Phase 3

Development Part 1

In this part you will begin building your project by loading and

pre-processing the dataset **Import**

Necessary libraries with dataset:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
data = pd.read_csv("E:\IBM\Sales.csv")
```

Exploring data analysis:

Explore the dataset to get an understanding of its structure and characteristics. This data is then subjected to thorough analysis, which includes identifying patterns, trends, and correlations. Fundamental analysis factors, such as company financials, news sentiment, and economic indicators, are often integrated into the dataset. Machine learning algorithms, like regression models and neural networks, can be used to develop predictive models

```
print(data.head())
print(data.info())
print(data.describe())
```

Data pre-processing:

Data pre-processing is a critical step that involves handling missing values, encoding categorical variables, and scaling/normalizing numerical features. Time-series data is typically resampled or aggregated to match the desired prediction horizon, which could be daily, hourly, or even minute-by-minute. This ensures that the data aligns with the modelling objectives.

Feature engineering is another critical aspect. New features may be created from the existing data, like moving averages, relative strength indicators, or other technical indicators, to provide the model with more meaningful input.

```
data = data.dropna()
data = pd.get_dummies(data, columns=['categorical_column'])
```

Splitting the dataset:

It involves machine learning, split your dataset into training and testing sets. This is crucial for model evaluation. splitting the dataset in stock price prediction is fundamental to building reliable and robust models, enabling proper training, tuning, and evaluation of the predictive algorithms while guarding against potential data leakage.

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
data['numerical_column'] = scaler.fit_transform(data['numerical_column'].values.reshape(-1, 1))
X = data.drop('target_column', axis=1)
y = data['target_column']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

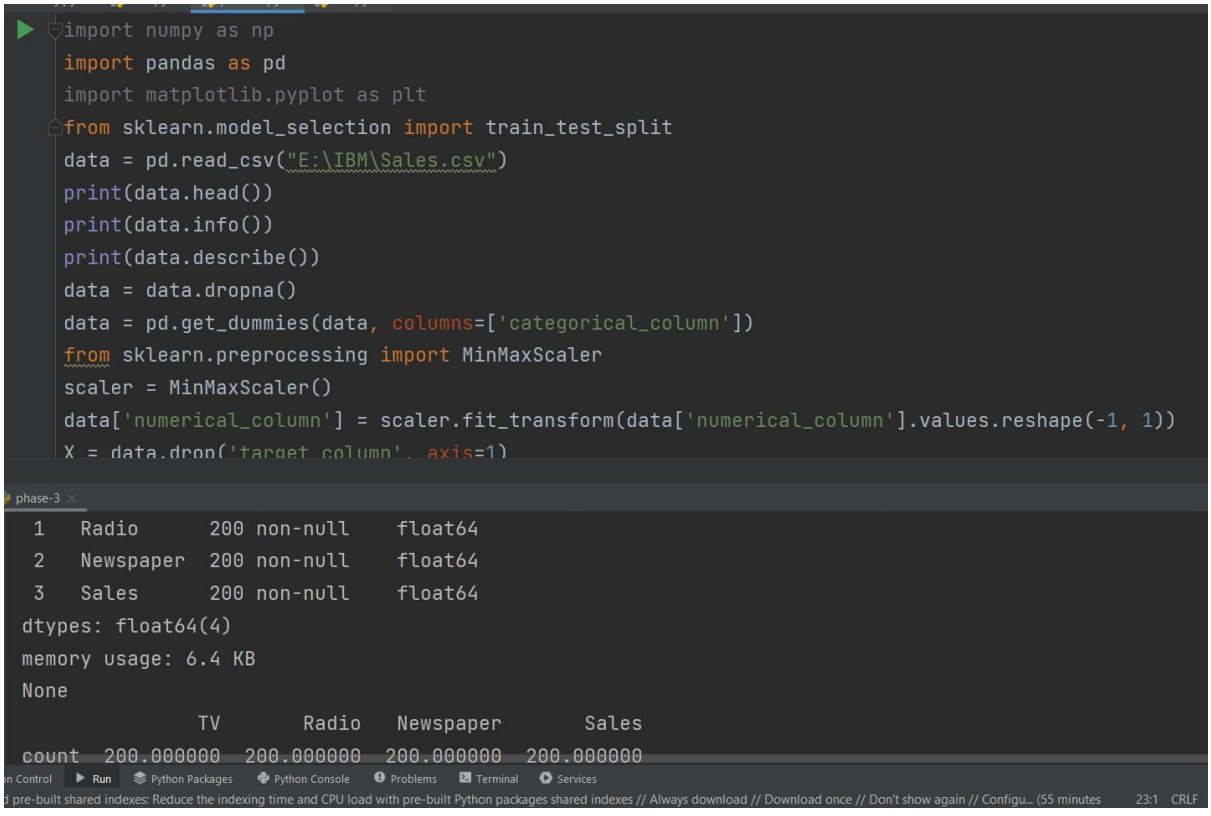
Building and training the models:

Depending on the project, we might build and train machine learning models using libraries like "**Scikit-Learn or deep learning frameworks like TensorFlow or PyTorch**".

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)
from sklearn.metrics import accuracy_score
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

After training the model, evaluate its performance using appropriate metrics

Output:



Team Leader: SK.Nawab Sohail

Team Member: MOHAMMED Madhar

Team Member: MOHAMMED Abdul Raheem

Team Member: SK.Imran