Minor project

Aim:

To write a code for stock price prediction to create a effective prediction systems for the stock market help traders, investors, and analyst by providing supportive information like the future direction of the stock market.

Code:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import tensorflow as tf

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense

data = pd.read\_csv(r'C:\Users\yogesh\Downloads\yostock4.csv')

print(data.columns)

data.rename(columns={'Unnamed: 0': 'Date'}, inplace=True)

data['Date'] = pd.to\_datetime(data['Date'])

data = data.dropna() # Drop missing values

data['MA\_20'] = data['Close'].rolling(window=20).mean() # Create 20-day moving average

data['Target'] = data['Close'].shift(-1) # Target is the next day's closing price

data = data.dropna() # Drop rows with NaN values after shift

plt.figure(figsize=(10, 5))

plt.plot(data['Date'], data['Close'], label='Close Price')

plt.plot(data['Date'], data['MA\_20'], label='20-Day MA', color='orange')

plt.title('Stock Closing Price with 20-Day Moving Average')

plt.legend()

plt.show()

X = data[['MA\_20']]

y = data['Target']

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

lr\_model = LinearRegression()

lr\_model.fit(X\_train, y\_train)

y\_pred = lr\_model.predict(X\_test)

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

print(f'Linear Regression RMSE: {rmse\_lr}')

X\_train\_lstm = np.array(X\_train).reshape((X\_train.shape[0], 1, X\_train.shape[1]))

X\_test\_lstm = np.array(X\_test).reshape((X\_test.shape[0], 1, X\_test.shape[1]))

lstm\_model = Sequential()

lstm\_model.add(LSTM(50, return\_sequences=False, input\_shape=(X\_train\_lstm.shape[1], X\_train\_lstm.shape[2])))

lstm\_model.add(Dense(1))

lstm\_model.compile(optimizer='adam', loss='mse')

history = lstm\_model.fit(X\_train\_lstm, y\_train, epochs=10, batch\_size=32, validation\_data=(X\_test\_lstm, y\_test))

y\_pred\_lstm = lstm\_model.predict(X\_test\_lstm)

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

print(f'LSTM RMSE: {rmse\_lstm}')

plt.figure(figsize=(10, 5))

plt.plot(y\_test.values, label='Actual Price', color='blue')

plt.plot(y\_pred, label='Predicted Price (LR)', color='red', linestyle='dashed')

plt.plot(y\_pred\_lstm, label='Predicted Price (LSTM)', color='green', linestyle='dotted')

plt.title('Actual vs Predicted Prices')

plt.legend()

plt.show()

Output:



