



Architecture Design Document

Written_By	Arpitha G
Document Version	1.0
Last Revised Date	25/09/2023

Index

Content	Page No
Abstract	4
1. Introduction	4
1.1 What is Architecture Design?	4
1.2 Scope	4
1.3 Constraints	4
2. Technical Specification	5
2.1 Dataset	5
2.2 Logging	7
2.3 DataBase	7
2.4 Deployment	7
3. Technology Stack	8
4. Proposed Solution	8
5. Architecture	8
5.1 Architecture Description	9
6. Model Training/Validation Workflow	11
7. User Input/Output Workflow	12

Abstract

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization's target.

1. Introduction

1.1 What is Architecture Design?

The goal of Architecture Design (AD) or a low-level design document is to give the internal design of the actual program code . Architecture Design describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Architecture Design(AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

1.3 Constraints

We only predict whether a transaction is fraud or not based on the given information.

2. Technical Specification

2.1 Dataset

We used a synthetic data set Generated using the simulator called PaySim as an approach to such a problem . PaySim uses aggregated data from the private data set to generate a synthetic data set that resembles the normal operation of transactions and injects malicious behaviour to later evaluate the performance of fraud detection methods.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

The data set consists of various data types from integer to floating to object as shown in Fig.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 11 columns):
#   Column              Dtype
---  -
0   step                int64
1   type                object
2   amount              float64
3   nameOrig            object
4   oldbalanceOrg       float64
5   newbalanceOrig      float64
6   nameDest            object
7   oldbalanceDest      float64
8   newbalanceDest      float64
9   isFraud             int64
10  isFlaggedFraud      int64
dtypes: float64(5), int64(3), object(3)
memory usage: 534.0+ MB
```

In the raw data, there can be various types of underlying patterns which also gives an in-depth knowledge about the subject of interest and provides insights into the problem. But caution should be observed with respect to data as it may contain null values, or redundant values, or various types of ambiguity, which also demands pre-processing of data. The dataset should therefore be explored as much as possible.

Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes.

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00

Preprocessing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns and model values for categorical columns. Maximum and minimum values in numerical columns, along with their percentile values for median, play an important factor in deciding which value to be chosen at priority for further exploration tasks and analysis. Data types of different columns are used further in label processing and a one-hot encoding scheme during the model building.

2.2 Logging

We should be able to log every activity done by the user

- The system identifies at which step logging require.
- The system should be able to log each and every system flow.
- Developers can choose logging methods. Also can choose database logging.
- The system should be not be hung even after using so much logging. Logging just because we can easily debug issuing so logging is mandatory to do.

2.3 DataBase

The system needs to store every request into the database and we need to store it in such a way that it is easy to retain and look into the records.

The system should capture every data that any user gave and the prediction that has been made by that input.

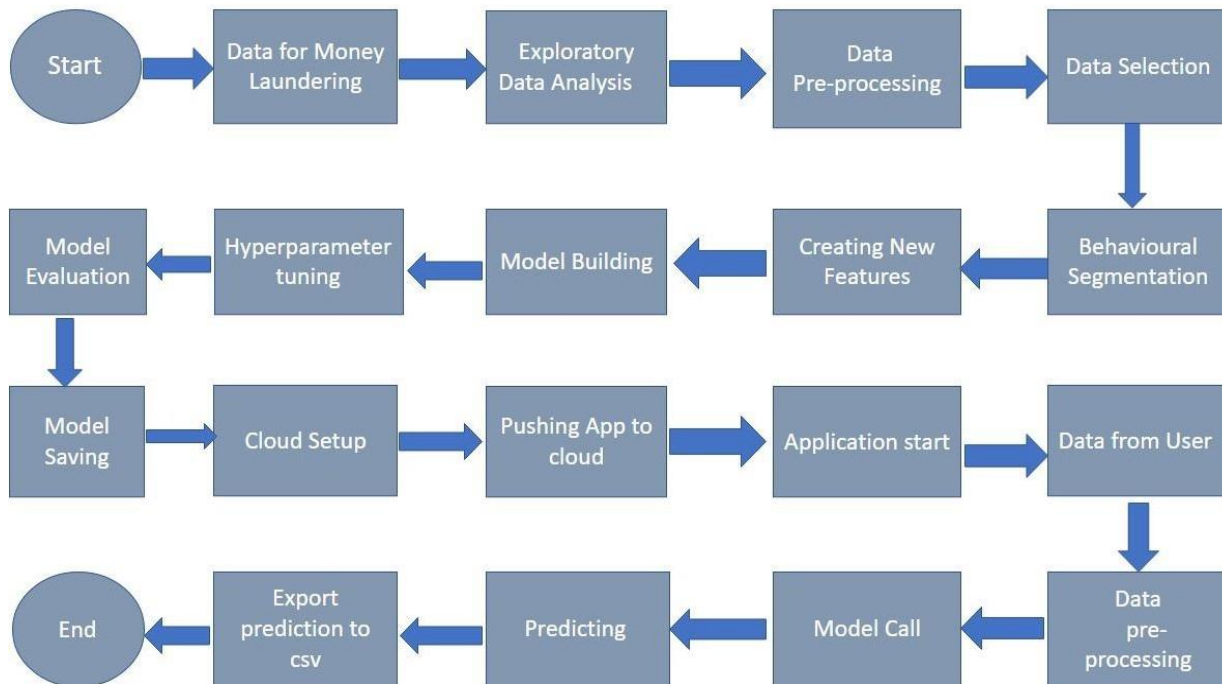
3. Technology Stack

Front End	HTML/JavaScript
Backend	Python/ Flask
Database	MongoDB

4. Proposed Solution

We will use performed EDA to find the important relation between different attributes and will use a machine-learning algorithm to predict the future sales demand. The client will be filled the required feature as input and will get results through the web application. The system will get features and it will be passed into the backend where the features will be validated and preprocessed and then it will be passed to a hyperparameter tuned machine learning model to predict the final outcome.

5. Architecture



5.1 Data Gathering

Data source: <https://www.kaggle.com/ealaxi/paysim1>

Train and Test data are stored in .csv format.

5.2 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because The attributes which contain these are of no use. It will not play role in contributing to the sales of an item from respective outlets.

Like if any attribute is having zero standard deviation, it means that's all the values are the same, its mean is zero. This indicates that either the sale is increasing or decrease that attribute will remain the same. Similarly, if any attribute is having full missing values, then there is no use in taking that attribute into an account for operation. It's unnecessary increasing the chances of dimensionality curse.

5.3 Data Transformation

Before sending the data into the database, data transformation is required so that data are converted into such form with which it can easily insert into the database.

5.4 Database Insertion

Both train and test data set are inserted into the database. Here MongoDB database is used to store the data set. Separate collections were created for both train and test sets.

5.5 Export as `CSV` from Database

From the database both the train and test data set are exported into the local system and stored into CSV files. Now this CSV file will have proceeded for further processing.

5.6 Data Preprocessing

In data preprocessing all the processes required before sending the data for model building are performed.

5.7 Feature Engineering

After preprocessing it was found that some of the attributes are not important to the item sales for the particular outlet. So those attributes are removed. Even one hot encoding is also performed to convert the categorical features into numerical features.

5.8 Parameter Tuning

Parameters are tuned using Randomized searchCV. algorithms used in this problem are decision tree , and XGBoost regressor. The parameters of all these algorithms are tuned and passed into the model.

5.9 Model Building

After doing all kinds of preprocessing operations mention above and performing scaling and hyperparameter tuning, the data set is passed into models, decision tree and XGBoost regressor.

5.10 Model Saving

Model is saved using pickle library in `.pkl` format.

5.11 Flask Setup for Data Extraction

After saving the model, the API building process started using Flask. Web application creation was created here. Whatever the data user will enter and then that data will be extracted by the model to predict the prediction of sales, this is performed in this stage.

5.12 GitHub

The whole project directory will be pushed into the GitHub repository.

7. User Input / Output Workflow.

