**CLOSE PRICE PREDICTION PROJECT**

**DOCUMENTATION**

**INTRODUCTION**

The project is a linear regression model that aims to predict the close price of a security using OHLCV data for each day. A trader, using the OHLCV data of the previous day, and open price of the current day, can predict the close price of a security, and accordingly place trades. The project uses three Regression Models (Linear Regression, Ridge/Lasso Regression, and Random Forest Regressor) to make the prediction.

**LIBRARIES INCLUDED**

1. SCIKIT-LEARN- The scikit-learn library is a machine learning library of python, providing numerous features for supervised and unsupervised 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 models- Linear Regression, Ridge Regression, and Lasso Regression.
* sklearn.pipeline- This module contains the Pipeline() function which allows 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. Since the project is using regression models, mean\_squared\_error function has been used to find the error for each model.
* sklearn.ensemble – This module contains functions that help enable ensemble techniques. It has been used for importing the Random Forest Regressor 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.
2. PolynomialFeatures- This function allows to create new features, using existing features.

* Sklearn.model\_selection- This module is used to import three 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 using a grid search, 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.

**FUNCTIONS**

1. predictor

This is the core function of the project. It takes the ticker of the security, whose price must be predicted, as an argument. After this, using the yahoo finance library, it generates OHLCV data for the security, from 2022 to 2024. The data is then converted to the proper format to avoid look ahead bias and is divided into features and targets. The features (x) contain - 'Open\_prev' (opening price of the previous day),'High\_prev' (highest price of the previous day), 'Low\_prev' (lowest price of the previous day), 'Close\_prev' (close price of the previous day), 'Volume\_prev' (volume of the previous day),'Open\_current’ (opening price of the current day), and the target (y) contains ‘Close’, which is the close price of that particular day.

The predictor function then trains the 3 regression models on this data and chooses two models which have the best performance for the strategy, by comparing the error of each model. Finally, data including the previous day’s OHLCV data, and the current day’s opening price is taken from the user, and a prediction regarding the close price is returned.

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. Finally, it trains the model, tests it on the testing set, and stores the model as well as its error. The function returns the model, as well as the error.

1. lasreg

The lasreg function is used to train a lasso 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 lasso regression model. After this, optimal value of hyperparameter 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 and the error is calculated. The function returns this best model, as well as the error.

1. ridgereg

The ridgereg function is used to train a ridge 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. After this, optimal value of hyperparameter 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 and the error is calculated. The function returns this best model, as well as the error.

1. forest

The forest 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, 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 and the error is calculated. The function returns this best model, as well as the error.

**WORKING**

This model would most suitably used, just after the opening price of the current day is available, to predict the close price, and hence decide whether the security should be bought and sold at the end of the day.

The user enters the ticker of the security, whose price must be predicted. This is passed as an argument to the predictor function, which gets data for the security, prepares it, and sends it to the linreg, lasreg, ridgereg, and forest functions, which are used to create models for linear regression, ridge regression, lasso regression, and random forest regressor.

After these models have been created, the predictor function compares the error generated by each of the linear, ridge, and lasso regression model, and chooses the one that gives the least error, along with the random forest model, to make the prediction. Finally, once the models are downloaded and ready for use, previous day’s OHLCV data, as well as the current day’s opening price, is taken from the user, and the close price of the current day is predicted and returned.

**LIMITATIONS**

1. USER INPUT HANDLING – The code depends on the input of the user and could crash if an invalid ticker symbol is entered.
2. MODEL PERFORMANCE EVALUATION – Only mean squared error is used as a metric to evaluate model performance. In addition, Mean Absolute Error and R-squared error could also have been used.
3. FEATURE ENGINEERING – Current features only include previous day’s financial data. Performance of the model can improve if additional factors such as news, sentiment, market condition etc. is considered.