# STOCK PRICE PREDICTION

## TEAM NO. 9

### Building a stock price prediction model involves several steps, including feature engineering, model training, and evaluation. Below, I'll outline each step in more detail and provide a Python code for each step.

### 1. Feature Engineering:

Feature engineering is the process of selecting, creating, and transforming relevant features from our data to improve the model's performance. In the context of stock price prediction, common features might include historical prices, trading volume, technical indicators, and sentiment analysis data.

import pandas as pd

import numpy as np

import yfinance as yf

from ta import add\_all\_ta\_features

# Define the stock symbol and download historical data

stock\_symbol = "MSFT"

start\_date = "13-03-1986"

end\_date = "07-01-2020"

data = yf.download(stock\_symbol, start=start\_date, end=end\_date)

# Calculate technical indicators

data = add\_all\_ta\_features(data, colprefix="ta\_")

# Create additional features if needed

data["Daily\_Return"] = data["Adj Close"].pct\_change()

data["Log\_Return"] = np.log(1 + data["Daily\_Return"])

# Drop missing values

data.dropna(inplace=True)

# Select relevant features

features = data[["ta\_volume", "ta\_rsi", "ta\_macd", "Daily\_Return", "Log\_Return"]]

# Define the target variable

data["Target"] = data["Close"].shift(-1)

data.dropna(inplace=True)

target = data["Target"]

### 2. Model Training:

Next, we need to choose a machine learning or deep learning model to train on our data. In this I'll use a simple linear regression model from scikit-learn. More sophisticated models like LSTM or ARIMA can be used as well:

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.2, random\_state=42)

# Initialize and train the model

model = LinearRegression()

model.fit(X\_train, y\_train)

### 3. Evaluation:

Evaluation is critical to assess our model's performance. Common metrics for stock price prediction include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Here's how we can evaluate the model using RMSE:

from sklearn.metrics import mean\_squared\_error

import math

# Make predictions

y\_pred = model.predict(X\_test)

# Calculate RMSE

rmse = math.sqrt(mean\_squared\_error(y\_test, y\_pred))

print("Root Mean Squared Error:", rmse)

This is a simplified one and real-world applications can be much more complex, involving more sophisticated models, hyperparameter tuning, and extensive feature engineering. It's also important to consider additional factors such as transaction costs, market dynamics, and data quality.

The stock price prediction is inherently uncertain, and models may not perform well in all market conditions. Therefore, it's crucial to exercise caution and consider using these predictions as one input among others in investment decisions.