

Python-DA-ML-AI

MENTORED BY : -----

TRAINING ORGANIZATION : Froyo Technology

PRESENTED BY : Bharat Yadav

SUBMITTED TO : Ajay Kumar Garg Engineering College
TRAINING DURATION - 6 WEEKS (1 JULY TO 31 JULY)

Agenda / Topics

- Python Programming Fundamentals.
- Data Analytics Fundamentals
- Working with Pandas
- Working with NumPy
- Working Data Visualization with Matplotlib
- Working with Seaborn
- Machine Learning Fundamentals
 - Linear Regression
 - Logistic Regression
- KNN , Naïve Bayes, Decision Tree Classification
- Project on Machine Learning

• Python Programming Fundamentals.

- Python is a high-level, general-purpose, dynamic, interpreted programming language. It supports multiple programming paradigms, including structured, object-oriented and functional.
- A basic Python curriculum can be broken down into 4 essential topics that include:
 1. Data types (int, float, strings)
 2. Compound data structures (lists, tuples, and dictionaries)
 3. Conditionals, loops, and functions
 4. Object-oriented programming and using external libraries

• Data Analytics Fundamentals

- Data analytics is the science of integrating heterogeneous data from diverse sources, drawing inferences, and making predictions to enable innovation, gain competitive business advantage, and help strategic decision-making.

• Data Analysis in 5 Steps

- STEP 1: DEFINE QUESTIONS & GOALS.
- STEP 2: COLLECT DATA.
- STEP 3: DATA WRANGLING.
- STEP 4: DETERMINE ANALYSIS.
- STEP 5: INTERPRET RESULTS.

Use of Data Analytics



Working with Pandas

- Pandas is an **open source Python package that is most widely used for data science/data analysis and machine learning tasks.** It is built on top of another package named Numpy, which provides support for multi-dimensional arrays.
- **Use of Pandas:-**
 - Data fill
 - Data normalization
 - Merges and joins
 - Data visualization
 - Statistical analysis
 - And much more

Working with NumPy

- NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.
- Use of Numpy :-
 - A powerful N-dimensional array object
 - Sophisticated (broadcasting) functions
 - Tools for integrating C/C++ and Fortran code
 - Useful linear algebra, Fourier transform, and random number capabilities

Working Data Visualization with Matplotlib

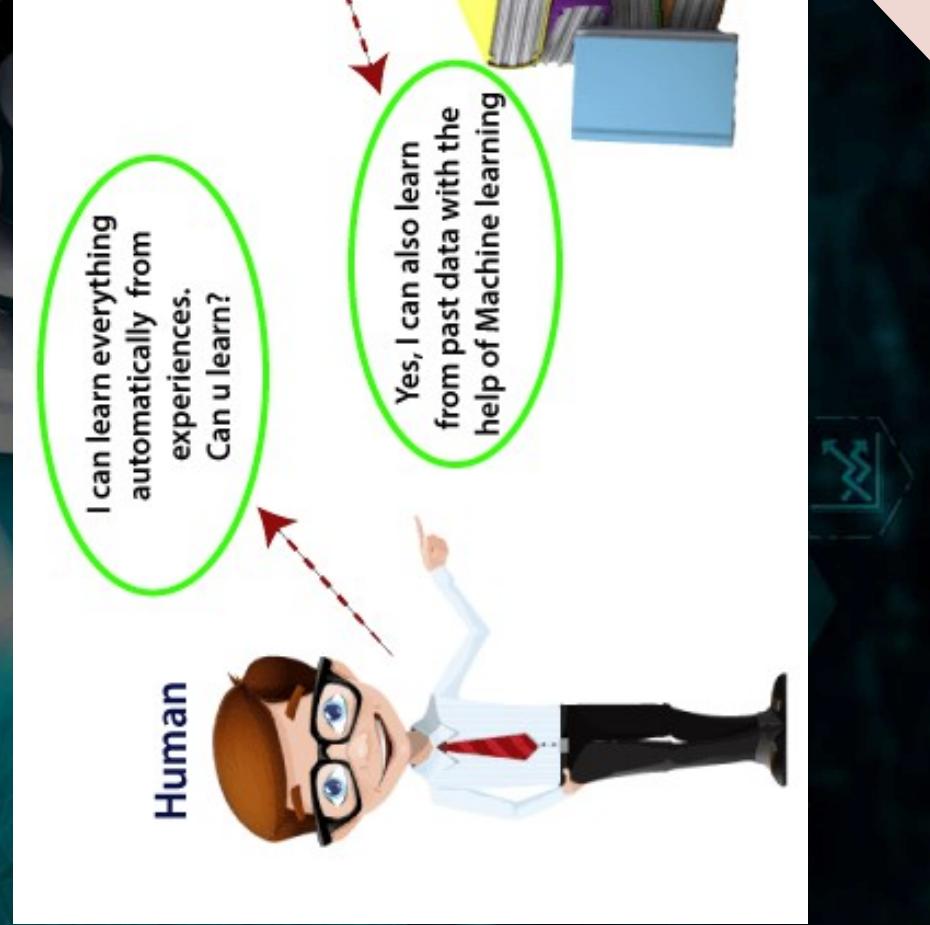
- Data and information visualization is an interdisciplinary field that deals with the representation of data and information. It is a particularly efficient way of communicating when the data or information is numerous as for example a timer
- Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, c

Working with SeaBorn

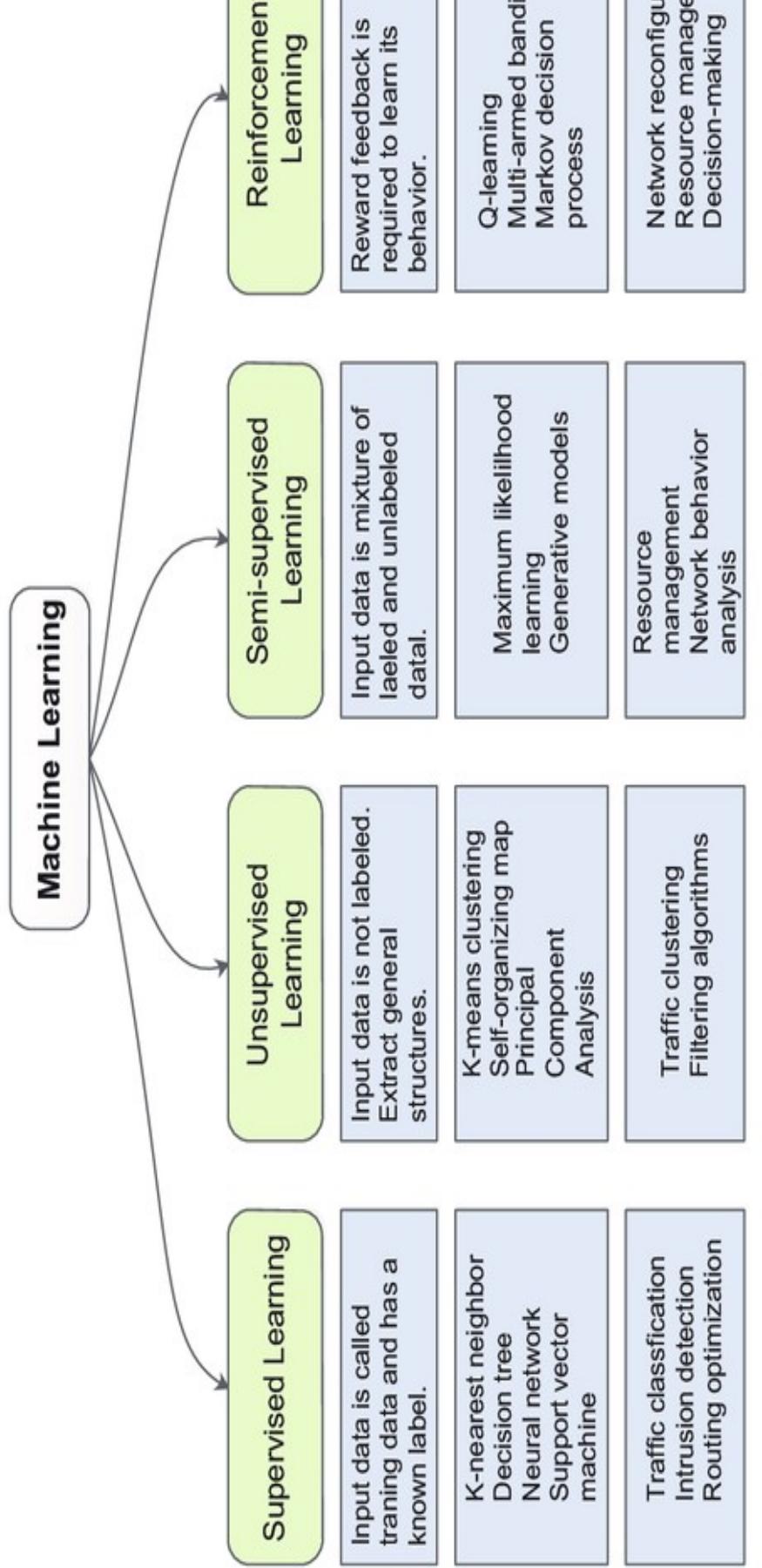
- Seaborn is a **library that uses Matplotlib underneath to plot graphs**. It will be to visualize random distributions.
- To work with seaborn one have to import seaborn as well as matplotlib as seaborn based on matplotlib.
- Seaborn helps to visualize the statistical relationships, To understand how variables are related to one another and how that relationship is dependent on other variables.
- This library is used to visualize our data we do not need to take care of the internal details; we just have to pass our data set or data inside the relplot() function, and calculate and place the value accordingly.

• Machine Learning Fundamentals

- Machine learning is a field of inquiry devoted to understanding and building methods that 'learn', that is, methods that leverage data to improve performance on some set of tasks. It is seen as an application of artificial intelligence.
- Its primary aim is to allow the computers to learn automatically without human intervention and adjust actions accordingly.



• Machine Learning Fundamentals



Linear Regression

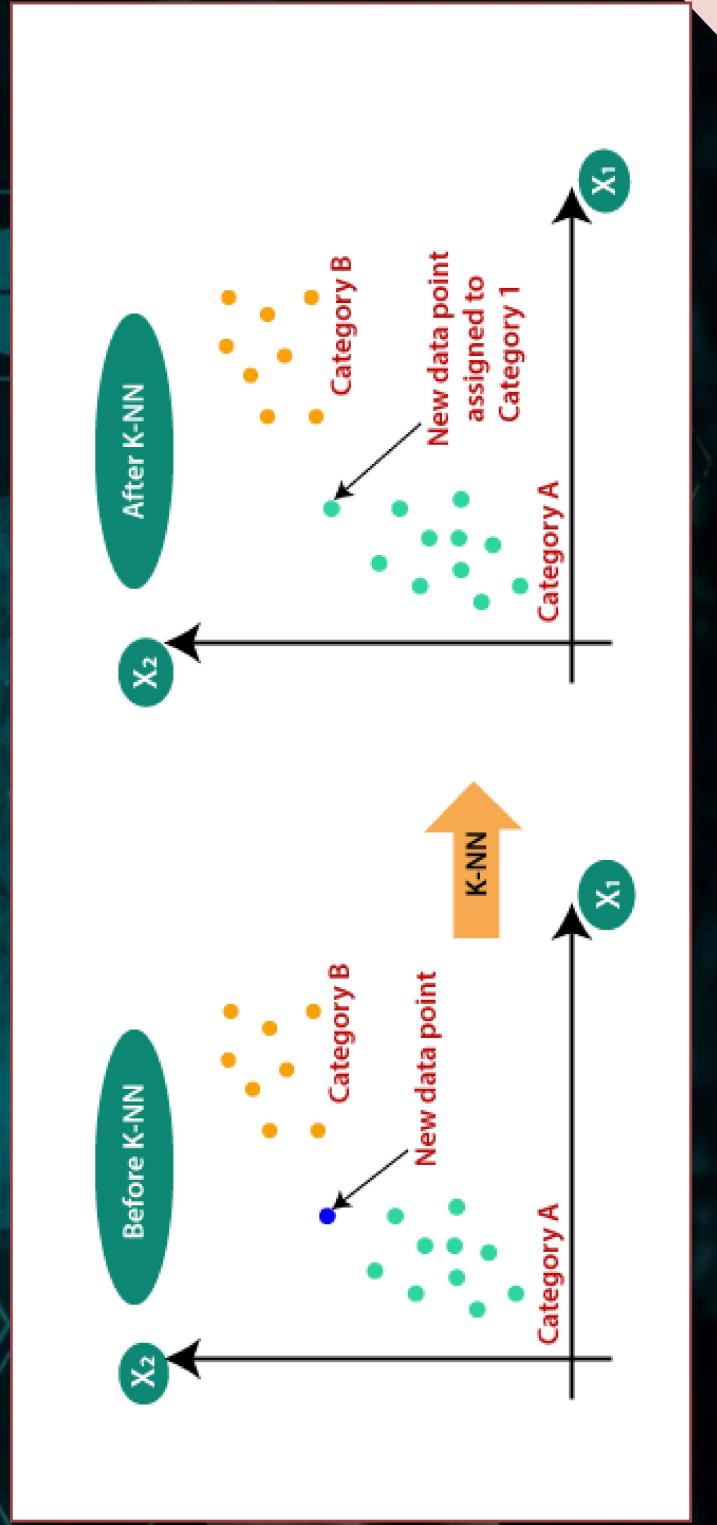
- Linear relationship is a simplest mathematical relationship between two variables X and Y.
- In a cause and effect relationship, the independent variable is the cause , and the dependent variable is the effect.
- Linear regression is a method for predicting the value for dependent variable Y, based on the value of an independent variable X.
- Mathematically we can represent linear regression as, $y = a_0 + a_1x + \epsilon$.
- When working with linear regression, our main goal is to find the best fit line that minimizes the error between predicted values and actual values should be minimized. The best fit line have the least error.
- Linear regression can be further divided into two types of the algorithm:
 - **Simple Linear Regression**
 - **Multiple Linear regression**

Logistic Regression

- Logistic regression is a **statistical analysis method to predict a binary outcome as yes or no, based on prior observations of a data set.**
- A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.
- There are three main types of logistic regression:-
 - **binary.**
 - **multinomial**
 - **ordinal.**

KNN

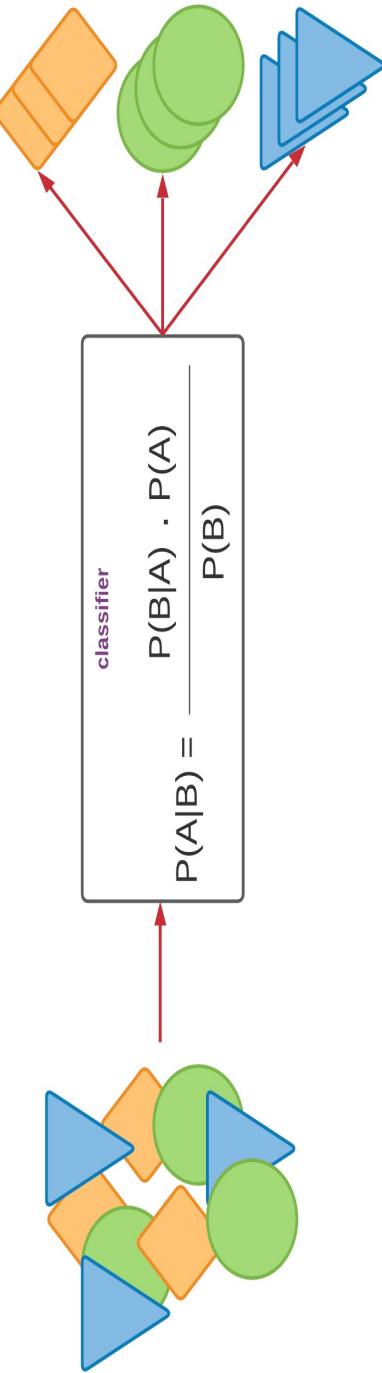
- The abbreviation KNN stands for "**K-Nearest Neighbour**". It is a supervised machine learning algorithm. The algorithm can be used to solve both classification and regression problems.



Naïve Bayes

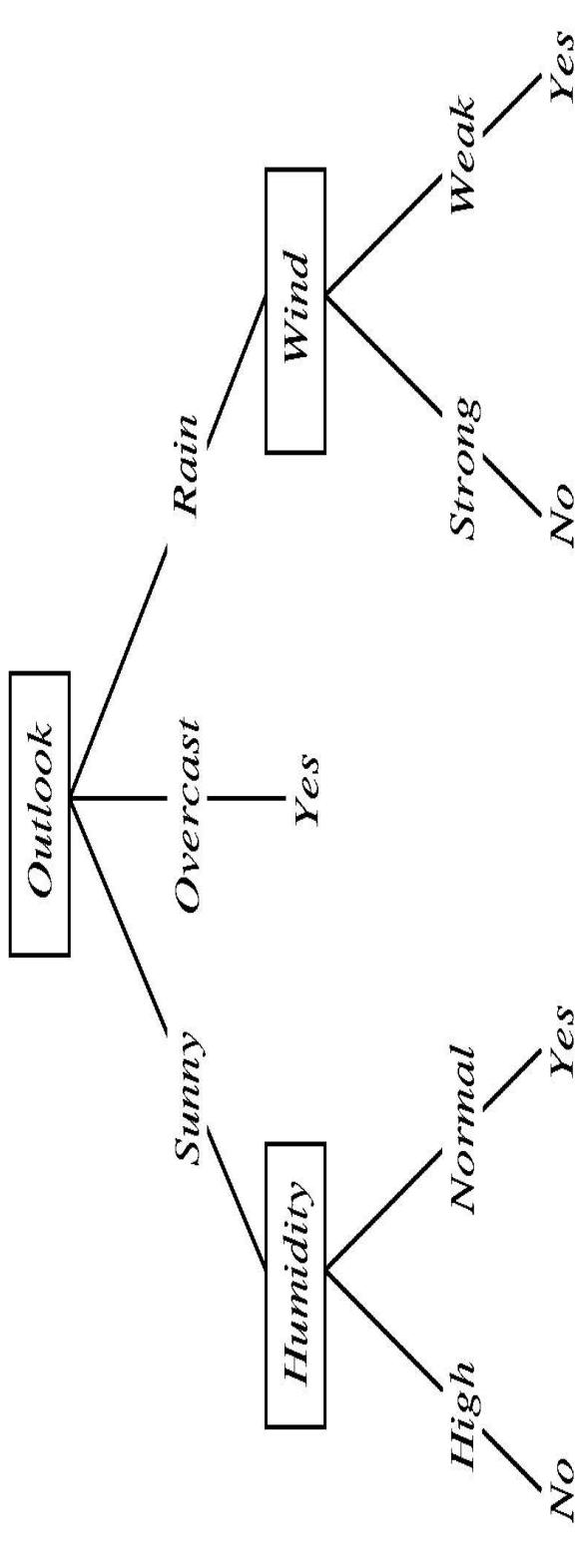
- Naïve Bayes is **one of the fast and easy ML algorithms to predict a class of data** can be used for Binary as well as Multi-class Classifications. It performs well in predictions as compared to the other Algorithms. It is the most popular choice for classification problems.

Naïve Bayes Classifier



Decision Tree Classification

- A decision tree is a flowchart-like structure in which each internal node represents a **test** on a feature (e.g. whether a coin flip comes up heads or tails), each leaf represents a **class label** (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from leaf represent **classification rules**.



Project On Machine Learning

Future Sales Analysis

Importing Dataset Using Pandas

Using read method of pandas we load the dataset into jupyter notebook.

The screenshot shows a Jupyter Notebook environment with a pink vertical sidebar on the left. The main area contains a code cell and its output.

```
In [2]: import pandas as pd
```

```
In [3]: data=pd.read_csv("future_sales_analysis.csv")
```

Out[3]:

Index	TV	Radio	Newspaper	Sales
0	1	230.1	37.8	69.2
1	2	44.5	39.3	45.1
2	3	177.2	45.9	69.3
3	4	151.5	41.3	58.5
4	5	180.8	10.8	58.4
...
195	196	38.2	3.7	13.8
196	197	94.2	4.9	8.1
197	198	177.0	9.3	6.4
198	199	283.6	42.0	66.2
199	200	232.1	8.6	8.7

200 rows × 5 columns

```
In [4]: data=data.drop(['Index'],axis=1)
```

Icons at the bottom of the screen include: WhatsApp, Telegram, File, WhatsApp, YouTube, Gmail, Maps, Login - HackerRank, Welcome to akash..., hyperskill - Google..., Classes, are you ROBOT?, Online Courses - Le..., Interface w...

Data Preprocessing/Data Wrangling

- In this step the acquired data get cleaned by removing the unwanted data and some errors in the data if any.
- This is one of the most important and time consuming step in Data analysis and tidy data.
- This uses most of the python libraries for visualizing the data like matplotlib, seaborn etc.
- It uses drop method to drop any column with the axis method if axis value is 0 then along row the data will get drop and if value is one then vice-versa.

Data Preprocessing Drop Step

```
In [4]: data.drop(['Index'],axis=1)
```

```
Out[4]:
```

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9
...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	9.7
197	177.0	9.3	6.4	12.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	13.4

200 rows × 4 columns

```
In [5]: data.info()
```

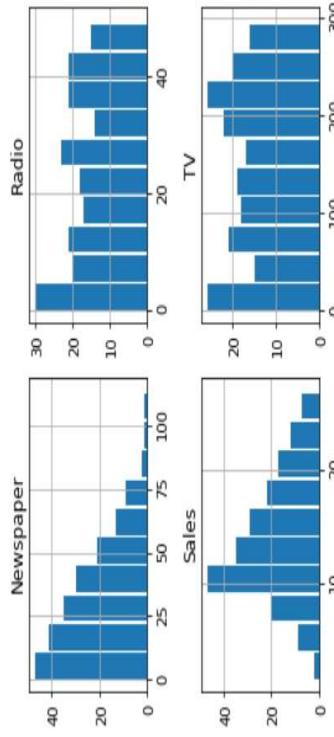
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype  
_____
 0   TV              200 non-null    float64
 1   Radio           200 non-null    float64
 2   Newspaper       200 non-null    float64
 3   Sales           200 non-null    float64
dtypes: float64(4)
memory usage: 7.6 KB
```

90°F
Haze



Data Preprocessing visualizing Step

```
In [9]: data.hist(rwidth=0.9)
plt.tight_layout()
```



```
In [5]: import matplotlib.pyplot as plt
```

```
In [20]: plt.subplot(2,2,1)
plt.title('Newspaper vs sales')
plt.scatter(data['Newspaper'], data['Sales'], s=10, c='r')

plt.subplot(2,2,2)
plt.title('TV vs Sales')
plt.scatter(data['TV'], data['Sales'], s=10, c='g')

plt.subplot(2,2,3)
```



Data Preprocessing Visualizing Step

The screenshot shows a Jupyter Notebook interface with several tabs at the top: WhatsAppApp, Untitled4, Untitled4, File | C:/Users/91891/Downloads/future_sales_analysis.html, Login - HackerRank, Welcome to akgec..., hyperskill - Google..., Classes, are you ROBOT?, Online Courses - Le..., and Interface with ...

In [5]:

```
import matplotlib.pyplot as plt
```

In [20]:

```
plt.subplot(2, 2, 1)
plt.title('Newspaper vs Sales')
plt.scatter(data['Newspaper'], data['Sales'], s=10, c='r')

plt.subplot(2, 2, 2)
plt.title('TV vs Sales')
plt.scatter(data['TV'], data['Sales'], s=10, c='g')

plt.subplot(2, 2, 3)
plt.title('Radio vs Sales')
plt.scatter(data['Radio'], data['Sales'], s=10, c='b')

plt.tight_layout()
```

The notebook displays three scatter plots side-by-side:

- Top-left plot: "Newspaper vs sales" with red dots.
- Top-right plot: "TV vs Sales" with green dots, showing a strong negative correlation.
- Bottom-center plot: "Radio vs Sales" with blue dots.

The bottom of the screen shows a taskbar with various icons, including a search bar, a magnifying glass, a mail icon, a file icon, a settings icon, and system status indicators like battery level and signal strength.

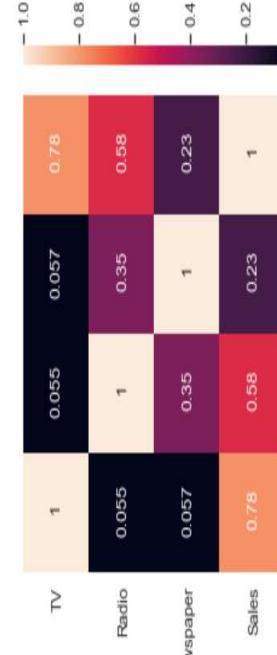
Data Preprocessing Visualizing Step

```
WhatsApp          ✕      Untitled4          ✕      ↵ → ⌂ ⌂ File | C:/Users/91891/Downloads/future_sales_analysis.html
Gmail            YouTube          Maps          ↵ Login - HackerRank          ↵ Welcome to akgec...
In [21]: import seaborn as sns
sns.set()

In [22]: correlation = data.corr()

Out[22]:
      TV      Radio      Newspaper      Sales
TV    1.000000  0.054809  0.056648  0.782224
Radio  0.054809  1.000000  0.354104  0.576223
Newspaper  0.056648  0.354104  1.000000  0.228299
Sales   0.782224  0.576223  0.228299  1.000000

In [24]: sns.heatmap(correlation, annot=True)
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1cfcc04b5788>
```



ENG IN

90°F Haze

Applying Machine learning Algorithms

- After getting the tidy data we have to build the model by ML algorithms based on the need of the data analysis.
- In this project we have used linear regression ML algorithm to analyse the future sales. We have two test sets X and Y.

The screenshot shows a Jupyter Notebook interface with several tabs at the top: WhatsApp, File (Untitled4), hyperskill - Google..., Classes, are you ROBOT?, Online Courses - Le..., Interface with a sto..., and a search bar. The main area contains the following Python code:

```
In [28]:  
y=data['Sales']  
x=data.drop(['Sales'],axis=1)  
print(x)  
print(y)
```

The output shows the data frames:

	TV	Radio	Newspaper
0	230.1	37.8	69.2
1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4
..
195	38.2	3.7	13.8
196	94.2	4.9	8.1
197	177.0	9.3	6.4
198	283.6	42.0	66.2
199	232.1	8.6	8.7

[200 rows x 3 columns]

	Sales
0	22.1
1	10.4
2	9.3
3	18.5
4	12.9
..	..
195	7.6
196	9.7
197	12.8
198	25.5
199	13.4

Name: Sales, Length: 200, dtype: float64

In [29]: `from sklearn.model_selection import train_test_split`

90°F Haze

Applying Linear Regression

- After having training and testing sets we apply linear regression algorithm two test sets by importing sklearn and its sub-module model_selection.

```
WhatsApp          X Untitled4
← → C File | C:/Users/91891/Downloads/future-sales-analysis.html
Gmail YouTube Maps Login - HackerRank Welcome to akiec... hyperskill - Google...
Name:   , Length:  200, dtype:  float64

In [29]: from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=1234)

In [30]: from sklearn.linear_model import LinearRegression

In [31]: model=LinearRegression()

In [32]: model.fit(X_train,Y_train)

Out[33]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

In [34]: y_pred=model.predict(X_test)

In [35]: #score of the model
model.score(X_test,Y_test)

Out[36]: 0.9073615858587188

In [37]: #coefficient of the Line
coefficient=model.coef_
coefficient

Out[37]: array([0.04566079, 0.18927341, 0.00237545])

In [38]: #intercept of the Line
intercept=model.intercept_
intercept

Out[38]: 2.8496682973458007
```

90°F
Haze

ENG IN

Calculating Error Using mean Square

```
In [36]: #score of the model  
model.score(X_test,Y_test)  
  
Out[36]: 0.90736158587188  
  
In [37]: #coefficient of the Line  
coefficient=model.coef_  
coefficient  
  
Out[37]: array([0.04560079, 0.18927341, 0.00237545])  
  
In [38]: #intercept of the Line  
intercept=model.intercept_  
intercept  
  
Out[38]: 2.8496682973458007  
  
In [39]: #equation of the Line  
#y=2.85+0.05TV+0.19Radio+0.002Newspaper  
  
In [40]: #error in the model  
from sklearn.metrics import mean_squared_error  
import math  
  
In [42]: rmse=math.sqrt(mean_squared_error(Y_test,y_pred))  
rmse  
  
Out[42]: 1.7046674275720533
```

In []:



Conclusion

- We have seen the score of the model is 90% and Error is 1.7 which is good so our model is good and it predicts almost accurate value.