

**HIGH-LEVEL DOCUMENT
(HLD)**

STORE SALES PREDICTION

Dipendra Singh
Dinansh Bhardwaj

Contents

Abstract.....	3
1. Introduction.....	4
1.1. Why this HLD Document?.....	4
1.2. Scope.....	4
1.3. Definitions.....	5
2. General Description.....	5
2.1. Product Perspective.....	5
2.2. Problem Statement.....	5
2.3. Proposed Solution.....	6
2.4. Further Improvements.....	6
2.5. Data Requirements.....	6
2.6. Tools Used.....	7
2.7. Constraints.....	8
2.8. Assumptions.....	8
3. Design Details.....	8
3.1. Process Flow.....	8
3.1.1. Model Training and Evaluation.....	9
3.1.2. Deployment Process.....	9
3.2. Event Log.....	10
3.3. Error Handling.....	10
3.4. Performance.....	10
3.5. Reusability.....	10
3.6. Application Compatibility.....	10
3.7. Resource Utilization.....	11
3.8. Deployment.....	11
4. Conclusion.....	11
5. References.....	11

Abstract

In this project, we applied machine learning techniques to a realworld problem of predicting stores sales. This kind of prediction enables store managers to create effective staff schedules that increase productivity and motivation. We used the popular open-source statistical programming language Python. We used feature selection, model selection to improve our prediction result. Because of the nature of our problem, the Gradient Boosting technique is used to measure the prediction accuracy.

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 before 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 user interface is implemented
- 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
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
Heroku	Cloud platform as a service supporting several programming languages
ML	Machine Learning is the study of computer algorithms that can improve automatically through experience and by the use of data

2. General Description

2.1. Product Perspective

The store sales prediction model is a supervised learning-based regression model which will help us to predict the sales of the product in the particular store.

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 datastore from the data warehouse, more anomalies and common patterns can be discovered.

2.3. Proposed Solution

The solution proposed here is a supervised learning-based regression model that can be implemented to perform the above-mentioned use cases. To gain better accuracy, we will try various regression algorithms like Linear Regression, Ridge Regression, Lasso Regression, Random Forest, etc.

2.4. Further Improvements

We can tune the model with Hyperparameter tuning. It is choosing the optimal set of parameters so that we can get optimal outcomes from the learning process. To improve the performance of the model, we can do some influential practices like Trade precision with recall, Feature engineering, enriching training data with public datasets, removing data leakage features, etc. Model tuning is a lengthy and repetitive process to test new ideas, retrain the model, evaluate the model, and compare the metrics.

2.5. Data Requirements

Data requirements completely depend on our problem statement. Our model requires data/information of an item and the related store like Item Visibility, Item Fat Content, Item MRP, Outlet Establishment Year, etc.

2.6. Tools Used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Seaborn, Matplotlib, SciPy, pylab, pickle, and Flask are used to build the whole model.



- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib and Seaborn are used.
- Heroku is used for the deployment of the model.
- Cassandra database is used to retrieve, insert, delete, and update the database.
- Front-end development is done using HTML/CSS.
- GitHub is used as a version control system.

2.7. Constraints

The store sales prediction model must be user-friendly, as automated as possible and users should not be required to know any of the workings.

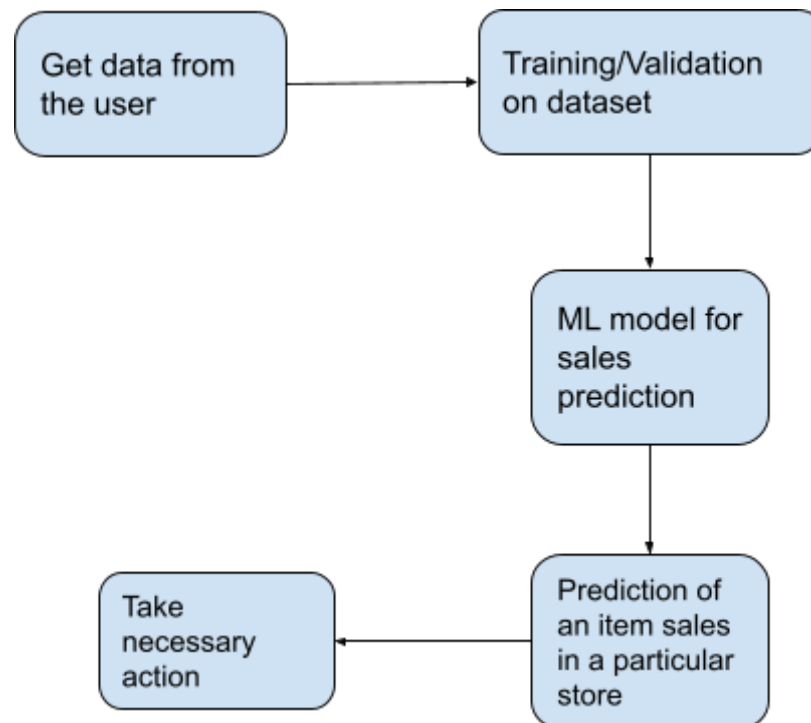
2.8. Assumptions

The main objective of the project is to implement the use case as previously mentioned (2.2 Problem Statement) with the help of different features. The regression-based ML model is used for predicting the above-mentioned use case based on the input data. It is also assumed that all aspects of this project have the ability to work together in the way the store manager or the designer is expecting.

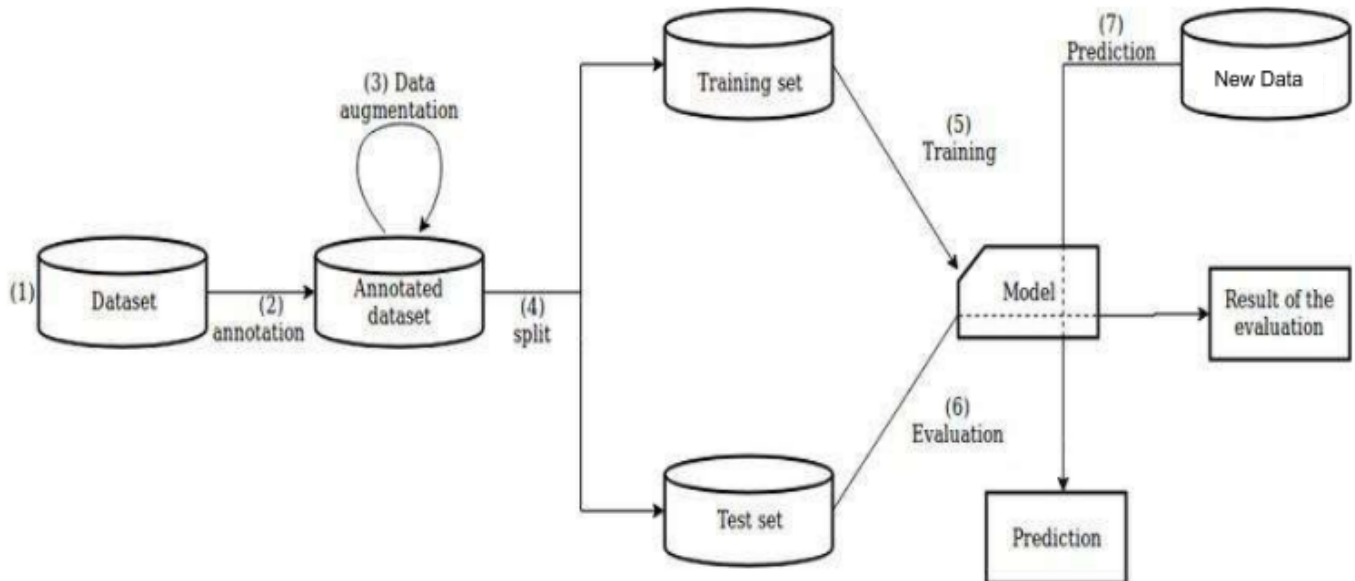
3. Design Details

3.1. Process Flow

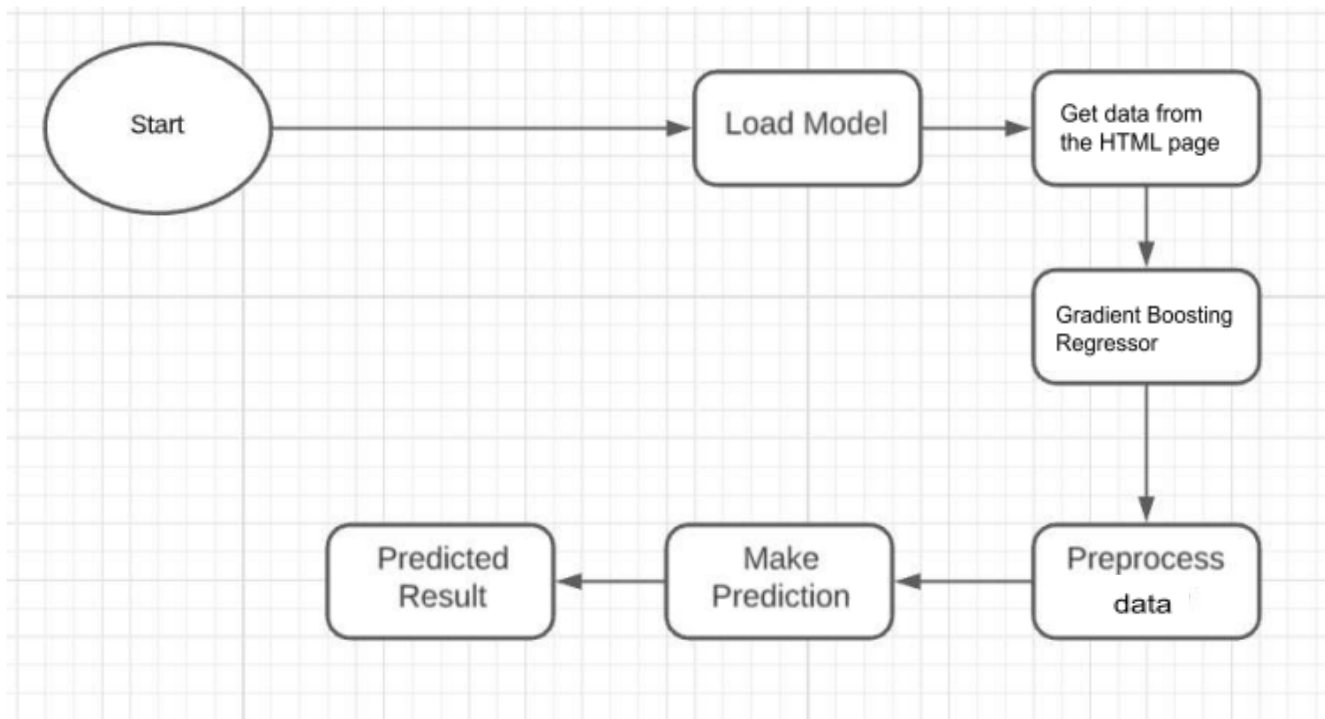
The process flow diagram is shown below:



3.1.1. Model Training and Evaluation



3.1.2. Deployment Process



3.2. 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 is required.
2. The system should be able to log every system flow.
3. The developer can choose the logging method. You can choose Py logging/ file logging.
4. The system should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3. Error Handling

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

3.4. Performance

The regression-based ML solution is used for predicting the sales of an item for a particular store with the help of various features. It will concern authorities and takes necessary action, so it should be as accurate as possible. So that it will not mislead the concerned authority (like store manager, etc). Also, model retraining is very important to improve performance.

3.5. Reusability

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

3.6. Application Compatibility

The different components for this project will be using Python as an interface between them.

3.7. Resource Utilization

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

3.8. Deployment



Heroku is a cloud platform as a service (PaaS) supporting several programming languages.

4. Conclusion

The designed ML model will predict the sales of the product in the particular store based on various key features used to train the regression algorithm. The manager can create effective staff schedules that increase productivity and motivation by using our model so that the store will yield more profit than the previous time.

5. References

1. [Supervised ML](#)
2. [Cassandra Database](#)
3. [Flask framework](#)
4. [Deployment on Heroku cloud](#)
5. [Event Logging](#)
6. [Some images and phrases](#)