iNeuron

High Level Design
STORES SALES PREDICTION

Revision number: 2.0
Last date of revision:15 – 09 – 2021

Document Version Control

Date Issued	Version	Description	Author
15/09/2021	1	Initial HLD-V1.0	Somay
16/09/2021	2	Final HLD-V2.0	Somay

Contents

Abstract

The purpose of this research is to construct a **sales prediction** model for retail **stores** using the deep learning approach, which has gained significant attention in the rapidly developing field of machine learning in recent years. Using such a model for analysis, an approach to **store** management could be formulated.

1. Introduction

1.1 Why this High-Level Document?

The purpose of this High-Level Design (HLD) Document is to add necessary details to the current project description to represent a suitable model for coding. This model 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 the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance and 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, application flow (Navigations), and technology architecture. The HLD uses non-technical to mildly-technical term which should be understandable to the administrator of the system.

1.3 Definitions

TERM	DESCRIPTION
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

The stores sales prediction is a machine learning based multi classification model which will helpus to predict sales yearly and month wise whether it is primary product is sales is high or not.

2.2 Problem Statement

Nowadays, shopping malls and Big Marts keep track of individual item sales data in order to forecast future client demand and adjust inventory management. In a data warehouse, these data stores hold a significant amount of consumer information and particular item details. By mining the data store from the data warehouse, more anomalies and common patterns can be discovered.

2.3 Proposed Solution

The proposed solution for this project is Machine learning algorithms can be implemented to predict the stores sales. Considering various features like item type, outlet size, item mrp, item year   as inputs from the web app, the implemented classification model will predict the output as store sales

Here, we have used Random Forest Classifier to predict whether the stores sales is high or not.

However, drawing a baseline model is important since it tells us how well other models have performed compared to base model. Here, the base model for Store sales dataset isLinear regression

1. Baseline Model: Linear Regression
2. Actual Model: Random Forest

2.4. Data Requirements

The sales data used for this project, taken from the UCI Machine Learning Repository. The given dataset contains records 5836 and 10 features. Features are distributed as 6 continuous features and 4 categorical features. The given features are:

Name	Description
Item Identifier	Identifier of the item
Item Weight	Weight of item
Item Fat Content	Low Regular
Item Visibility	Visibility of the item
Item Type	Dairy, soft drinks , Fruit , Frozen Food

Outlet Identifier	Identifier of the Outlet
Outlet Year	Year of the outlet (1999, 2001,2003)
Outlet Size	Medium, High, Low
Outlet Location	Tier1, Tier2, Tier3
Outlet Type	Super Market 1 Super Market 2, Grocery stores
Item Sales	Sales Of the Item

2.5. Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask, Heroku, Git.
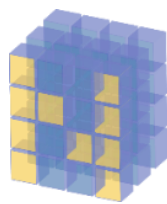
- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- Heroku is used for deployment of the model.
- MongoDB is used to retrieve, insert, delete, and update the database
- Frontend development is done using HTML/CSS
- Python Flask is used for backend development.
- GitHub is used as version control system.

2.6 Constraints

The Stores sales prediction application must be user friendly, as automated as possible and users should not be required to know any of the workings.
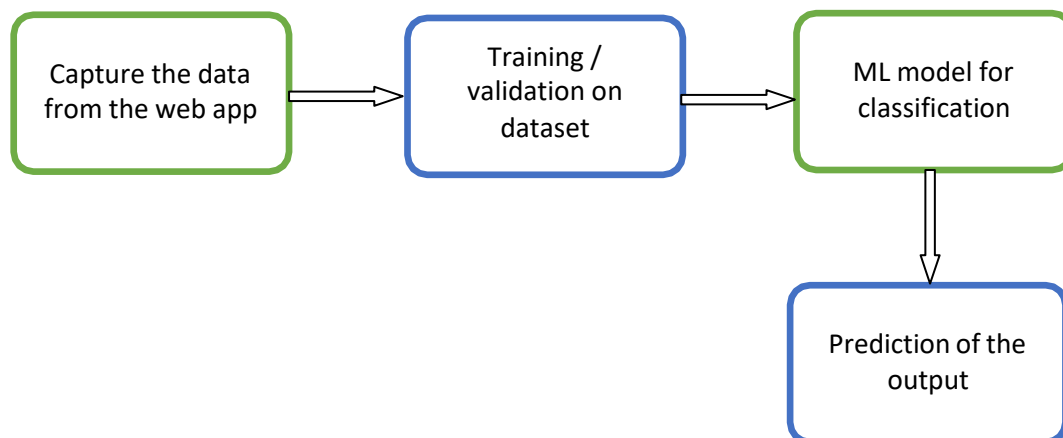
2.7 Assumptions

The main objective of this project is to predict the sales and based on various factors of individuals. A machine learning-based classification model has been used for predicting the use case based on the user input.It is also assumed that all aspects of this project have the ability to work together as the designers is expecting.
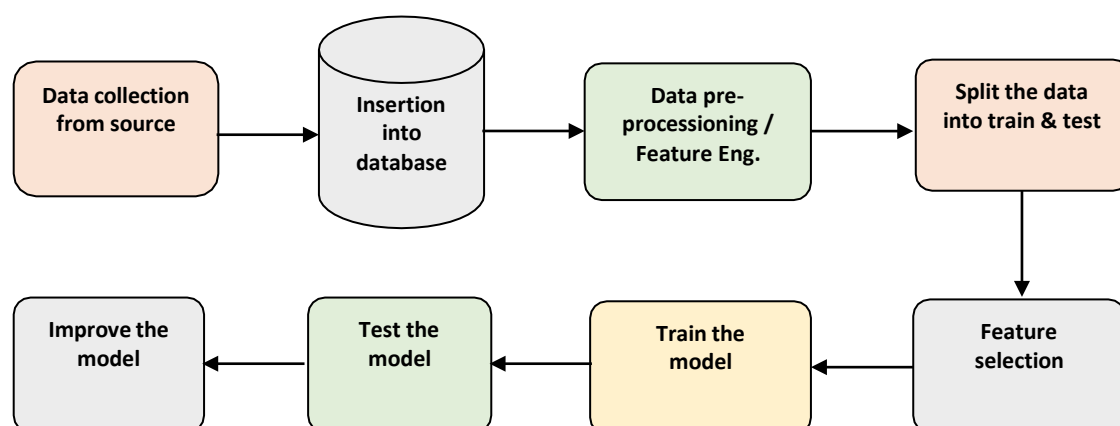
3. Design Details

3.1 Process Flow

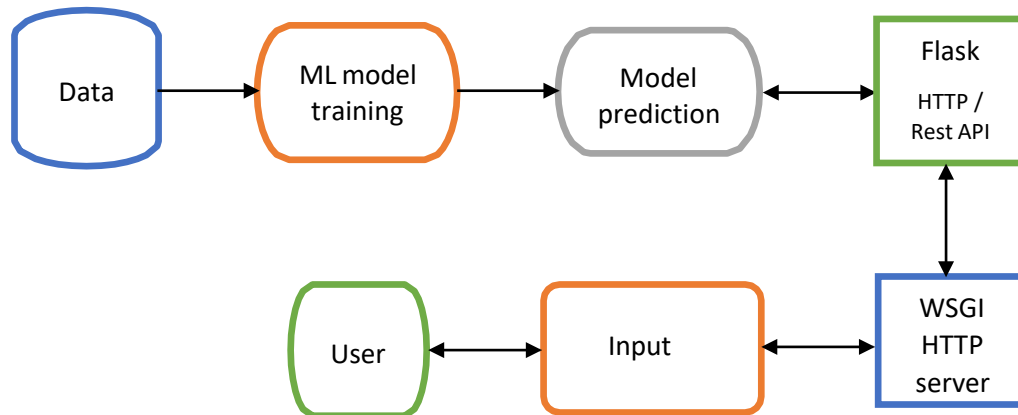For predicting the store sales, we will use classification model. Below is the process flow diagram is as shown below.

Proposed Methodology

3.1.1 Model Training and Evaluation

3.1.2 Deployment Process

```
Data → ML model training → Model prediction ↔ Flask HTTP / Rest API
                                                    ↕
User ↔ Input ↔ WSGI HTTP server
```

3.2 Event Log

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

Initial Step-By-Step Description:

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

3.3 Error Handling

Should error 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

4. Performance

We can observe that the accuracy of the predicted output was seen at 80% using Random Forest classifier. Other classification models such as logistic regression and decision tree have given good accuracy above 80

4.1 Reusability

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

4.2 Application Compatibility

The different components for this project will be using 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.

4.3 Resource Utilization

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

4.4 Deployment

Deployment and IDE

The Final Deployment or web page is deployed by flask and html Framework

5. Conclusion

In this project, about two machine learning classification models are evaluated for the given dataset to predict the stores. The stores sales data was used to develop the two classification models, and the predicted the output from these models by comparing with actual target to check the accuracies of these models. It has been found that Random Forest model which is built upon linear regression is the best performing the model among them and improved the model further with hyper parameter tuning.

6. References

1. [Classification of Thyroid Disease using Machine Learning](#)
2. [Thyroid Disease: Causes, Symptoms, Risk Factors, Testing & Treatment (clevelandclinic.org)](#)
3. [Thyroid disease - Wikipedia](#)