test, Y_pred_ran_for))
```

```
R2 score: 0.7226846570984489
```

Fig 9.6

Chapter 10 – EXECUTION

STEP 1: In Anaconda navigator we need to create root server in environment to execute python file in server.

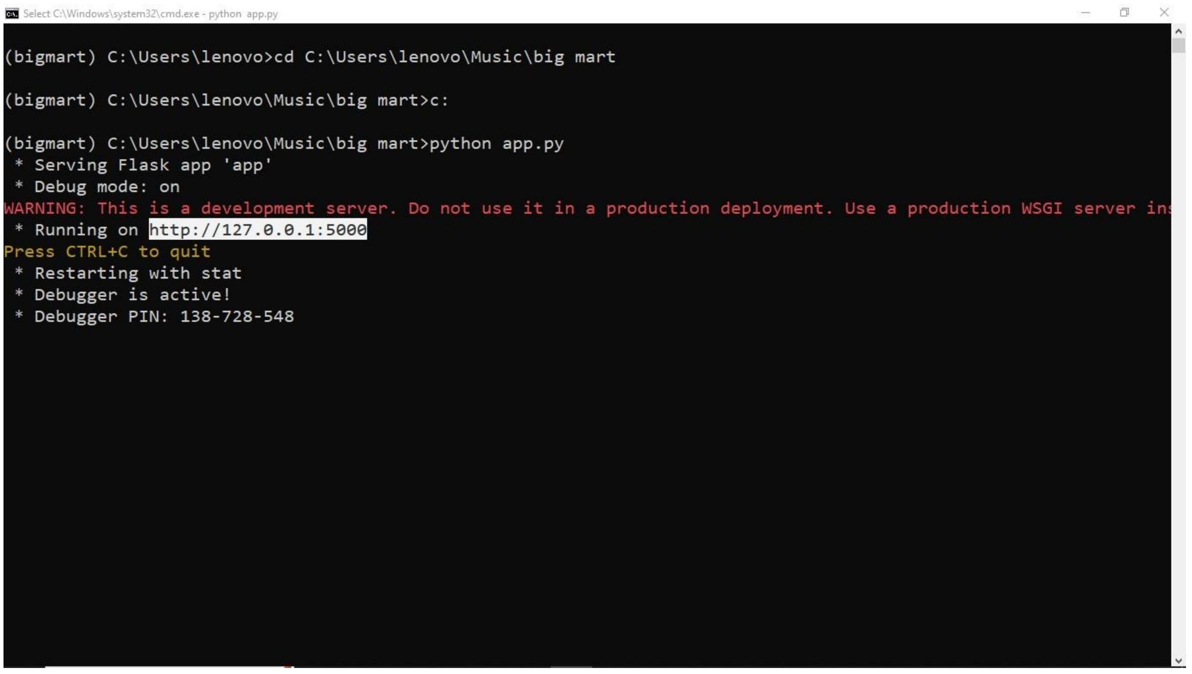


Name	T	Description	Version
✓ _r-mutex		A mutex package to ensure environment exclusivity between anaconda r and mro.	1.0.0
✓ anyio		High level compatibility layer for multiple asynchronous event loop implementations on python	3.5.0
✓ argon2-cffi		The secure argon2 password hashing algorithm.	21.3.0
✓ argon2-cffi-bindings		Low-level python cffi bindings for argon2	21.2.0
✓ asttokens		The asttokens module annotates python abstract syntax trees (asts) with the positions of tokens and text in the source code that generated them.	2.0.5
✓ attrs		Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	22.1.0
✓ backcall		Specifications for callback functions passed in to an api	0.2.0
✓ beautifulsoup4		Python library designed for screen-scraping	4.12.2
✓ bleach		Easy, whitelist-based html-sanitizing tool	4.1.0
✓ blinker		Fast, simple object-to-object and broadcast signaling	1.6.2
✓ bzip2		High-quality data compressor	1.0.8

281 packages available

Fig 10.1

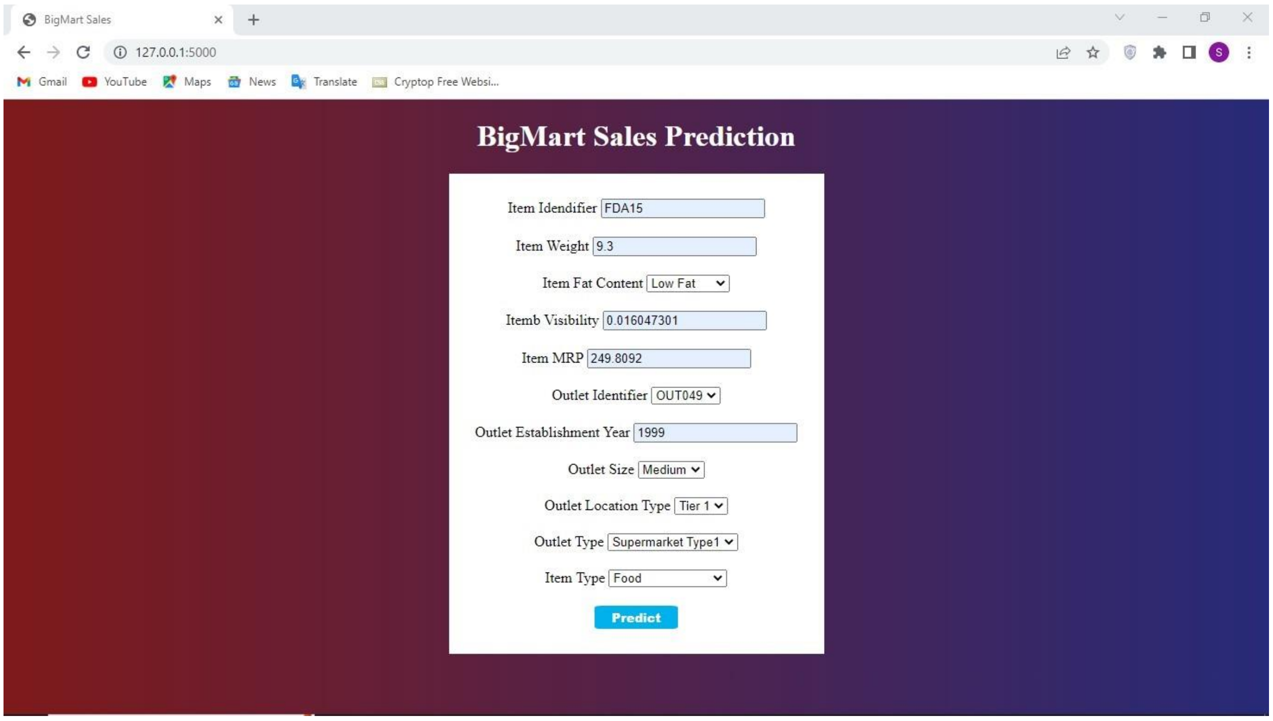
STEP 2: Open terminal from server file and open destination file in it. And execute **app.py file** and open the given IP Address in any browser.



```
(bigmart) C:\Users\lenovo>cd C:\Users\lenovo\Music\big mart

(bigmart) C:\Users\lenovo\Music\big mart>c:

(bigmart) C:\Users\lenovo\Music\big mart>python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server ins
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 138-728-548
```

Fig 10.2

STEP 3: Enter the details of the item as it is shown below.

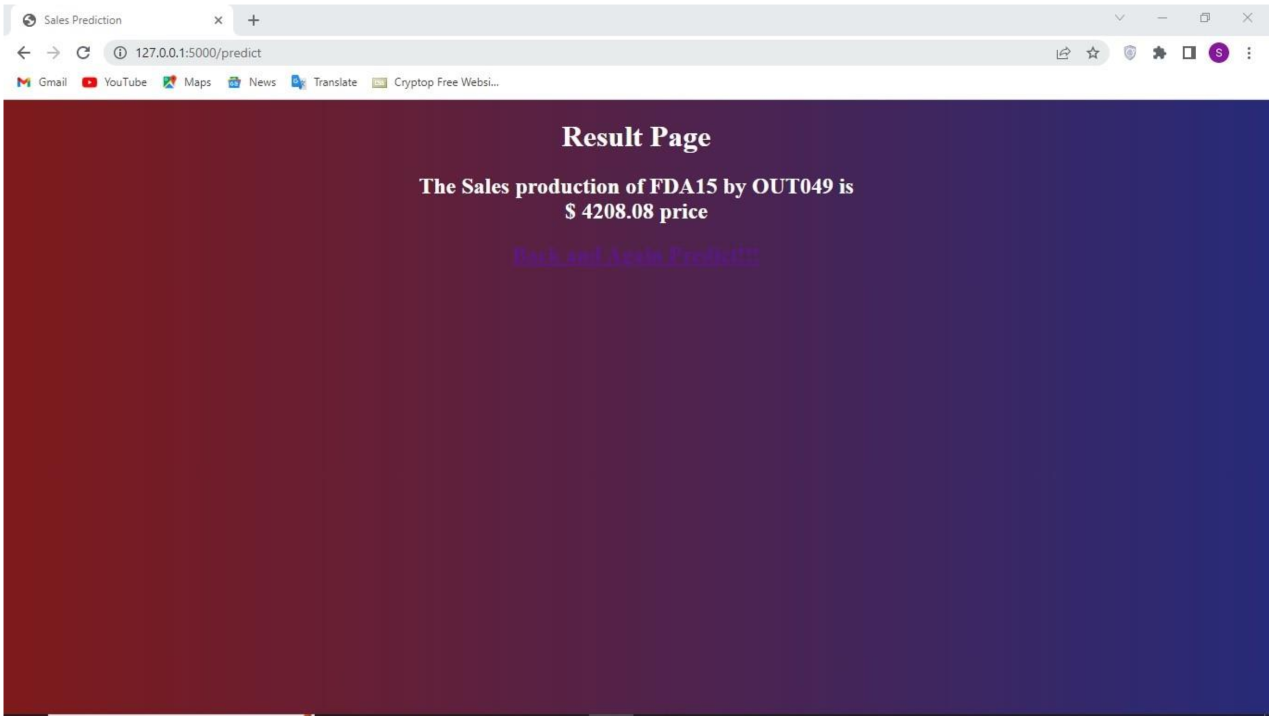

BigMart Sales
127.0.0.1:5000

BigMart Sales Prediction

Item Idendifier	FDA15
Item Weight	9.3
Item Fat Content	Low Fat
Itemb Visibility	0.016047301
Item MRP	249.8092
Outlet Identifier	OUT049
Outlet Establishment Year	1999
Outlet Size	Medium
Outlet Location Type	Tier 1
Outlet Type	Supermarket Type1
Item Type	Food

Fig 10.3

STEP 4: After filling all the details click **"Predict"** and we get the predicted price.


Sales Prediction
127.0.0.1:5000/predict

Result Page

The Sales production of FDA15 by OUT049 is
$ 4208.08 price

Back and Again Predict!!!

Fig 10.4

Chapter 11 – TESTING

Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

- Meets the business and technical requirements that guided its design and Development.
- Works as expected and can be implemented with the same characteristics.

TESTING METHODS

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Functions: Identified functions must be exercised.
- Output: Identified classes of software outputs must be exercised.
- Systems/Procedures: system should work properly

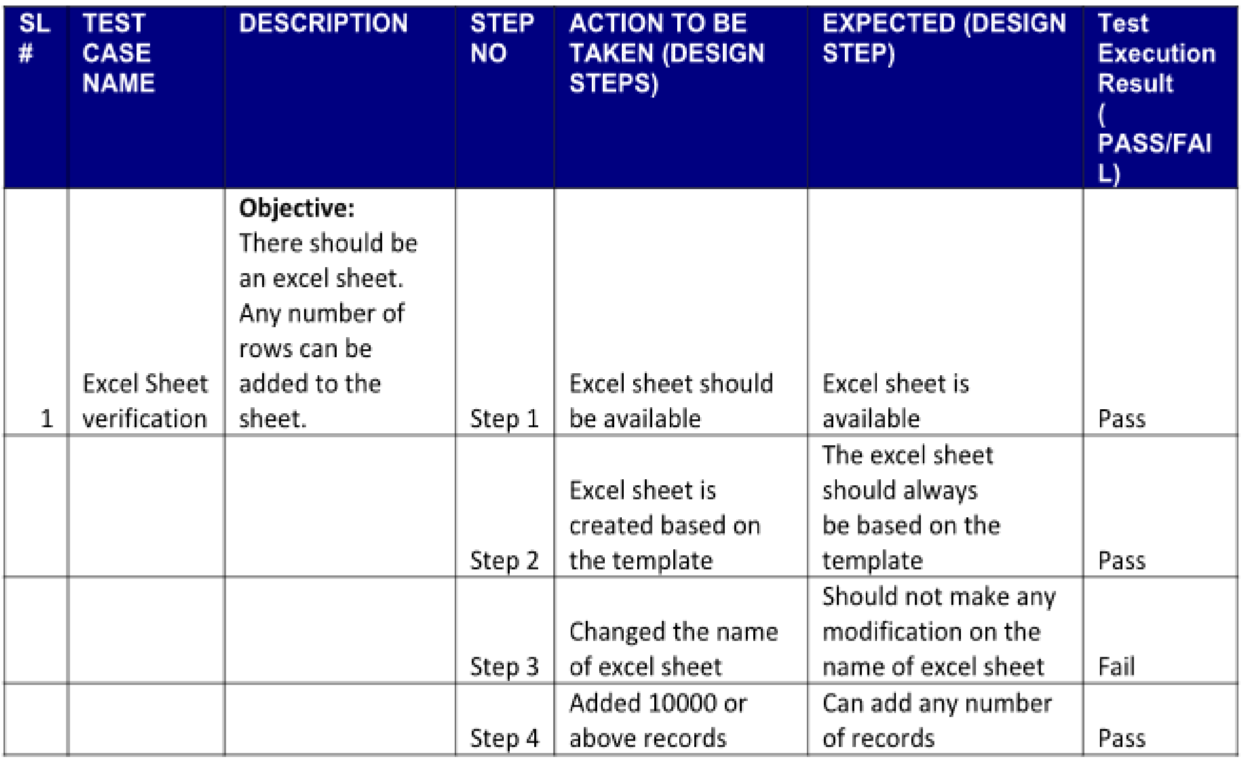
Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

Test Case for Excel Sheet Verification:

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset .

Test Case 1 :



SL #	TEST CASE NAME	DESCRIPTION	STEP NO	ACTION TO BE TAKEN (DESIGN STEPS)	EXPECTED (DESIGN STEP)	Test Execution Result (PASS/FAIL)
1	Excel Sheet verification	**Objective:** There should be an excel sheet. Any number of rows can be added to the sheet.	Step 1	Excel sheet should be available	Excel sheet is available	Pass
			Step 2	Excel sheet is created based on the template	The excel sheet should always be based on the template	Pass
			Step 3	Changed the name of excel sheet	Should not make any modification on the name of excel sheet	Fail
			Step 4	Added 10000 or above records	Can add any number of records	Pass

OUTCOME

For retailers like Big Mart, the resulting data can be used to forecast future sales volume using various machine learning techniques like big mart.

A predictive model was developed using Xgboost, Linear regression, Polynomial regression, and Ridge regression techniques for forecasting the sales of a business such as Big -Mart, and it was discovered that the model outperforms existing models.

Chapter 12 – CONCLUSION

In this work, the effectiveness of various algorithms on the data on revenue and review of, best performance-algorithm, here propose a software to using regression approach for predicting the sales centered on sales data from the past the accuracy of linear regression prediction can be enhanced with this method, polynomial regression, Ridge regression, and Xgboost regression can be determined. So, we can conclude ridge and Xgboost regression gives the better prediction with respect to Accuracy, MAE and RMSE than the Linear and polynomial regression approaches. In future, the forecasting sales and building a sales plan can help to avoid unforeseen cash flow and manage production, staff and financing needs more effectively.

FUTURE ENHANCEMENT

In future, the forecasting sales and building a sales plan can help to avoid unforeseen cash flow and manage production, staff and financing needs more effectively.In future work we can also consider with the ARIMA model which shows the time series graph.

REFERENCES

[1]	Ching Wu Chu and Guoqiang Peter Zhang, “A comparative study of linear and nonlinear models for aggregate retails sales forecasting”, Int. Journal Production Economics, vol. 86, pp. 217- 231, 2003.
[2] Wang, Haoxiang. "Sustainable development and management in consumer electronics using soft computation." Journal of Soft Computing Paradigm (JSCP) 1, no. 01 (2019): 56.- 2. Suma, V., and Shavige Malleshwara Hills. "Data Mining based Prediction of D

[3] Suma, V., and Shavige Malleshwara Hills. "Data Mining based Prediction of Demand in Indian Market for Refurbished Electronics." Journal of Soft Computing Paradigm (JSCP) 2, no. 02 (2020): 101-110

[4] Giuseppe Nunnari, Valeria Nunnari, “Forecasting Monthly Sales Retail Time Series: A Case Study”, Proc. of IEEE Conf. on Business Informatics (CBI), July 2017.

[5]https://halobi.com/blog/sales-forecasting-five-uses/. [Accessed: Oct. 3, 2018]

[6] Zone-Ching Lin, Wen-Jang Wu, “Multiple LinearRegression Analysis of the Overlay Accuracy Model Zone”, IEEE Trans. on Semiconductor Manufacturing, vol. 12, no. 2, pp. 229 – 237, May 1999.

[7] O. Ajao Isaac, A. Abdullahi Adedeji, I. Raji Ismail, “Polynomial Regression Model of Making Cost Prediction In Mixed Cost Analysis”, Int. Journal on Mathematical Theory and Modeling, vol. 2, no. 2, pp. 14 – 23, 2012.

[8] C. Saunders, A. Gammerman and V. Vovk, “Ridge Regression Learning Algorithm in Dual Variables”, Proc. of Int. Conf. on Machine Learning, pp. 515 – 521, July 1998.IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 56, NO. 7, JULY 2010 3561.

[9] ”Robust Regression and Lasso”. Huan Xu, Constantine Caramanis, Member, IEEE, and Shie Mannor, Senior Member, IEEE. 2015 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration.”An improved Adaboost algorithm based on uncertain functions”.Shu Xinqing School of Automation Wuhan University of Technology.Wuhan, China Wang Pan School of the Automation Wuhan University of Technology Wuhan, China.

[10] Xinqing Shu, Pan Wang, “An Improved Adaboost Algorithm based on Uncertain Functions”, Proc. of Int. Conf. on Industrial Informatics – Computing Technology, Intelligent Technology, Industrial Information Integration, Dec. 2015.

[11] A. S. Weigend and N. A. Gershenfeld, “Time series prediction: Forecasting the future and understanding the past”, Addison-Wesley, 1994.

[12] N. S. Arunraj, D. Ahrens, A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting, Int. J. Production Economics 170 (2015) 321-335P

[13] D. Fantazzini, Z. Toktamysova, Forecasting German car sales using Google data and multivariate models, Int. J. Production Economics 170 (2015) 97-135.

[14] X. Yua, Z. Qi, Y. Zhao, Support Vector Regression for Newspaper/Magazine Sales Forecasting, Procedia Computer Science 17 (2013) 1055–1062.

[15] E. Hadavandi, H. Shavandi, A. Ghanbari, An improved sales forecasting approach by the integration of genetic fuzzy systems and data clustering: a Case study of the printed circuit board, Expert Systems with Applications 38 (2011) 9392–9399.

[16] P. A. Castillo, A. Mora, H. Faris, J.J. Merelo, P. GarciaSanchez, A.J. Fernandez-Ares, P. De las Cuevas, M.I. Garcia-Arenas, Applying computational intelligence methods for predicting the sales of newly published books in a real editorial business management environment, Knowledge-Based Systems 115 (2017) 133-151.

[17] R. Majhi, G. Panda and G. Sahoo, "Development and performance evaluation of FLANN based model for forecasting of stock markets".Expert Systems with Applications, vol. 36, issue 3, part 2, pp. 6800-6808, April 2009.

[18] Pei Chann Chang and Yen-Wen Wang, "Fuzzy Delphi and back propagation model for sales forecasting in PCB industry", Expert systems with applications, vol. 30,pp. 715-726, 2006.

[19] R. J. Kuo, Tung Lai HU and Zhen Yao Chen "application of radial basis function neural networks for sales forecasting", Proc. of Int. Asian Conference on Informatics in control, automation, and robotics, pp. 325- 328, 2009.

[20] R. Majhi, G. Panda, G. Sahoo, and A. Panda, "On the development of Improved Adaptive Models for Efficient Prediction of Stock Indices using Clonal-PSO (CPSO) and PSO Techniques",International Journal of Business Forecasting and Market Intelligence, vol. 1, no. 1, pp.50-67, 2008.

[21]Suresh K and Praveen O, "Extracting of Patterns Using Mining Methods Over Damped Window," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, pp. 235-241, DOI: 10.1109/ICIRCA48905.2020.9182893.

[22] Shobha Rani, N., Kavyashree, S., & Harshitha, R. (2020). Object Detection in Natural Scene Images Using Thresholding Techniques. Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020, Iciccs, 509–515.

[23] https://www.kaggle.com/brijbhushannanda1979/bigmartsales- data. [Accessed: Jun. 28, 2018].