**NEXT DAY POSITION CLASSIFICATION PROJECT**

**DOCUMENTATION**

**INTRODUCTION**

The aim of this project is to take data of a security at the end of the current day and predict whether the security should be bought at the open and sold at the close of the next day. It involves using both regression and classification to predict the return of the next day and classify whether the security should be bought or not.

**LIBRARIES INCLUDED**

1. SCIKIT-LEARN- The scikit-learn library is a machine learning library of python, providing numerous features for supervised machine learning. Several modules and functions of the sklearn library have been used in the project:

* sklearn. linear\_model- This is used to import the machine learning model- Linear Regression.
* sklearn. pipeline- This module contains the Pipeline() function which allow the main events i.e., Preprocessing of data, and initialization of the model, to take place one after another.
* sklearn.metrics- This module allows to import functions for model performance metrics. Mean squared error function has been imported from this module.
* sklearn.ensemble – This module contains functions that help enable ensemble techniques. It has been used for importing the Random Forest Regressor and Random Forest Classifier model.
* sklearn.preprocessing- This module provides functions for preprocessing data. Using this module, the following functions have been imported:

1. StandardScaler- This function allows to scale the data such that mean is 0 and standard deviation is 1.

* Sklearn.model\_selection- This module is used to import two important functions:

1. Train\_test\_split- this allows us to divide the dataset into training and testing data.
2. GridSearchCV- this function allows to find the best model of a hyperparameter, along with cross validation.
3. JOBLIB- The joblib library has been used to download and re-use the machine learning models created for each time the code is run.
4. NUMPY – This is the python library for performing numerical computations.
5. PANDAS – This is the python library used for performing operations related to data.
6. YFINANCE- This library has been used to access OHLCV data from yahoo finance.
7. TALIB – This library provides numerous functions to perform technical analysis of stocks. It provides functions for various indicators, oscillators, and candlestick patterns.
8. DATETIME – This library is used for working with and manipulating dates and times.
9. DATEUTIL – This library is an extension of the datetime library and extends its functionalities.

**FUNCTIONS**

1. model

This is the core function of the project. It asks the user to enter the ticker of the security, generates OHLCV data for each day for the past year from the current day and creates two dataframes- one for regression and one for classification, by calling the dataprepare function. The dataframe used for regression is then sent to the linear regression and random forest regression models where they are trained to predict the return of the next day, and the classification dataframe is sent to the random forest classifier model where it learns how to classify whether a stock should be bought or not. These models are then downloaded. Finally, OHLCV data, as well as the Average True Range (ATR), Exponential Moving Average (EMA), and Chande Momentum Oscillator (CMO) value at the end of the day is taken from the user, and either a ‘Buy’ or ‘Don’t Buy’ message is returned by the function.

1. dataprepare

The dataprepare function is used to prepare data for the regression and classification models. It takes the ticker, start date, and end date, all of which have been taken from the model function, and generates OHLCV data. It then calculates the CMO,ATR, and EMA values at the end of each day and buying return for the end of the next day. This dataframe is the one being given to the regression models with all columns except the ‘tomorrow\_buying\_return’ column as features, and the latter as target. A copy of this dataframe is made, and according to the returns, a position column is added, and this dataframe is the one being given to the classification models, with the position as target, and other columns as features.

1. linreg

The linreg function is used to train a linear regression model on the data received. It takes two arguments- features, and targets. It then divides the data into training and testing data, then creates a pipeline for creating polynomial features, standardizing the data, and instantiating the linear regression model. This model is then tested on the testing set. The function returns the model.

1. forestreg

The forestreg function is used to train a Random Forest Regressor model on the data received. It takes two arguments- features, and targets. It then divides the data into training and testing data, then creates a pipeline for creating polynomial features, standardizing the data, and instantiating the linear regression model. After this, optimal value of hyperparameter i.e., number of estimators is found and cross validation is done using the GridSearchCV, and the best model is returned. This model is then tested on the testing set and is saved. The function returns this best model.

1. **forestclas**

The forestclas function is used to train a random forest classifier model on the data received. It takes two arguments- features, and targets. It then divides the data into training and testing data, then creates a pipeline for creating polynomial features, standardizing the data, and instantiating the linear regression model. After this, optimal value of hyperparameter i.e., number of estimators is found and cross validation is done using the GridSearchCV, and the best model is returned. This model is then tested on the testing set and is saved. The function returns this best model.

**WORKING**

As the code runs, the model function is called and the user is prompted to enter the ticker symbol of a security, and the current date. The date of the previous day and one year before the previous day is calculated, and all this is sent as arguments to the dataprepare function. Here, two dataframes- one for regression and one for classification are prepared.

The linear regression and random forest classification models are trained to calculate the return of the next day, while the random forest classification model is used to classify whether a security should be bought at the open of the next day and sold at the close, or not. Once trained and tested, these models are saved using the joblib library.

Finally, the user is prompted to enter data such as the OHLCV data, as well as ATR, EMA, and CMO at the end of the day. Using this data, the regression model predicts the return of the next day. All this data (including the return calculated by the regression model) is used by the classification model and a ‘Buy’ or ‘Don’t Buy’ message is returned by the function.

**LIMITATIONS**

1. USER INPUT HANDLING – The code depends on the input of the user and could crash or produce wrong results if the user enters inaccurate or invalid data.
2. LACK OF MODEL EVALUATION – No metrics for evaluating the performance of the regression or classification models have been used. In addition, only one model for classification i.e., Random Forest Classifier has been used.