High Level Design (HLD)

INVESTMENT PREDICTION

Document Version Control

Date Issued	Version	Description	Author
05/04/2023	1	Initial HLD – V1.0	Mohit Chatterjee

Contents

Abstract

In the era of big data for predicting stock market prices and trends has become even more popular than before. I collected 11 years of GOOGLE's stock data and proposed a comprehensive customization of feature engineering and machine learning-based model for predicting price trend of stock markets. The proposed solution is comprehensive as it includes pre-processing of the stock market dataset, utilization of multiple feature engineering techniques, combined with a customized machine learning based system for stock market price trend prediction. We conducted comprehensive evaluations on frequently used machine learning models and conclude that our proposed solution outperforms due to the comprehensive feature engineering that we built. The system achieves overall high accuracy for stock market trend prediction. With the detailed design and evaluation of prediction term lengths, feature engineering, and data pre-processing methods, this work contributes to the stock analysis research community both in the financial and technical domains.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

1.3 Definitions

Term	*Description*
Individual	Investment Prediction
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2 General Description

2.1 Product Perspective

An Investment Prediction Model is a tool that aims to help investors make informed decisions about which investments to choose based on their expected returns. The model uses historical financial data, economic indicators, and other relevant information to predict future returns for various investments.

2.2 Problem statement

The problem statement of an Investment Prediction Model typically involves several key factors, including:

- Data sources: The model must identify the relevant data sources that will be used to train the model, such as historical financial data, economic indicators, and market trends.
- Prediction horizon: The model must determine the time horizon over which it will make predictions. This could be short-term (days or weeks), medium-term (months or quarters), or long-term (years).
- Accuracy: The model must achieve a high level of accuracy in its predictions to be useful for investors. This may involve developing a model that can account for various sources of uncertainty and variability in the data.
- Interpretability: The model must be interpretable, meaning that investors can understand how the model makes its predictions and the factors that influence those predictions. This can help investors to trust the model and use it more effectively.

2.3 PROPOSED SOLUTION

The solution proposed here an investment Prediction Model is to develop a reliable and accurate tool for predicting investment returns, which can help investors make informed decisions about their investments and manage their portfolios more effectively.

2.4 FURTHER IMPROVEMENTS

Need more data to predict more accurately and need UI which have a feature to upload company's dataset and our model can predict the price by doing all feature engineering part.

2.5 Tools used

Python programming language and **MongoDB** is used to retrieve, insert, delete, and update the database, **VSCode** is used as IDE. **GitHub** is used as version control system.

2.6 Constraints

An Investment Prediction Model is to develop a reliable and accurate tool for predicting investment returns, which can help investors make informed decisions about their investments and manage their portfolios more effectively.

2.7 Assumptions

The aim of making an Investment Prediction model is to contain certain assumptions that help the users analyze the financial performance of the company. The assumptions are a crucial component of the financial models and are necessary for the users to understand the financial performance of a company.

2.8 Event log

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

2.9 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3 Performance

An investment Prediction Model is to develop a reliable and accurate tool for predicting investment returns, which can help investors make informed decisions about their investments and manage their portfolios more effectively.

3.1 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.2 Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

3.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

3.4 Deployment

3.5 KPIs (Key Performance Indicators)

1. Time and workload reduction using the Investment Prediction.

4 Conclusion

we are predicting the closing stock price of any given organization, we have developed an application for predicting close stock price using LSTM algorithm. We have used datasets belonging to Google, Nifty50, TCS, Infosys and Reliance Stocks and achieved above 93% accuracy for these datasets. In the future, we can extend this application for predicting cryptocurrency trading and also, we can add sentiment analysis for better predictions.

5 References

1. Github : https://github.com/Mohitchatterjee/Investment_Prediction
2. Linkedin : LinkedIn Post

Investment Prediction

Objective:

In the era of big data for predicting stock market prices and trends has become even more popular than before. I collected 11 years of GOOGLE's stock data and proposed a comprehensive customization of feature engineering and machine learning-based model for predicting price trend of stock markets. The proposed solution is comprehensive as it includes pre-processing of the stock market dataset, utilization of multiple feature engineering techniques, combined with a customized machine learning based system for stock market price trend prediction. We conducted comprehensive evaluations on frequently used machine learning models and conclude that our proposed solution outperforms due to the comprehensive feature engineering that we built. The system achieves overall high accuracy for stock market trend prediction. With the detailed design and evaluation of prediction term lengths, feature engineering, and data pre-processing methods, this work contributes to the stock analysis research community both in the financial and technical domains.

Benefits:

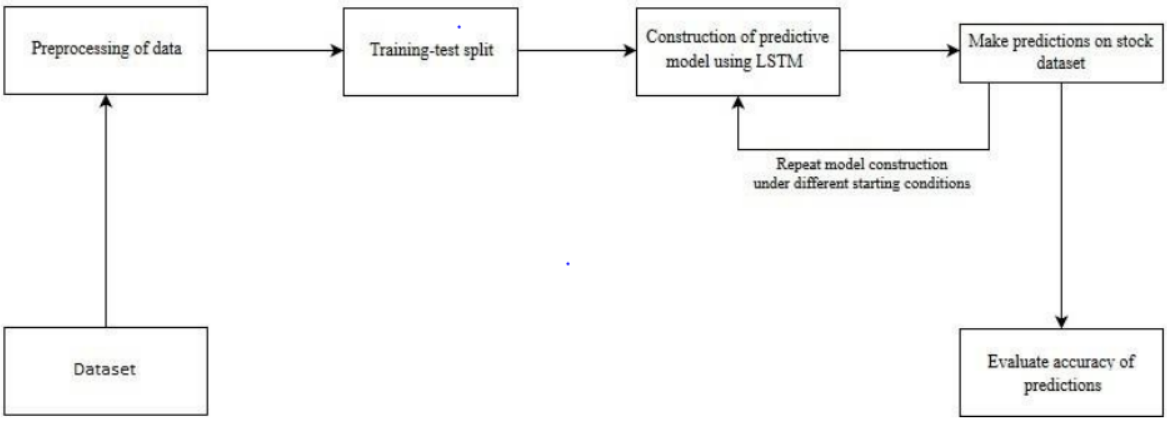
- Predict Stocks Price.
- Gives better insight of companies results.
- Helps in easy flow for managing resources.

Data Sharing Agreement :

- Sample file name (ex fraudDetection_20062021_101010)
- Length of date stamp(8 digits)
- Length of time stamp(6 digits)
- Number of Columns
- Column names
- Column data type

Data Sharing Agreement :

- Sample file name (ex fraudDetection_20062021_101010)
- Length of date stamp(8 digits)
- Length of time stamp(6 digits)
- Number of Columns
- Column names
- Column data type



Model Training:

- Data Export from Db :

The accumulated data from db is exported in csv format for model training

- Data Preprocessing
  - Performing EDA to get insight of data like identifying distribution , outliers ,trend

among data etc.

  - Check for null values in the columns. If present impute the null values.
  - Encode the categorical values with numeric values.
  - Perform Standard Scalar to scale down the values.

- Clustering –
  - KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using KneeLocator function. The idea behind clustering is to implement different algorithms on the structured data
  - The Kmeans model is trained over preprocessed data, and the model is saved for further use in prediction
- Model Selection –

After the clusters are created, we find the best model for each cluster. By using 2 algorithms “SVM” and "XGBoost". For each cluster both the hyper tunned algorithms are used. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction

Prediction:

- The testing files are shared in the batches and we perform the same Validation operations ,data transformation and data insertion on them.
  - The accumulated data from db is exported in csv format for prediction
  - We perform data pre-processing techniques on it.
  - Logistic Regression model created during training is loaded and clusters for the preprocessed data is predicted
  - Once the prediction is done. The predictions are saved in csv format and shared.

Q & A:

Q1) What's the source of data?

The data for training is provided by the client in multiple batches and each batch contain multiple files

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What's the complete flow you followed in this Project?

Refer slide 4th for better Understanding

Q 4) After the File validation what you do with incompatible file or files which didn't pass the validation?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q 5) How logs are managed?

We are using different logs as per the steps that we follow in validation and modeling like File validation log , Data Insertion ,Model Training log , prediction log etc.

Q 6) What techniques were you using for data pre-processing?

- Removing unwanted attributes
- Visualizing relation of independent variables with each other and output

variables

- Checking and changing Distribution of continuous values
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.
- Scaling the data

Q 7) How training was done or what models were used?

- Before diving the data in training and validation set we performed clustering over fit to divide the data into clusters.
- As per cluster the training and validation data were divided.
- The scaling was performed over training and validation data
- Algorithms like SVM , XGBoost were used based on the recall final model was used for each cluster and we saved that model .

Q 8) How Prediction was done?

The testing files are shared by the client .We Perform the same life cycle till the data is clustered .Then on the basis of cluster number model is loaded and perform prediction. In the end we get the accumulated data of predictions.

- Q 9) What are the different stages of deployment?
  - When the model is ready we deploy it in Fire environment .Where SIT and UAT is performed over it.
  - Once We get Sign off from Fire we deploy in Earth and UAT is performed over it.
  - After getting the sign off from Earth we deploy in production