Project Title

"Predicting Stock Market Trends of S&P 500 using Machine Learning"

Goal of the Project

To predict whether the **stock market will go up or down tomorrow** using machine learning.
We used real stock data and trained a model to help understand market trends.

Tools and Libraries Used

- yfinance – for downloading stock data
- pandas – for handling and cleaning data
- matplotlib – to create graphs
- sklearn – to apply machine learning (Random Forest model)

Data Used

- Data Source: **Yahoo Finance**
- Ticker: ^GSPC (S&P 500 index)
- Time Period: Full history, used data from **1990 onwards**

What is the S&P 500?

The **S&P 500** (short for **Standard & Poor's 500**) is:

A **stock market index** that shows how **500 large companies** listed in the **U.S. stock market** are performing.

These companies are from different industries like:

- Technology (e.g., Apple, Microsoft)
- Health (e.g., Pfizer)
- Finance (e.g., JPMorgan)
- Consumer goods (e.g., Coca-Cola)

Step-by-Step Process

1. Download the Data

Used yfinance to get the S&P 500 historical data.

python
CopyEdit

```
import yfinance as yf
sp500 = yf.Ticker("^GSPC").history(period="max")
```

2. Clean the Data

Removed extra columns: Dividends, Stock Splits

Created a new column:

"Tomorrow" = next day's closing price

"Target" = 1 if price goes up, 0 if goes down

3. Train the Model

Used a **Random Forest Classifier**

Selected columns: Close, Open, High, Low, Volume

Split the data into:

Training: Most of the data

Testing: Last 100 rows

python
CopyEdit

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, min_samples_split=100,
random_state=1)
model.fit(train[predictors], train["Target"])
```

4. Make Predictions

Used the model to predict the test data

Measured how accurate it is using precision_score

5. Improved the Model

Added rolling average and trend values for:

2 days, 5 days, 60 days, 250 days, 1000 days

Example new columns:

Close_Ratio_5

Trend_5

6. Backtesting

Created a function to test the model over time (not just once).

Checked if the model gives consistent results.

Results

Model gives good accuracy.

Able to predict whether the market will rise or fall the next day.

Helpful in learning how stock market behaves.