**Six Weeks Summer Training Report**

on

**Data Science By IBM and Allsoft solutions**

**A training report**

Submitted in partial fulfilment of the requirements for the award of degree of

**B. TECH Computer Science and Engineering (Data Science (ML and AI))**

**Submitted to**

**LOVELY PROFESSIONAL UNIVERSITY**

# PHAGWARA, PUNJAB

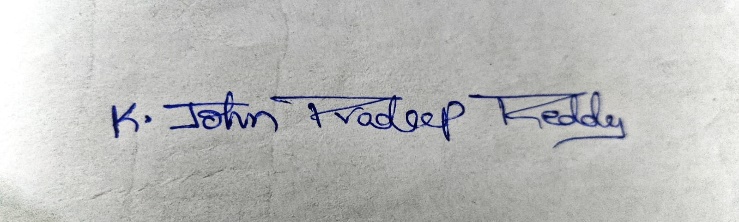


**From 06/06/2023 to 20/07/2023**

**SUBMITTED BY**

**Name of student: Karri John Pradeep Reddy**

**Registration Number: 12109211**

**Signature of the student:** 

**Student Declaration**

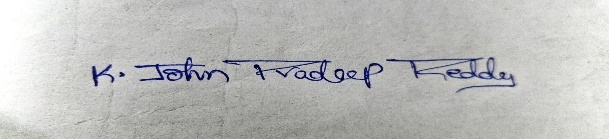
# To whom so ever it may concern

I, **Karri John Pradeep Reddy, 12109211,** hereby declare that the work done by me on “**Data Science**” from **June, 2023** to **July, 2023**, is a record of original work for the partial fulfillment of the requirements for the award of the degree, **B. TECH Computer Science And Engineering (Data Science (ML and AI)).**

Name of Student : Karri John Pradeep Reddy

Registration no : 12109211

Dated :

Signature of the Student:

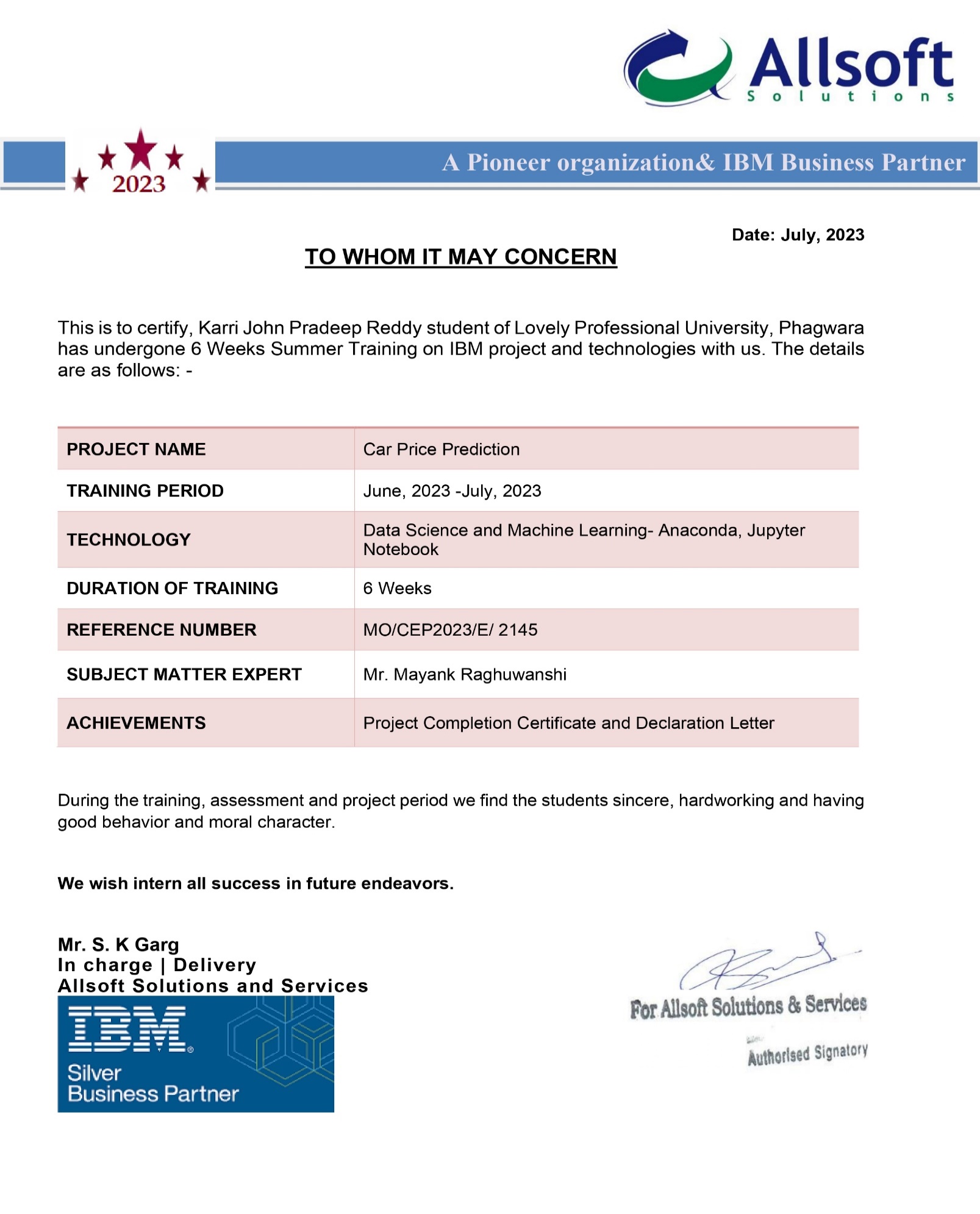
**Acknowledgement:**Top of Form

With heartfelt appreciation, I would like to extend my acknowledgment to the collective efforts of numerous well-wishers who, in their own unique ways, have contributed to the successful completion of the Summer Training. Accomplishing any technological endeavour is a collaborative effort, reliant on the support of many individuals. In preparing this report, I have also sought assistance from various sources. It is now my endeavour to express my profound gratitude to those who offered their valuable assistance.

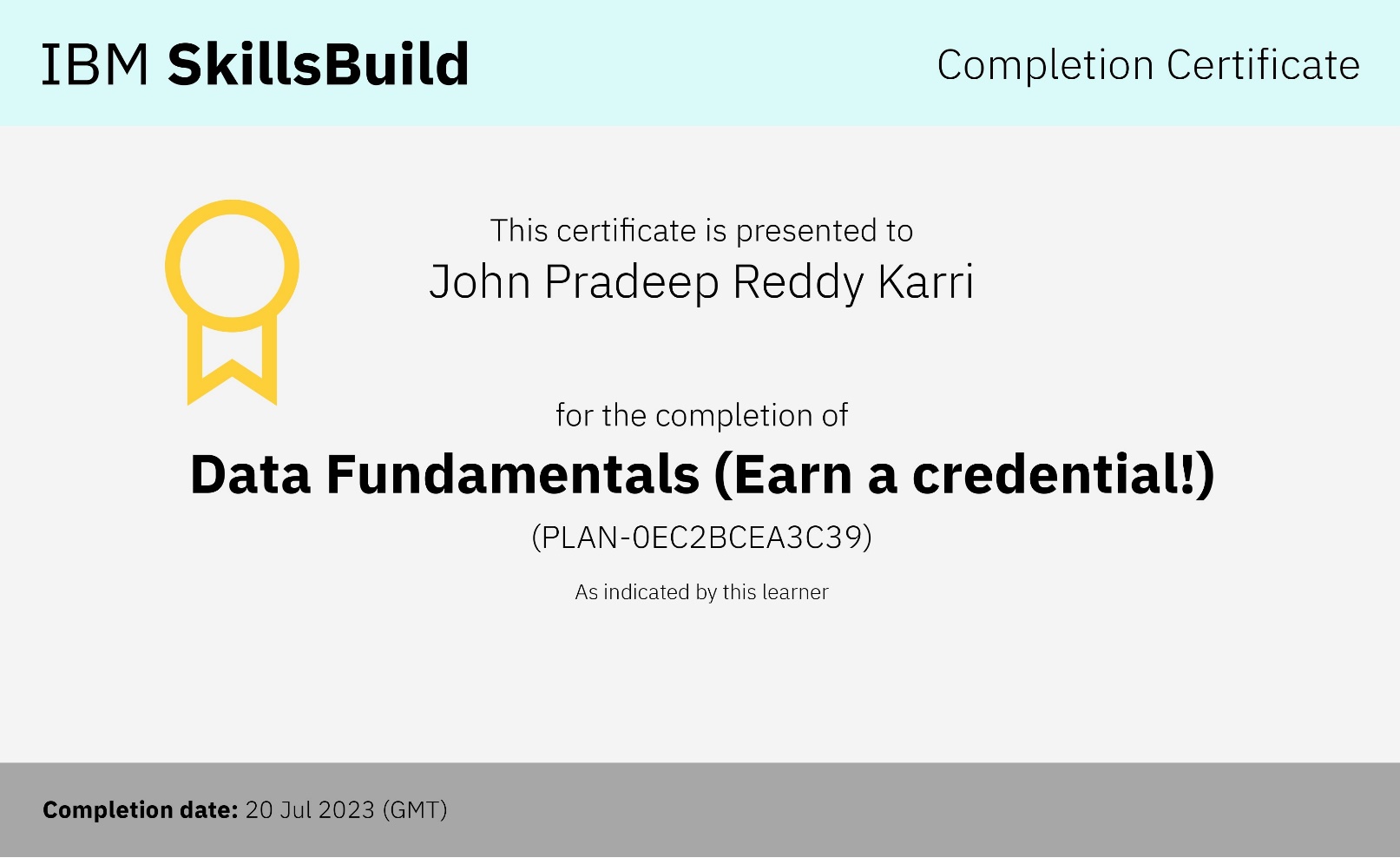
First and foremost, I wish to convey my deep gratitude and indebtedness to our Training mentor, Mr. Mayank Raghuwanshi and Mr. Abdul . their unwavering support and guidance throughout the training have been instrumental in my journey. Without his valuable insights and direction, this would not have achieved the level of success it has. At every step of the project, his supervision and counsel have played a pivotal role in shaping this training experience into a resounding accomplishment.

Project Completion Certificate:

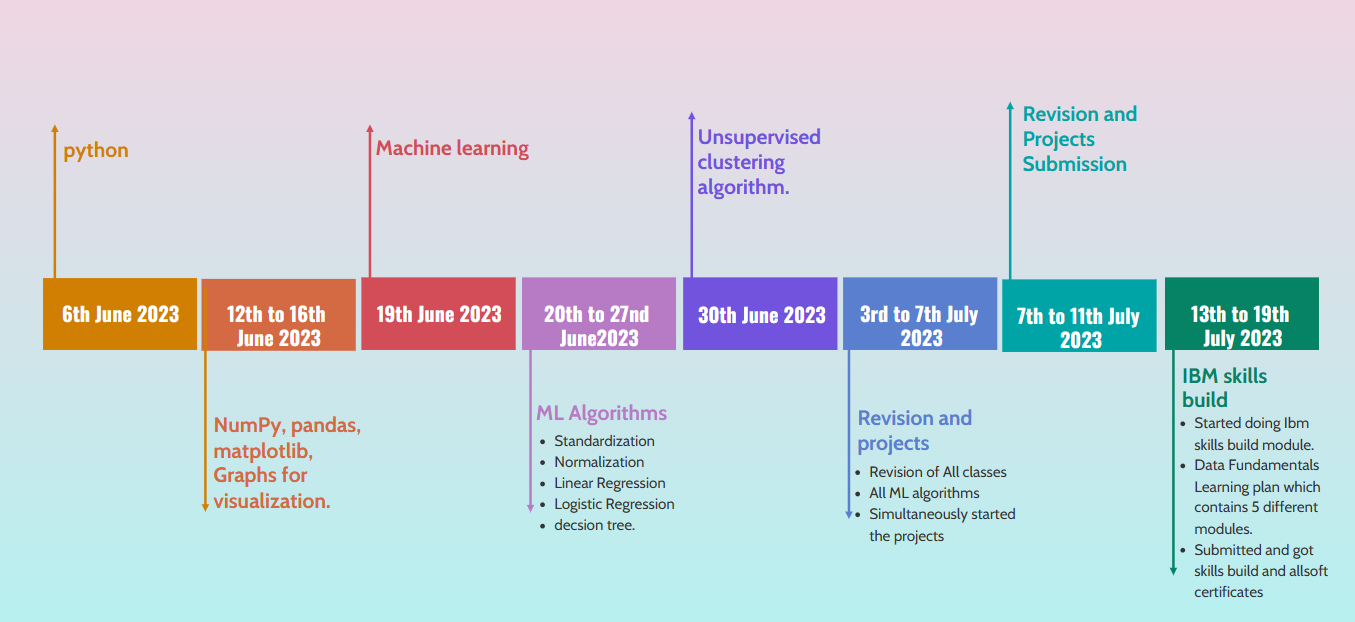


Declaration Letter: 

***IBM Skills build Certificate:***

******

***TimeLine Of Summer Training:***

******

***A cover of a book

Description automatically generated***

**TABLE OF CONTENTS:**

1. Introduction to Data Science page no: 9
2. Applications of Data Science page no: 10
3. Python Introduction page no: 11 to 12
4. Statistics page no: 12 to 13
5. Predictive Modelling page no: 13to 17

• Stages of Predictive Modelling

1. Model Building page no: 18 to 19
2. Algorithms of Machine Learning. Page no 22 to 23
3. Big Data. Page no 23 to 27
4. Project. Page no 28 to 38
5. Reason for choosing data science. Page no 39
6. Learning Outcome and future scope Page no 40
7. Bibliography. Page no 41

**Introduction to Data Science**

**Data Science**

The field of bringing insights from data using scientific techniques is called **data science**

**Amazon Go –** No checkout line.

**Computer Vision -** The advancement in recognizing an image by a computer involves processing large sets of image data from multiple objects of same category. For example, Face recognition.

# Spectrum of Business Analysis:

What can happen?

Given data is

collected and used.

**Big Data**

Why

did it

happen?

**Detective Analysis**

What’s happening

now?

**Dashboards**

What is likely to

happen?

Predictive Analysishappen? Predictive Analysis is likely to

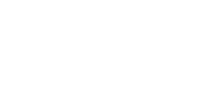
happen?

**Predictive**

**analysis**

What happened?

**Reporting**



Complexity



Value added to organization.

Reporting / Management Information System

To track what is happening in organization.

Detective Analysis

Asking questions based on data we are seeing, like. Why something happened?

Dashboard / Business Intelligence

Utopia of reporting. Every action about business is reflected in front of screen.

Predictive Modelling

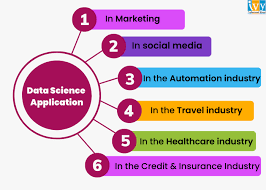
Using past data to predict what is happening at granular level.

## Big Data

Stage where complexity of handling data gets beyond the traditional system.

Can be caused because of volume, variety, or velocity of data. Use specific tools to analyse such scale data.

## Application of Data Science



• Recommendation System

Example-In Amazon recommendations are different for different users according to their past search.

### • Social Media

1. Recommendation Engine
2. Ad placement
3. Sentiment Analysis

* Deciding the right credit limit for credit card customers.
* Suggesting right products from e-commerce companies
  1. Recommendation System
  2. Past Data Searched
  3. Discount Price Optimization
* How google and other search engines know what are the more relevant results for our search query?
  1. Apply ML and Data Science
  2. Fraud Detection
  3. AD placement
  4. Personalized search results

## Python Introduction

Python is a general-purpose, interpreted programming language. Its object-oriented programming methodology is straightforward but efficient, and it includes good high-level data structures. Python is a fantastic language for scripting and quick application development in many domains on most platforms because to its clean syntax, dynamic typing, and nature of being an interpreted language.

**Python for Data science:**

Why Python???

1. Python is an open source language.
2. Syntax as simple as English.
3. Very large and Collaborative developer community.
4. Extensive Packages.

* UNDERSTANDING OPERATORS:
  + Theory of operators: - Operators are symbolic representation of Mathematical tasks.
* VARIABLES AND DATATYPES:
* Variables are named bounded to objects. Data types in python are int (Integer), Float, Boolean and strings.
* CONDITIONAL STATEMENTS:
  + If-else statements (Single condition)
  + If- elif- else statements (Multiple Condition)
* FUNCTIONS:
  + Functions are re-usable piece of code. Created for solving specific problem.
  + Two types: Built-in functions and User- defined functions.
  + Functions cannot be reused in python.
* LISTS: A list is an ordered data structure with elements separated by comma and enclosed within square brackets.
* DICTIONARY: A dictionary is an unordered data structure with elements separated by comma and stored as key: value pair, enclosed with curly braces {}.

**Statistics:**

**Descriptive Statistic**

# Mode:

It is a number which occurs most frequently in the data series.

It is robust and is not generally affected much by addition of couple of new values. Code import pandas as pd data=pd.read\_csv( "Mode.csv") //reads data from csv file

data.head() //print first five lines

mode\_data=data['Subject'].mode() //to take mode of subject column print(mode\_data)

**Mean:**

import pandas as pd data=pd.read\_csv( "mean.csv") //reads data from csv file

data.head() //print first five lines

mean\_data=data[Overallmarks].mean() //to take mode of subject column print(mean\_data)

# Median:

Absolute central value of data set.

import pandas as pd data=pd.read\_csv( "data.csv") //reads data from csv file

data.head() //print first five lines

median\_data=data[Overallmarks].median() //to take mode of subject column print(median\_data)

# Types of Variables:

* Continuous – Which takes continuous numeric values. E.g.-marks
* Categorial-Which have discrete values. E.g.- Gender
* Ordinal – Ordered categorial variables. E.g.- Teacher feedback
* Nominal – Unordered categorial variable. E.g.- Gender

## Outliers

Any value which will fall outside the range of the data is termed as a outlier. E.g.- 9700 instead of 97.

### Reasons of Outliers

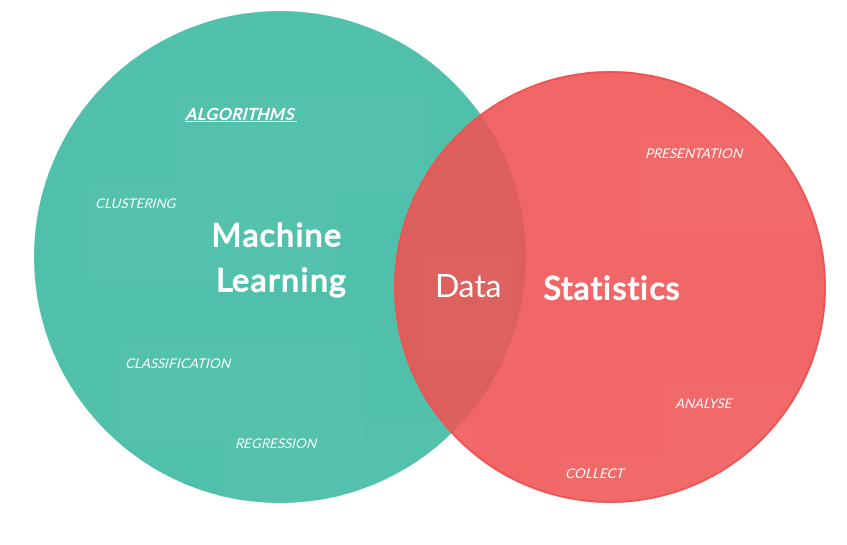
* Typos-During collection. E.g.-adding extra zero by mistake.
* Measurement Error-Outliers in data due to measurement operator being faulty.
* Intentional Error-Errors which are induced intentionally. E.g.-claiming smaller amount of alcohol consumed then actual.
* Legit Outlier—These are values which are not actually errors but in data due to legitimate reasons.

E.g. - a CEO’s salary might be high as compared to other employees. **Interquartile Range (IQR):**

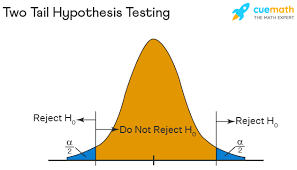
Is difference between third and first quartile from last. It is robust to outliers.

## Histograms

Histograms depict the underlying frequency of a set of discrete or continuous data that are measured on an interval scale.

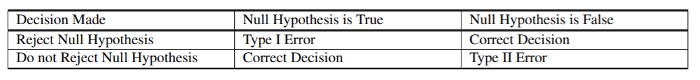
**Inferential Statistics:**

Inferential statistics allows to make inferences about the population from the sample data.



## Hypothesis Testing:

Hypothesis testing is a kind of statistical inference that involves asking a question, collecting data, and then examining what the data tells us about how to proceed. The hypothesis to be tested is called the nullhypothesis and given the symbol Ho. We test the null hypothesis against an alternativehypothesis, which is given the symbol Ha.



## T Tests:

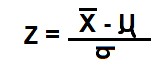
When we have just a sample not population statistics.

Use sample standard deviation to estimate population standard deviation.

T test is more prone to errors, because we just have samples.

## Z Score:

The distance in terms of number of standard deviations, the observed value is away from mean, is standard score or z score.



+Z – value is above mean.

-Z – value is below mean.

The distribution once converted to z- score is always same as that of shape of original distribution.

**Chi Squared Test:**

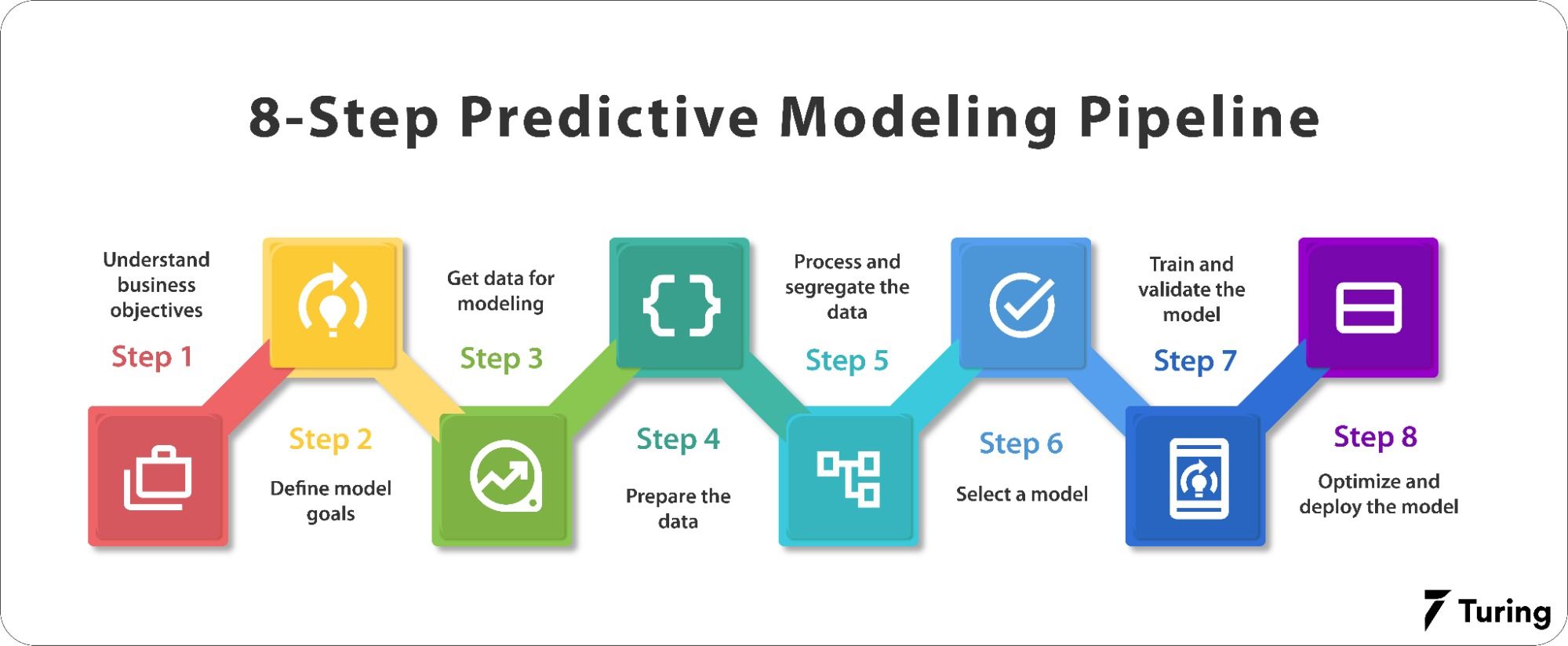
To test categorical variables.

## Correlation:

Determine the relationship between two variables.

It is denoted by r. The value ranges from -1 to +1. Hence, 0 means no relation. Syntax import pandas as pd import numpy as np data=pd.read\_csv("data.csv") data.corr()

## Predictive Modelling:

Making use of past data and attributes we predict future using this data. Eg- 

|  |  |
| --- | --- |
| Past | Horror Movies |
| Future | Unwatched Horror Movies |

### **Predicting stock price movement:**

1. Analysing past stock prices.
2. Analysing similar stocks.
3. Future stock price required.

## Types:

### 1. Supervised Learning

Supervised learning is a type algorithm that uses a known dataset (called the training dataset) to make predictions. The training dataset includes input data and response values.

* Regression-which have continuous possible values. Eg-Marks
* Classification-which have only two values. Eg-Cancer prediction is either 0 or 1.

### 2. Unsupervised Learning

Unsupervised learning is the training of machine using information that is neither classified nor. Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

* **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behaviour.
* **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

## Stages of Predictive Modelling:

1. Problem definition
2. Hypothesis Generation
3. Data Extraction/Collection
4. Data Exploration and Transformation
5. Predictive Modelling
6. Model Development/Implementation

**Problem Definition:**

Identify the right problem statement, ideally formulate the problem mathematically.

## Hypothesis Generation:

List down all possible variables, which might influence problem objective. These variables should be free from personal bias and preferences.

Quality of model is directly proportional to quality of hypothesis.

## Data Extraction/Collection:

Collect data from different sources and combine those for exploration and model building.

While looking at data we might come across new hypothesis.

## Data Exploration and Transformation:

Data extraction is a process that involves retrieval of data from various sources for further data processing or data storage.

*Steps of Data Extraction*

* Reading the data Eg- From csv file
* Variable identification
* Univariate Analysis
* Bivariate Analysis
* Missing value treatment
* Outlier treatment
* Variable Transformation

### Variable Treatment

It is the process of identifying whether variable is

1. Independent or dependent variable
2. Continuous or categorical variable

Why do we perform variable identification?

1. Techniques like supervised learning require identification of dependent variable.
2. Different data processing techniques for categorical and continuous data.

Categorical variable- Stored as object.

Continuous variable-Stored as int or float.

### **Univariate Analysis:**

1. Explore one variable at a time.
2. Summarize the variable.
3. Make sense out of that summary to discover insights, anomalies, etc.

### **Bivariate Analysis:**

* When two variables are studied together for their empirical relationship.
* When you want to see whether the two variables are associated with each other.
* It helps in prediction and detecting anomalies.

**Missing Value Treatment:**

Reasons of missing value

1. Non-response – E.g.-when you collect data on people’s income and many choose not to answer. 2. Error in data collection. E.g.- Faculty data

3. Error in data reading.

### **Types:**

1. MCAR (Missing completely at random): Missing values have no relation to the variable in which missing value exist and other variables in dataset.
2. MAR (Missing at random): Missing values have no relation to the in which missing value exist and the variables other than the variables in which missing values exist.
3. MNAR (Missing not at random): Missing values have relation to the variable in which missing value exists

### Identifying

1. describe():- gives statistical analysis.

2. Isnull() :- Output will we in True or False

### **Different methods to deal with missing values:**

1. Imputation :-

Continuous-Impute with help of mean, median or regression mode.

Categorical-With mode, classification model.

1. Deletion :-

Row wise or column wise deletion. But it leads to loss of data.

### **Outlier Treatment:**

### Reasons of Outliers:

1. Data entry Errors
2. Measurement Errors
3. Processing Errors
4. Change in underlying population

Types of Outlier

# Univariate

Analysing only one variable for outlier. Eg – In box plot of height and weight.

Weight will we analysed for outlier

# Bivariate

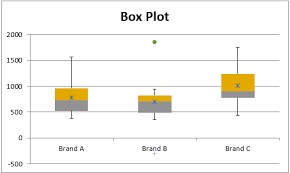
Analysing both variables for outlier.

Eg- In scatter plot graph of height and weight. Both will we analysed.

Identifying Outlier

# Graphical Method

* Box Plot :



* Scatter Plot :

# Formula Method

Using Box Plot

< Q1 - 1.5 \* IQR or > Q3+1.5 \* IQR

Where IQR= Q3 – Q1

Q3=Value of 3rd quartile

Q1=Value of 1st quartile

**Outliers:**

1. Deleting observations.
2. Transforming and binning values.
3. Imputing outliers like missing values.
4. Treat them as separate.

**Variable transformation is a process in which:**

1.We replace a variable with some function of that variable. Eg – Replacing a variable x with its log.

2.We change the distribution or relationship of a variable with others. Used to –

3.Change the scale of a variable

4.Transforming non-linear relationships into linear relationship

5.Creating symmetric distribution from skewed distribution.

Common methods of Variable Transformation – Logarithm, Square root, Cube root, Binning, etc.

## 

## Model Building:

It is a process to create a mathematical model for estimating / predicting the future based on past data.

E.g.

A retail wants to know the default behaviour of its credit card customers. They want to predict the probability of default for each customer in next three months.

* Probability of default would lie between 0 and 1.
* Assume every customer has a 10% default rate.

Probability of default for each customer in next 3 months=0.1

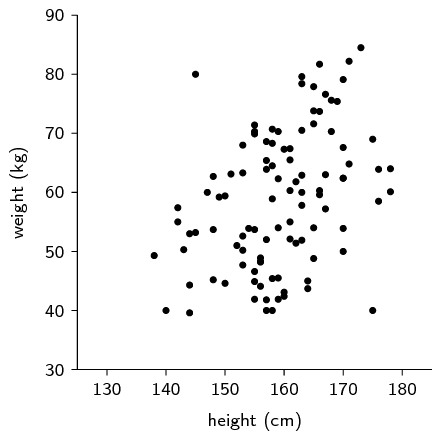
It moves the probability towards one of the extremes based on attributes of past information.

A customer with volatile income is more likely (closer to) to default.

A customer with healthy credit history for last years has low chances of default (closer to 0).

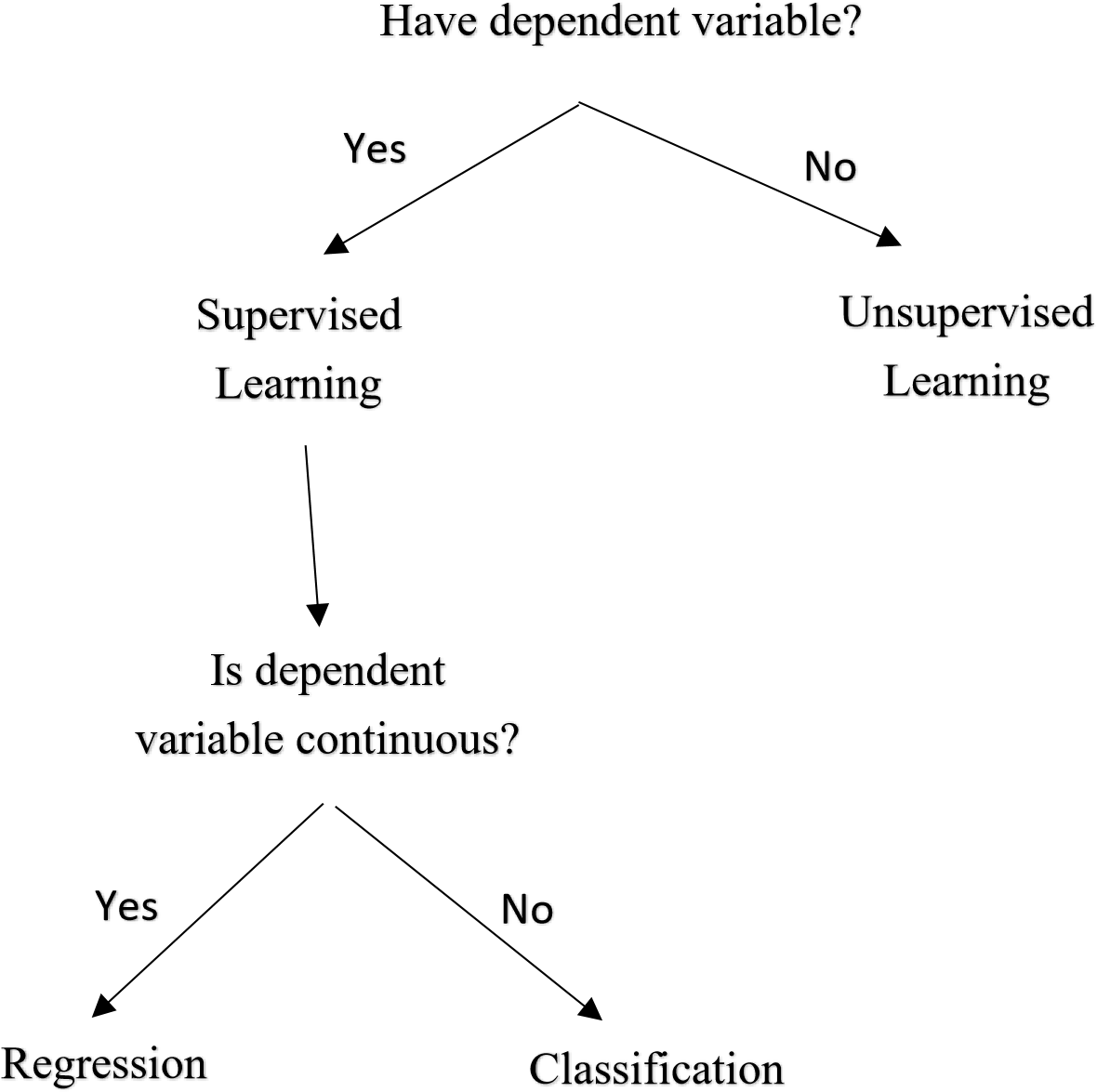
### **Steps in Model Building:**

1. Algorithm Selection
2. Training Model
3. Prediction / Scoring



### **Algorithm Selection:**

Example-



Eg- Predict the customer will buy product or not.

# Algorithms:

* Logistic Regression
* Decision Tree
* Random Forest

## Training Model

It is a process to learn relationship / correlation between independent and dependent variables.

We use dependent variable of train data set to predict/estimate.

**Dataset**

* Train

Past data (known dependent variable).

Used to train model.

* Test

Future data (unknown dependent variable) Used to score. Prediction / Scoring

It is the process to estimate/predict dependent variable of train data set by applying model rules.

# Algorithms of Machine Learning:

## Linear Regression:

Linear regression is a statistical approach for modelling relationship between a dependent variable with a given set of independent variables.

It is assumed that the wo variables are linearly related. Hence, we try to find a linear function. That predicts the response value(y) as accurately as possible as a function of the feature or independent variable(x).

0

2

4

6

8

10

12

14

0

1

2

3

4

5

6

7

8

9

Y

-

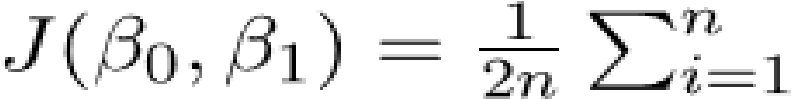
Values

The equation of regression line is

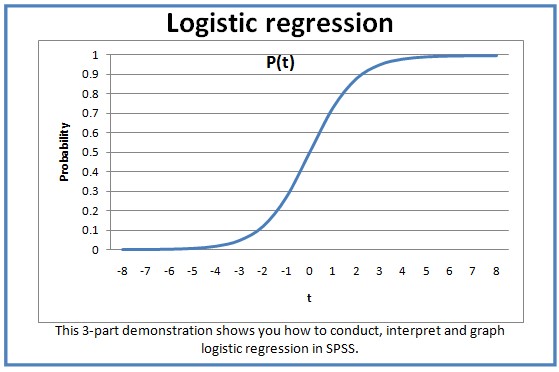
represented as:

T

he squared error or cost function, J as:



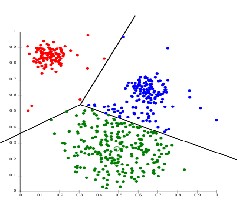
## Logistic Regression:

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) that in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent variable,](https://en.wikipedia.org/wiki/Dependent_variable) although many more complex [extensions](https://en.wikipedia.org/wiki/Logistic_regression#Extensions) exist.

C = -y (log(y) – (1-y) log(1-y))

## K-Means Clustering (Unsupervised learning):

K-means clustering is a type of unsupervised learning, which is used when you have unlabelled data (i.e., data without defined categories or groups). The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.



**Introduction to Big Data:**

**Why is data Important?**

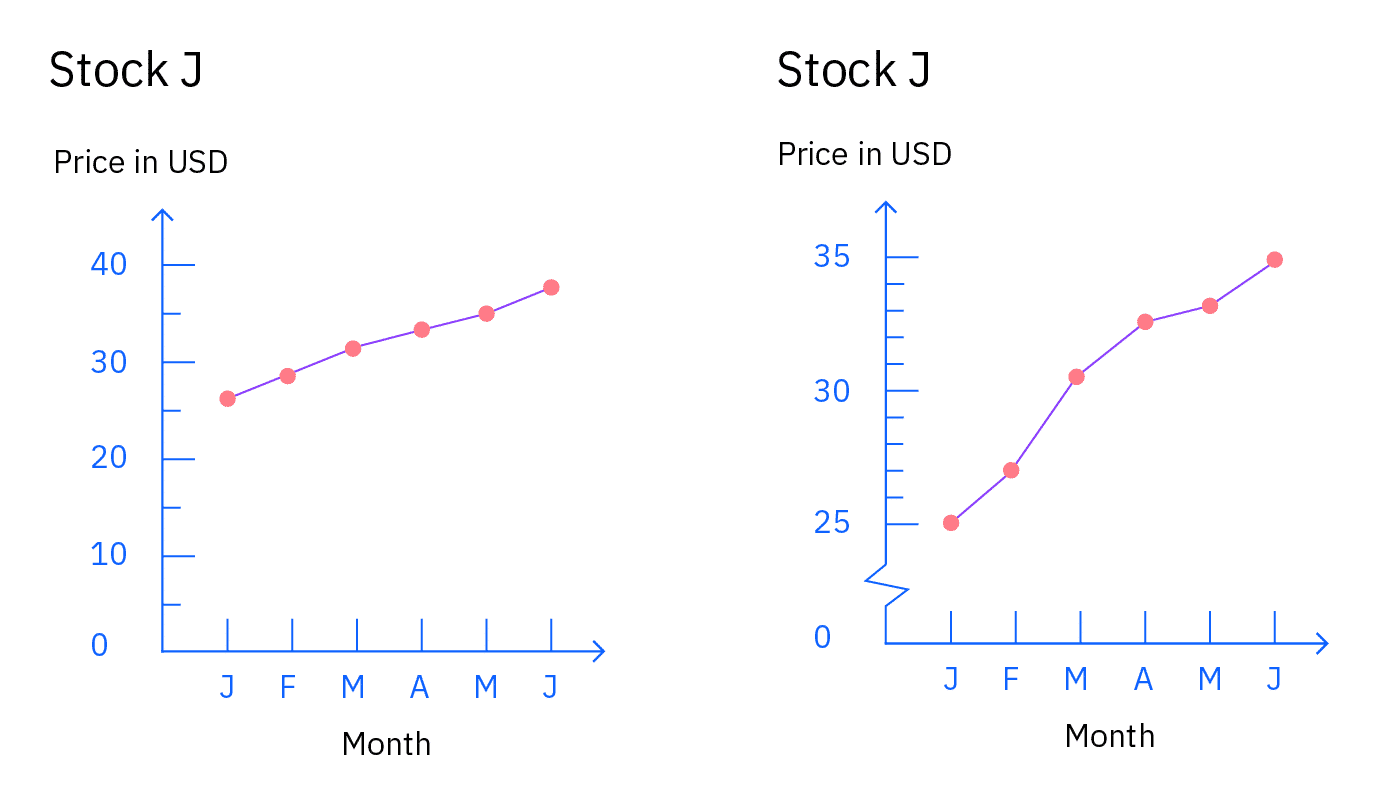
Data is critical for business and has big value. Data is one of the most valuable assets organizations can have, whether in business, finance, healthcare, retail, technology, marketing, or other industries. The number of companies using data insights continues to grow. Data insights have the potential to help many companies:

* Improve operations.
* Better understand end users or customers.
* Drive efficiency.
* Reduce costs.
* Increase profits.
* Find new innovations.
* **Data is a problem solver.**

Data analysts spend a lot of time working in a database. A database is an organized collection of structured data in a computer system. Transforming data into standard format (or tidy data) makes storage and analysis easier.

**Then what is Big Data:**

There is no official definition for big data, but according to tech giants Big Data is high-volume, high-velocity, and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making and process automation.

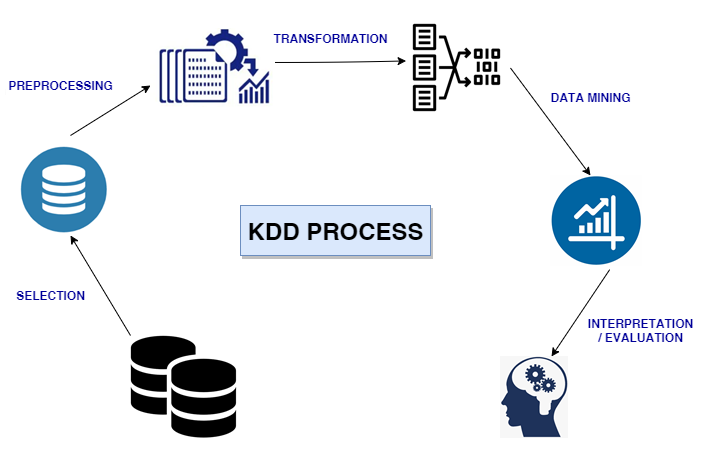


**Data can be misleading.**

While each line chart presents the trending price over time for "Stock J", the vertical scale (or y-axis) for Price is different. The scales show the data in two different increments. Notice the second chart is misleading because it doesn’t depict $0 to 25 for Price like the first chart does.

And, it shows Price in $5 increments. It makes it look like "Stock J" increased in price faster! The first chart is a more accurate depiction because it does not skip the price from $0 to $25 and shows Price consistently in $10 increments.

The key point here is to be precise in how you choose to depict data.

There are four types of data analytics that answer key questions, build on each other, and increase in complexity:

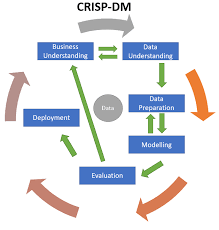
* Descriptive
* Diagnostic
* Predictive
* Prescriptive

**There are three classic and widely adopted data science methodologies:**

* CRISP-DM stands for Cross-Industry Standard Process for Data Mining. consists of six phases with arrows indicating the most important and frequent dependencies between phases:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modelling
5. Evaluation
6. Deployment

* KDD stands for Knowledge Discovery in Database.

1. Selection
2. Preprocessing
3. Transformation
4. Data Mining
5. Interpretation/Evaluation

* SEMMA stands for its five steps:

1. **S**ample
2. **E**xplore
3. **M**odify
4. **M**odel
5. **A**ssess

**Overview of Data Tools and Languages:**

* open-source industry tools. Git and GitHub are two related, but separate, platforms that are extremely popular and widely used by open-source contributors.

1. Host your own open source project. To do this, you create an online repository and add files.
2. Contribute to an existing open source project that’s public. To do this, you access a copy of the project’s repository, make updates, and request a review of the changes to you want to contribute.

* Structured Query Language (SQL) is a standard language to communicate with databases. With SQL, you can:

1. Execute queries against a database.
2. Retrieve data from a database.
3. Insert records in a database.
4. Update records in a database.
5. Delete records from a database.
6. Create new databases.
7. Create new tables in a database.
8. Create stored procedures in a database.
9. Create views in a database.
10. Set permissions on tables, procedures, and views.

* Python:

1. You can use Python to connect to database systems and
2. read and modify files.
3. Python can handle big data and perform complex mathematics.
4. You can pair Python with a data manipulation and analysis software library, like pandas. Python can help you obtain insights and create data visualizations.
5. Python is very popular for data analysis also it is open source programming language.

* IBM Watson Studio: It's a collaborative data science and machine learning environment.

1. IBM Watson Studio works with open source tools.
2. IBM Watson Studio offers a graphical interface with built-in operations. 
3. You don’t need to know how to code to use the tool.
4. And, IBM Watson Studio has a built-in data refinery tool.

* Tableau:

1. Analyze large volumes of data.
2. Create different dashboards, charts, graphics, maps, stories and more to help make business decisions.
3. Perform tasks without programming experience. It offers an intuitive interface.
   * Design interactive visualizations.
4. Tableau is a popular data visualization and business intelligence software for deriving meaningful insights from data. Many businesses use Tableau for pictorial and graphical representations of data

* Matplotlib:

1. A Python Matplotlib script is structured so that, in most instances, a few lines of code can generate a visual data plot.
2. You can create different types of plots, such as scatterplots, histograms, bar charts, and more.
3. The visualizations can be static, animated, and interactive.
4. You can export to many different types of file formats.

* Google Sheets is a free tool you can use to perform tasks like entering, analyzing, and visualizing data to make data-driven decisions.

**Project:**

## Problem Description:

Provided with following files: cardata.csv

Divide the data into test data and train data by specifying the target variable selling price.

**Car Price Predictor**

# Importing the required libraries

|  |
| --- |
| **import** numpy **as** np **import** pandas **as** pd  **import** matplotlib.pyplot **as** plt **import** seaborn **as** sns **from** sklearn.model\_selection **import** train\_test\_split **from** sklearn.linear\_model **import** LinearRegression **from** sklearn.linear\_model **import** Lasso **from** sklearn **import** metrics **import** warnings  warnings**.**filterwarnings("ignore") |

[1]:

# Data Collection and Pre-Processing

## Importing the file "car data" using pandas read function and loading it into a pandas data frame

|  |  |
| --- | --- |
| cars**=**pd**.**read\_csv(r"C:\Users\johnp\OneDrive\Desktop\car data.csv") cars |  |

In [3]:

Out[3]: **Car\_Name Year Selling\_Price Present\_Price Kms\_Driven Fuel\_Type Seller\_Type Transmission**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | sx4 | 2013 | 4.75 | 9.54 | 43000 | Diesel |  | Dealer | Manua |
| **2** | ciaz | 2017 | 7.25 | 9.85 | 6900 | Petrol |  | Dealer | Manua |
| **3** | wagon r | 2011 | 2.85 | 4.15 | 5200 | Petrol |  | Dealer | Manua |
| **4** | swift | 2014 | 4.60 | 6.87 | 42450 | Diesel |  | Dealer | Manua |

**

**

In [4]: *#getting number of datapoints using shape attribute* cars**.**shape

|  |  |
| --- | --- |
| Out[4]: | (301, 9) |

In [5]: *#getting the statistical data of each property of the car* cars**.**describe()

Out[5]: **Year Selling\_Price Present\_Price Kms\_Driven Owner**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **count** | 301.000000 | 301.000000 | 301.000000 | 301.000000 | 301.000000 |
| **mean** | | 2013.627907 | 4.661296 | 7.628472 | 36947.205980 | 0.043189 | |
| **std** | | 2.891554 | 5.082812 | 8.644115 | 38886.883882 | 0.247915 | |
| **min** | | 2003.000000 | 0.100000 | 0.320000 | 500.000000 | 0.000000 | |
| **25%** | | 2012.000000 | 0.900000 | 1.200000 | 15000.000000 | 0.000000 | |
| **50%** | | 2014.000000 | 3.600000 | 6.400000 | 32000.000000 | 0.000000 | |
| **75%** | | 2016.000000 | 6.000000 | 9.900000 | 48767.000000 | 0.000000 | |
| **max** | | 2018.000000 | 35.000000 | 92.600000 | 500000.000000 | 3.000000 | |

In [6]: *#getting some information about the data points in the data set* cars**.**info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 301 entries, 0 to 300 Data columns (total 9 columns):

# Column Non-Null Count Dtype --- ------ -------------- ----- 0 Car\_Name 301 non-null object

1. Year 301 non-null int64
2. Selling\_Price 301 non-null float64 3 Present\_Price 301 non-null float64
3. Kms\_Driven 301 non-null int64
4. Fuel\_Type 301 non-null object
5. Seller\_Type 301 non-null object 7 Transmission 301 non-null object 8 Owner 301 non-null int64 dtypes: float64(2), int64(3), object(4) memory usage: 21.3+ KB

In [7]: *#checking if there are any null values in the data set* cars**.**isnull()**.**sum()

Car\_Name 0 Out[7]:

Year 0 Selling\_Price 0

Present\_Price 0

Kms\_Driven 0

Fuel\_Type 0 Seller\_Type 0

Transmission 0 Owner 0 dtype: int64

## There are no "NAN" values in the data set

In [8]: *#getting the distribution of categorical data in the data set* print(cars**.**Fuel\_Type**.**value\_counts()) print(cars**.**Seller\_Type**.**value\_counts()) print(cars**.**Transmission**.**value\_counts())

Petrol 239 Diesel 60

CNG 2

Name: Fuel\_Type, dtype: int64

Dealer 195

Individual 106 Name: Seller\_Type, dtype: int64

Manual 261

Automatic 40

Name: Transmission, dtype: int64

In [9]: cars**.**Car\_Name**.**unique()

array(['ritz', 'sx4', 'ciaz', 'wagon r', 'swift', 'vitara brezza',

Out[9]:

's cross', 'alto 800', 'ertiga', 'dzire', 'alto k10', 'ignis',

'800', 'baleno', 'omni', 'fortuner', 'innova', 'corolla altis',

'etios cross', 'etios g', 'etios liva', 'corolla', 'etios gd',

'camry', 'land cruiser', 'Royal Enfield Thunder 500',

'UM Renegade Mojave', 'KTM RC200', 'Bajaj Dominar 400', 'Royal Enfield Classic 350', 'KTM RC390', 'Hyosung GT250R',

'Royal Enfield Thunder 350', 'KTM 390 Duke ',

'Mahindra Mojo XT300', 'Bajaj Pulsar RS200',

'Royal Enfield Bullet 350', 'Royal Enfield Classic 500',

'Bajaj Avenger 220', 'Bajaj Avenger 150', 'Honda CB Hornet 160R',

'Yamaha FZ S V 2.0', 'Yamaha FZ 16', 'TVS Apache RTR 160',

'Bajaj Pulsar 150', 'Honda CBR 150', 'Hero Extreme',

'Bajaj Avenger 220 dtsi', 'Bajaj Avenger 150 street',

'Yamaha FZ v 2.0', 'Bajaj Pulsar NS 200', 'Bajaj Pulsar 220 F',

'TVS Apache RTR 180', 'Hero Passion X pro', 'Bajaj Pulsar NS 200',

'Yamaha Fazer ', 'Honda Activa 4G', 'TVS Sport ',

'Honda Dream Yuga ', 'Bajaj Avenger Street 220',

'Hero Splender iSmart', 'Activa 3g', 'Hero Passion Pro',

'Honda CB Trigger', 'Yamaha FZ S ', 'Bajaj Pulsar 135 LS',

'Activa 4g', 'Honda CB Unicorn', 'Hero Honda CBZ extreme',

'Honda Karizma', 'Honda Activa 125', 'TVS Jupyter',

'Hero Honda Passion Pro', 'Hero Splender Plus', 'Honda CB Shine',

'Bajaj Discover 100', 'Suzuki Access 125', 'TVS Wego',

'Honda CB twister', 'Hero Glamour', 'Hero Super Splendor',

'Bajaj Discover 125', 'Hero Hunk', 'Hero Ignitor Disc',

'Hero CBZ Xtreme', 'Bajaj ct 100', 'i20', 'grand i10', 'i10',

'eon', 'xcent', 'elantra', 'creta', 'verna', 'city', 'brio',

'amaze', 'jazz'], dtype=object)

We have bikes mixed in the dataset we need to delete the bikes as they decrease the efficiency of the model

All the bikes present price in the dataset is less than 2 lakh so we use this condition to eliminate all the bikes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| bikes**=**cars[cars["Present\_Price"]**<=**2.0] bikes |  |  |  |  |

**100** 2016 1.75

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **101** | UM  Renegade Mojave | 2017 | 1.70 | 1.82 | 1400 | Petrol | Individual | Manu |
| **102** | KTM  RC200 | 2017 | 1.65 | 1.78 | 4000 | Petrol | Individual | Manu |
| **103** | Bajaj  Dominar 400 | 2017 | 1.45 | 1.60 | 1200 | Petrol | Individual | Manu |
| **104** | Royal  Enfield  Classic 350 | 2017 | 1.35 | 1.47 | 4100 | Petrol | Individual | Manu |
| **...** | ... | ... | ... | ... | ... | ... | ... |  |
| **196** | Activa 3g | 2008 | 0.17 | 0.52 | 500000 | Petrol | Individual | Automa |
| **197** | Honda CB twister | 2010 | 0.16 | 0.51 | 33000 | Petrol | Individual | Manu |
| **198** | Bajaj  Discover 125 | 2011 | 0.15 | 0.57 | 35000 | Petrol | Individual | Manu |
| **199** | Honda CB  Shine | 2007 | 0.12 | 0.58 | 53000 | Petrol | Individual | Manu |
| **200** | Bajaj  Pulsar 150 | 2006 | 0.10 | 0.75 | 92233 | Petrol | Individual | Manu |

98 rows × 9 columns

**

**

|  |
| --- |
| *# delete all rows with column "Present Price" has value less than 2.0* dropBikes **=** cars[ (cars["Present\_Price"] **<=** 2.0)]**.**index cars**.**drop(dropBikes , inplace**=True**) cars**.**shape |

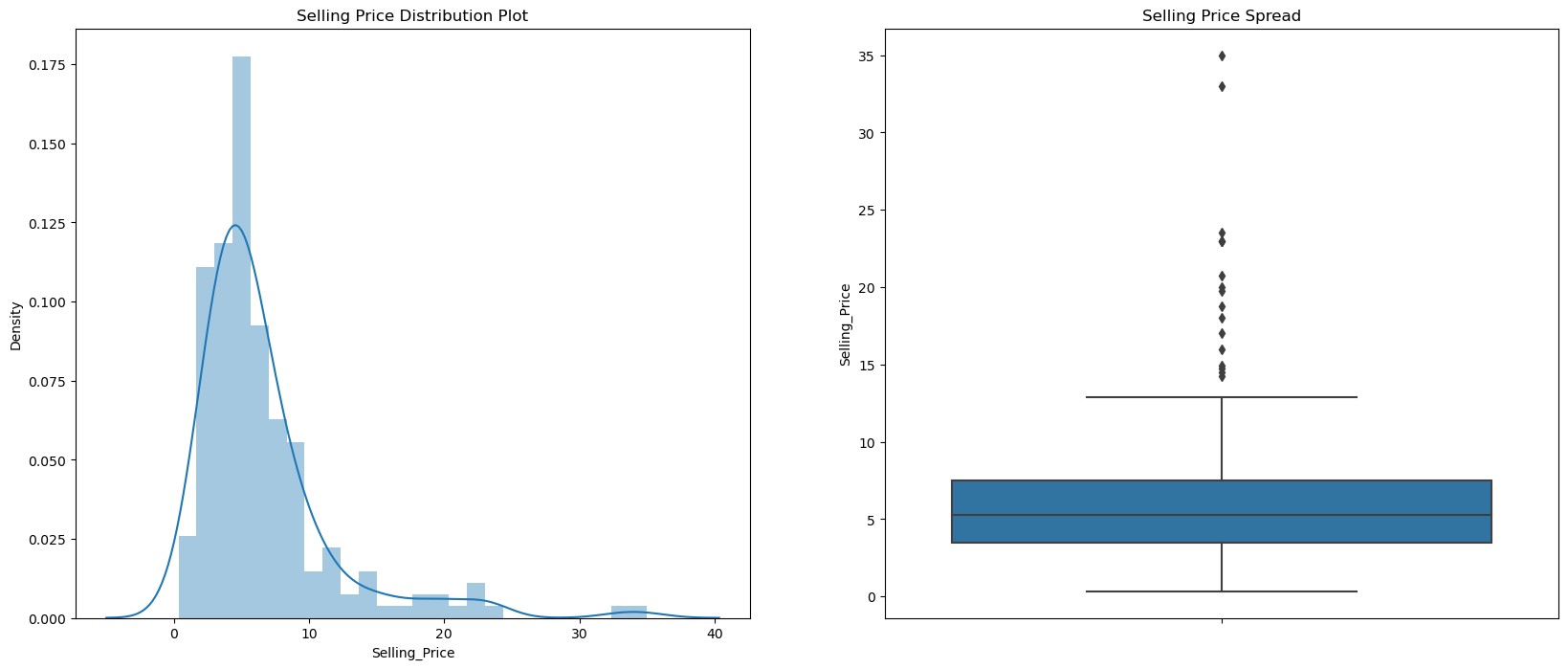
In [11]:

(203, 9) Out[11]:

|  |
| --- |
| plt**.**figure(figsize**=**(20,8)) plt**.**subplot(1,2,1)  plt**.**title('Selling Price Distribution Plot') sns**.**distplot(cars**.**Selling\_Price)  plt**.**subplot(1,2,2)  plt**.**title('Selling Price Spread') |

In [12]:

sns**.**boxplot(y**=**cars**.**Selling\_Price) plt**.**show()



## Inference:-

There is no significant diference between the mean and median the data is not scattered so much

In [13]: print(cars**.**Selling\_Price**.**describe(percentiles **=** [0.25,0.50,0.75,0.85,0.90,1]))

count 203.000000 mean 6.611724 std 5.153441 min 0.350000 25% 3.505000

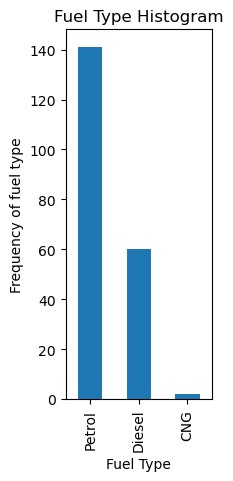
50% 5.250000 75% 7.500000

85% 9.425000

90% 11.490000 100% 35.000000 max 35.000000 Name: Selling\_Price, dtype: float64

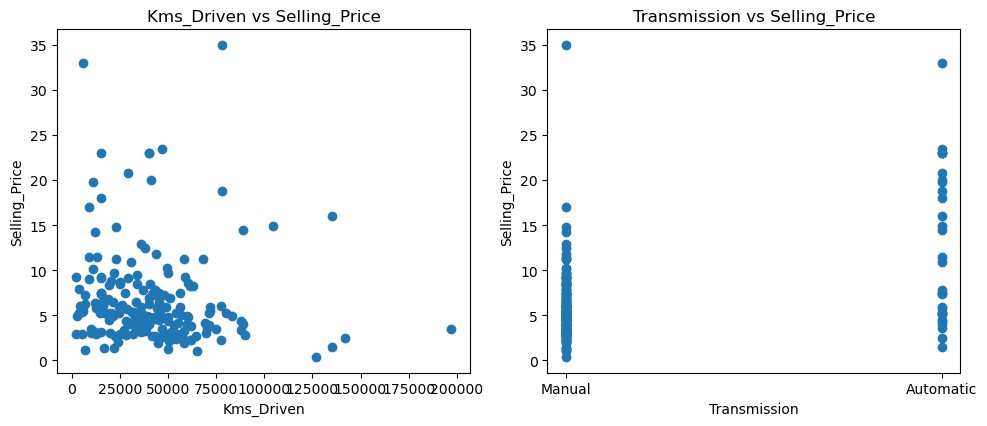
## Visualizing the categorical data

In [14]: plt**.**subplot(1,3,2) plt1 **=** cars**.**Fuel\_Type**.**value\_counts()**.**plot(kind**=**"bar") plt**.**title("Fuel Type Histogram") plt1**.**set(xlabel **=** "Fuel Type", ylabel**=**"Frequency of fuel type") plt**.**show()



|  |
| --- |
| **def** scatter(x,fig):  plt**.**subplot(5,2,fig) plt**.**scatter(cars[x],cars["Selling\_Price"]) plt**.**title(x**+**" vs Selling\_Price") plt**.**ylabel("Selling\_Price") plt**.**xlabel(x) plt**.**figure(figsize**=**(10,20))  scatter("Kms\_Driven", 1) scatter("Transmission", 2) plt**.**tight\_layout() |

In [15]:



## Inference:-

Kms driven is inversly proportional to the selling price for any car.

The lowest price of the cars start with manual gear transmission and the mean price of the automatic transmission is more compared to the manual transmission

## Encoding of categorical data

|  |
| --- |
| *#Encoding the "Fuel\_Type"*  cars**.**replace({"Fuel\_Type":{"Petrol":0,"Diesel":1,"CNG":2}},inplace**=True**)  *#Encoding the "Fuel\_Type"*  cars**.**replace({"Seller\_Type":{"Dealer":0,"Individual":1}},inplace**=** **True**) *#Encoding the "Fuel\_Type"*  cars**.**replace({"Transmission":{"Manual":0,"Automatic":1}},inplace**=** **True**) cars**.**head() |

In [16]:

Out[16]: **Car\_Name Year Selling\_Price Present\_Price Kms\_Driven Fuel\_Type Seller\_Type Transmission**

**0** ritz 2014 3.35 5.59 27000 0 0 0

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | sx4 | 2013 | 4.75 |  | 9.54 | 43000 | 1 | 0 | 0 |
| **2** | ciaz | 2017 | 7.25 |  | 9.85 | 6900 | 0 | 0 | 0 |
| **3** | wagon r | 2011 | 2.85 |  | 4.15 | 5200 | 0 | 0 | 0 |
| **4** | swift | 2014 | 4.60 |  | 6.87 | 42450 | 1 | 0 | 0 |

**

**

### Splitting the Data and Target

|  |  |  |
| --- | --- | --- |
| In [17]: In [18]: | x**=**cars**.**drop(["Car\_Name","Selling\_Price"],axis**=**1) y**=**cars["Selling\_Price"] |  |

**Separating Training data and Test data**

In [20]: x\_train, x\_test, y\_train, y\_test**=**train\_test\_split(x,y,test\_size**=**0.1,random\_state**=**2)

**Model Training**

# Linear Regression

In [21]: *# linear regression model loading* linear\_reg**=**LinearRegression()

In [22]: linear\_reg**.**fit(x\_train,y\_train)

*# the equation is y=mx+c*

LinearRegression()

|  |  |
| --- | --- |
| Out[22]: | Model Evaluation |

In [23]: *#training data prediction* training\_data\_prediction**=**linear\_reg**.**predict(x\_train)

In [24]: *# R-Squared error*

error\_score**=**metrics**.**r2\_score(y\_train, training\_data\_prediction) print("R-Squared error", error\_score)

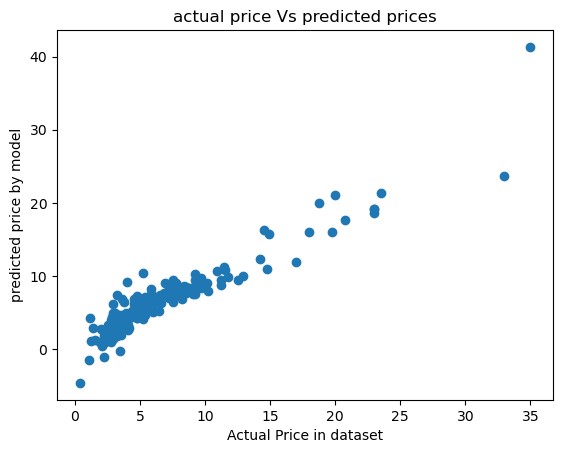
R-Squared error 0.8798333053726144

Another method to predict the accuracy of the outcome is by plotting the values predicted by the model to the actual values

## Visualizing the correct price Vs predicted prices

In [25]: plt**.**scatter(y\_train, training\_data\_prediction) plt**.**xlabel("Actual Price in dataset") plt**.**ylabel("predicted price by model") plt**.**title("actual price Vs predicted prices") plt**.**show

<function matplotlib.pyplot.show(close=None, block=None)> Out[25]:



|  |
| --- |
| *#testing data prediction*  testing\_data\_prediction**=**linear\_reg**.**predict(x\_test) |

In [26]:

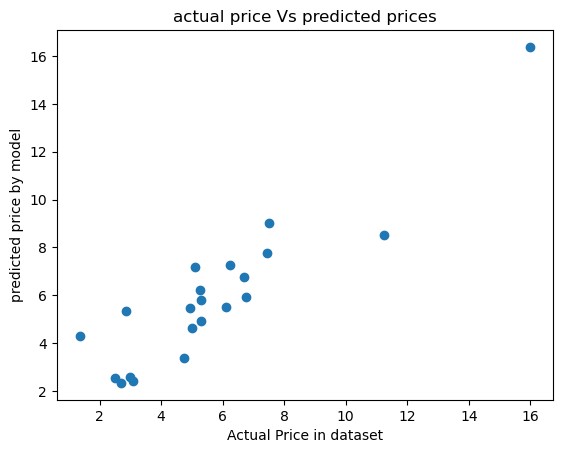
|  |
| --- |
| *# R-Squared error*  error\_score**=**metrics**.**r2\_score(y\_test, testing\_data\_prediction) print("R-Squared error", error\_score) |

In [27]:

|  |
| --- |
| plt**.**scatter(y\_test, testing\_data\_prediction) plt**.**xlabel("Actual Price in dataset") plt**.**ylabel("predicted price by model") plt**.**title("actual price Vs predicted prices") plt**.**show |

R-Squared error 0.8294993134054333 In [28]:

<function matplotlib.pyplot.show(close=None, block=None)> Out[28]:



### Inference:-

By seeing the plot we can say that there isn't much scattering and the graph is linear If we have more data in the dataset we can have better pediction by the way it's not a bad prediction but not upto the mark

We now try the lasso regression because generally if the variables are positively correlated then the linear regression works well but in other cases when more variables are that support the target variables then lasso regression works best

## 2.Lasso Regression

|  |
| --- |
| *# linear regression model loading* lasso\_reg**=**Lasso() |

In [29]:

In [30]: lasso\_reg**.**fit(x\_train,y\_train)

|  |  |
| --- | --- |
| Out[30]: | Lasso() |

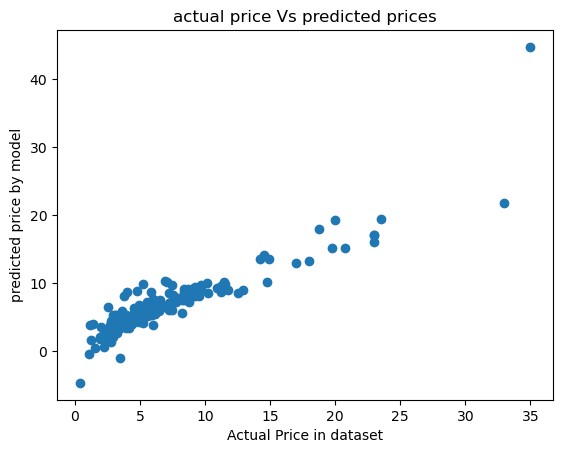
In [31]: *#training data prediction* training\_data\_prediction**=**lasso\_reg**.**predict(x\_train)

In [32]: *# R-Squared error* error\_score**=**metrics**.**r2\_score(y\_train, training\_data\_prediction) print("R-Squared error", error\_score)

R-Squared error 0.8337662123845258

In [33]: plt**.**scatter(y\_train, training\_data\_prediction) plt**.**xlabel("Actual Price in dataset") plt**.**ylabel("predicted price by model") plt**.**title("actual price Vs predicted prices") plt**.**show

<function matplotlib.pyplot.show(close=None, block=None)> Out[33]:



|  |
| --- |
| *#testing data prediction*  testing\_data\_prediction**=**lasso\_reg**.**predict(x\_test) |

In [34]:

|  |
| --- |
| *# R-Squared error*  error\_score**=**metrics**.**r2\_score(y\_test, testing\_data\_prediction) print("R-Squared error", error\_score) |

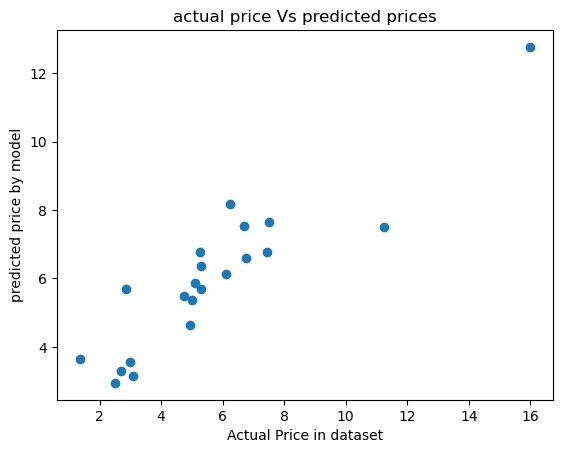
In [35]:

|  |
| --- |
| plt**.**scatter(y\_test, testing\_data\_prediction) plt**.**xlabel("Actual Price in dataset") plt**.**ylabel("predicted price by model") |

R-Squared error 0.767656650980445 In [36]:

plt**.**title("actual price Vs predicted prices") plt**.**show

<function matplotlib.pyplot.show(close=None, block=None)> Out[36]:



## Inference :-

By comparing the two models linear regression is working better than lasso regression but the change is low so I can say that both are best for this project.

If the data set is larger and had more columns then there will be significant change between the both models.

## Reason for choosing data science:

Data Science has become a revolutionary technology that everyone seems to talk about. Hailed as the

‘Sexiest job of the 21st century’. Data Science is a buzzword with very few people knowing about the technology in its true sense.

While many people wish to become Data Scientists, it is essential to weigh the pros and cons of data science and give out a real picture. In this article, we will discuss these points in detail and provide you with the necessary insights about Data Science.

### **Advantages: -**

1. It’s in Demand.
2. Abundance of Positions.
3. A Highly Paid Career.
4. Data Science is Versatile.

### **Disadvantages: -**

1. Mastering Data Science is near to impossible.
2. A large Amount of Domain Knowledge is Required.
3. improper Data May Yield Unexpected Results.
4. The problem of Data Privacy.
5. Need Experienced persons, hence freshers have less openings.

**Learning Outcome:**

After completing the training, I am able to:

* Develop relevant programming abilities.
* Demonstrate insights by statistical analysis of data.
* Demonstrate skill in data management.
* I am able to analyze the data in IBM Watson studio.
* Use machine learning models for data predictions.
* Apply data science concepts and methods to solve problem in real-world contexts and will communicate these solutions effectively.

**Conclusion and Future Scope:-**

* By doing this project I came to know working of machine learning algorithms and using them to create a car price predictor.
* and I came to know that the dataset is not quite good for making good predictions there are bikes mixed in the dataset and after removing bikes from the list the data even became smaller also there should be more features that directly impact the car price like wheel size, ground clearance, engine type, torque and horse power of the engine. then the prediction will be better. Mayank sir and Abdul sir helped me in making this project successful.

**Bibliography:**

* Google.
* IBM Skills build.
* Wikipedia.
* Python documentation.
* Kaggle.
* Geeks for Geeks.