Icon

Description automatically generated

**Low Level Design**

Fraud Transaction Detection

|  |  |
| --- | --- |
| Created By | Shraddha Manoj Gujar |

Index

1 . Introduction

* 1. What is a Low-Level design document?
  2. Applications

2. Architecture

3. Architecture Description

3.1 Data Description

3.2 Data Pre-processing

3.3 Data Modelling And Training

3.4 Best Fit Model

3.5 Saving Model

3.6 Front-end Website For project

3.7 Form Data from the website

3.8 Fed Data To ML model

3.9 Prediction

3.10 Deployment

4. Unit Test Cases

1. Introduction

1.1 What is a Low-Level design document?

* The LLD stands for Low-Level Design, in which the designer will focus on the components like a User interface (UI).
* LLD describes detailed description of each and every module means it includes actual logic for every system component and it goes deep into each modules specification.
* The goal a low-level design document is to give the internal logical design of the future state application architecture.
* Structure the design and start with the table of contents with the most important chapters of the document.
* LLD describes the class diagrams with the methods and relations between classes and program specs
* Low-level design is created based on the high-level design.
* It is also known as micro level/detailed design. It is created by designers and developers. It converts the High Level Solution into Detailed solution. It is created second means after High Level Design.

1.2 Applications

* LLD can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms.
* Designer and developer prepare the Low-Level Design.
* The output bases in the low-level design are the unit test plan and program specification.

1. Architecture



Data Validation

App Starts

Data from user

End

Display Output

Prediction

3.Architecture Description

3.1 Data Description

The dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions .

Initially raw data is stored as csv file for the project which was provided by iNeuron

Columns:

▪ Time: Time elapsed in seconds since the first transaction.

▪ V1, V2, ..., V28: Anonymized features resulting from a PCA transformation to

protect user identities.

▪ Amount: Transaction amount.

▪ Class: Binary variable indicating whether the transaction is fraudulent (1) or

not (0).

3.2 Data **Pre-processing**

* **Performing EDA to get insight of data like identifying distribution, statistics , outlier among data etc.**
* **Check for null values in the columns. If present impute the null values.**
* **Visualizing scatter plot of independent variables with each other and output variables.**
* **Perform Standard Scalar to scale down the values.**
* **Perform PCA**
* Data cleaning: this step involves identifying and removing missing, inconsistent, or irrelevant data. This can include removing duplicate records, filling in missing values, and handling outliers.
* Data integration: this step involves combining data from multiple sources, such as databases, spreadsheets, and text files. The goal of integration is to create a single, consistent view of the data.
* Data transformation: this step involves converting the data into a format that is more suitable for the data mining task. This can include normalizing numerical data, creating dummy variables, and encoding categorical data.
* Data reduction: this step is used to select a subset of the data that is relevant to the data mining task. This can include feature selection (selecting a subset of the variables) or feature extraction (extracting new variables from the data).
* Steps Involved in Data Pre-processing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

(a). Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways.

Some of them are:

Ignore the tuples:

This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

Fill the Missing values:

There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

(b). Noisy Data:

Noisy data is a meaningless data that can’t be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways :

* Binning Method:

This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

* Regression:

Here data can be made smooth by fitting it to a regression function. The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

* Clustering:

This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

3.3 Data Modelling And Training:-

Created pipeline:- Created pipeline for every model (Logistic regression, Gradient boosting classifier, Random forest classifier, etc.) We splitted our datset in training and testing set.

Before diving the data in training and validation set we performed preprocessing and featuring scaling.  
The scaling was performed over training and validation data

We splitted our datset in training and testing set.

Algorithms like K-mean, XGBoost, Random Forest etc were used and select our final model and we saved that as Model.pkl.

Model selection :- We find the best model for each pipeline. For each pipeline, we used hyper tuned algorithms. After calculating the AUC scores for all pipeline models

3.4 Best Fit Model

Algorithms like K-mean, XGBoost, Random Forest etc were used and select our final model. We have found “Random forest” best fit for our data.

3.5 Saving Model:-

we saved our model as Model.pkl.

3.6 Front-end Website For project :-

* Created “index.html”
* Created Form for our features (independent variables) and used “Post” method.
* We feed all that data to our saved best fit model ( Random forest model).
* Then if our output is “0” then our transaction is not a fraud transaction otherwise it is a fraud transaction.
* Display this result on screen.

3.7 Form Data from the website:-

Form data is collected from user interface.

3.8 Fed Data To ML model:

Data filled in form is fed to ML model so that it will generate the result.

3.9 Prediction:

ML model will predict that transaction was fraud or not and it will be displayed.

3.10 Deployment

Deployment is done on vercel.com

4. Unit Test Cases

|  |  |  |
| --- | --- | --- |
| **Test Case Description** | **Prerequisite** | **Expected Result** |
| Verify whether the Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads completely for the user when the URL is accessed | 1. Application URL is accessible 2. Application is deployed | The Application should load completely for the user when the URL is accessed |