

Low Level Design (LLD)

Credit Card Fraud Detection



AGAMDEEP
SINGH

Revision Number: 1.8

UNLOCKING INSIGHTS FROM DATA, EMPOWERING THE FUTURE

Agamdeep Singh

Document Version Control

Date Issued	Version	Description	Author
21/05/2023	1.1	First Draft	Agamdeep Singh
23/05/2023	1.2	Added Workflow chart	Agamdeep Singh
26/05/2023	1.3	Constraints, Additional Exception Scenarios	Agamdeep Singh
29/05/2023	1.4	Added KPIs	Agamdeep Singh
1/06/2023	1.5	Added user I/O flowchart	Agamdeep Singh
8/06/2023	1.6	Added EHR, LSTM model diagrams	Agamdeep Singh
15/06/2023	1.7	Added dataset overview and updated user I/O flowchart.	Agamdeep Singh
17/06/2023	1.8	Restructure and reformat LLD	Agamdeep Singh

Contents

Document Version Control	2
Abstract	4
1 Introduction	4
1.1 Why this Low-Level Design Document?	5
1.2 Scope	5
1.3 Constraints	5
1.4 1.3 Risks	5
1.5 1.4 Out of Scope	5
2 Technical specifications	6
2.1 Predicting Disease	6
2.2 Logging	7
2.3 Database	7
3 Technology stack	8
4 Proposed Solution	8
5 Model training/validation workflow	9
6 User I/O workflow	9
7 Exceptional scenarios	9

Abstract

In the twenty-first century, we live in the internet era. Nowadays, internet transactions have become a widespread way of payment, with credit cards being one of the most popular. However, there is a fear of being a victim of deception. Recently, there have been several fraudulent credit card transactions.

Within two days, we discovered 492 fraudulent transactions out of 284,807 transactions in European countries, according to the statistics.

It is critical that credit card companies detect fraudulent credit card transactions so that customers are not charged for products they did not purchase.

To address this grave issue, we attempted to develop a Web Application that uses machine learning techniques to detect these sorts of scams.

1 Introduction

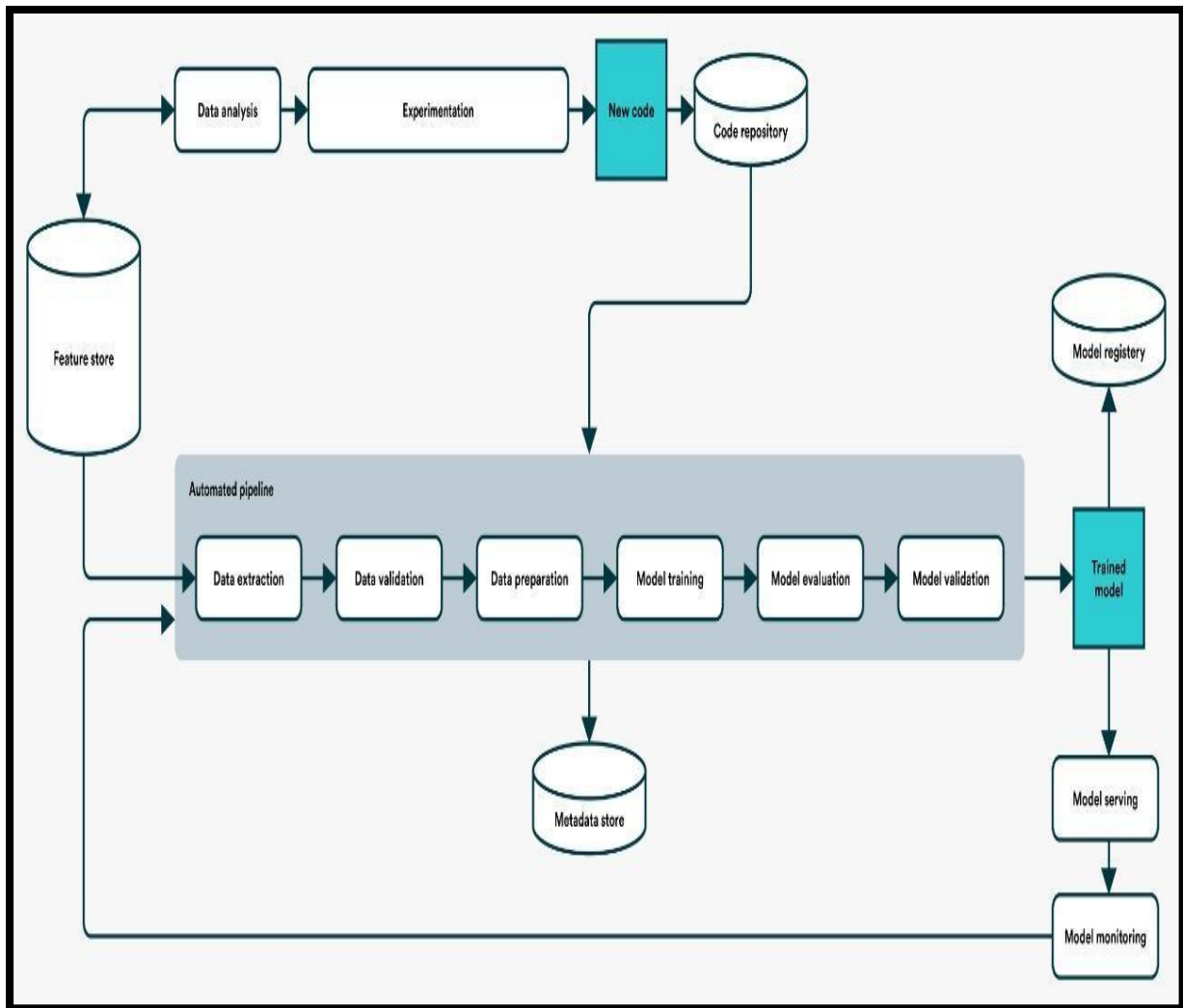
1.1 Why this Low-Level Design Document?

This document's goal is to provide a clear overview of credit card fraud. It will explain the system's goal and characteristics, its interfaces, what the system will perform, the limitations under which it must function, and how the system will respond to external stimuli. This document is meant for both stakeholders and system developers, and it will be sent to upper management for approval.

The main objective of the project is to predict whether a credit card transaction is fraud or not.

Credit card fraud information, such as:

- **Time**
- **V1 to V27 features**
- **Amount**



1.2 Scope

This software system will be implemented as a Web application. This technology will be designed to identify fraud and prevent clients from being paid for products they did not acquire.

1.3 Risks

Specific dangers that have been discovered or should be evaluated should be documented.

1.4 Out of Scope

Define particular actions, skills, and objects that are not part of the project's scope.

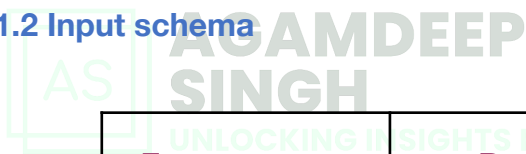
2.1 Dataset

2.1.1 Credit Card dataset overview

Consists of one separate table, which contains credit card information and, most crucially, credit card historical data. The transcript table, on the other hand, contains credit card information.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
me	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25
0	-1.36	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.2396	0.098698	0.3638	0.0908	-0.5516	-0.62	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.1289
0	1.192	0.26615	0.16648	0.448154	0.060018	-0.08236	-0.079	0.085102	-0.255	-0.167	1.612727	1.065	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.167
1	-0.966	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.2376	0.377436	-1.387	-0.055	-0.22649	0.178	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.6473
2	-1.158	0.87774	1.548718	0.403034	-0.40719	0.095921	0.5929	-0.27053	0.8177	0.7531	-0.82284	0.538	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.200
2	-0.426	0.96052	1.141109	-0.16825	0.420987	-0.02973	0.4762	0.260314	-0.569	-0.371	1.341262	0.36	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.232
4	1.23	0.141	0.045371	1.202613	0.191881	0.272708	-0.005	0.081213	0.465	-0.099	-1.41691	-0.15	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.7501
7	-0.644	1.41796	1.07438	-0.4922	0.948934	0.428118	1.1206	-3.80786	0.6154	1.2494	-0.61947	0.291	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.419
7	-0.894	0.28616	-0.11319	-0.27153	2.669599	3.721818	0.3701	0.851084	-0.392	-0.41	-0.70512	-0.11	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.3732
9	-0.338	1.11959	1.044367	-0.22219	0.499361	-0.24676	0.6516	0.069539	-0.737	-0.367	1.017614	0.836	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375	-0.12079	-0.38505	-0.069
10	1.449	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.423	0.048456	-1.72	1.6267	1.199644	-0.67	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.2513
10	0.385	0.61611	-0.8743	-0.09402	2.924584	3.317027	0.4705	0.538247	-0.559	0.3098	-0.25912	-0.33	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.767
10	1.25	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.689	-0.22749	-2.094	1.3237	0.227666	-0.24	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.1611
11	1.069	0.28772	0.828613	2.71252	-0.1784	0.337544	-0.097	0.115982	-0.221	0.4602	-0.77366	0.323	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.5482
12	-2.792	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.423	-1.90711	0.7557	1.1511	0.844555	0.793	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.232
12	-0.752	0.34549	2.057323	-1.46864	-1.15839	-0.07785	-0.609	0.003603	-0.436	0.7477	-0.79398	-0.77	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.039
12	1.103	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.586	0.18938	0.7823	-0.268	-0.45031	0.937	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.3642
13	-0.437	0.91897	0.924591	-0.72722	0.915679	-0.12787	0.7076	0.087962	-0.665	-0.738	0.324098	0.277	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.342
14	-5.401	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.56	0.160842	1.2331	0.3452	0.91723	0.97	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.481
15	1.493	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.081	-0.05313	-1.979	1.6381	1.077542	-0.63	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.3329
17	0.962	0.32846	-0.17148	2.109204	1.129566	1.696038	0.1077	0.521502	-1.191	0.7244	1.69033	0.407	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492	-0.04851	-1.37187	0.3908
18	1.167	0.50212	-0.0673	2.261569	0.428804	0.089474	0.2411	0.138082	-0.989	0.9222	0.744786	-0.53	-2.10535	1.12687	0.003075	0.424425	-0.45448	-0.09887	-0.8166	-0.30717	0.018702	-0.06197	-0.10385	-0.37042	0.607
18	0.247	-0.27767	1.185471	-0.0926	-1.31439	-0.15012	-0.946	-1.61784	1.5441	-0.83	-0.5832	0.535	-0.45338	0.081393	1.555204	-1.39689	0.783131	0.436631	2.177807	-0.23098	1.65018	0.200454	-0.18535	0.423073	0.8203

2.1.2 Input schema



Feature name	Datatype	Null/Required
Time	float	Required
V1to V27	float	Required
Time	float	Required

2.2 Predicting Fraud

- The system displays whether fraud happened or not.
- The system presents the set of inputs required from the user.
- The user gives required information.
- The system should be able to predict whether fraud for the chosen transaction based on the user information.

2.3 Logging

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

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

2.4 Database

Every request must be stored in the database, and it must be stored in such a way that retraining the model is simple.

1. The User gives required information.
2. The system saves all data provided by the user or received on request to the database. You may pick your own database, such as MongoDB, MySQL, or Cassandra.

2.5 Deployment

Deployment is done using AWS



2.6 Technology stack

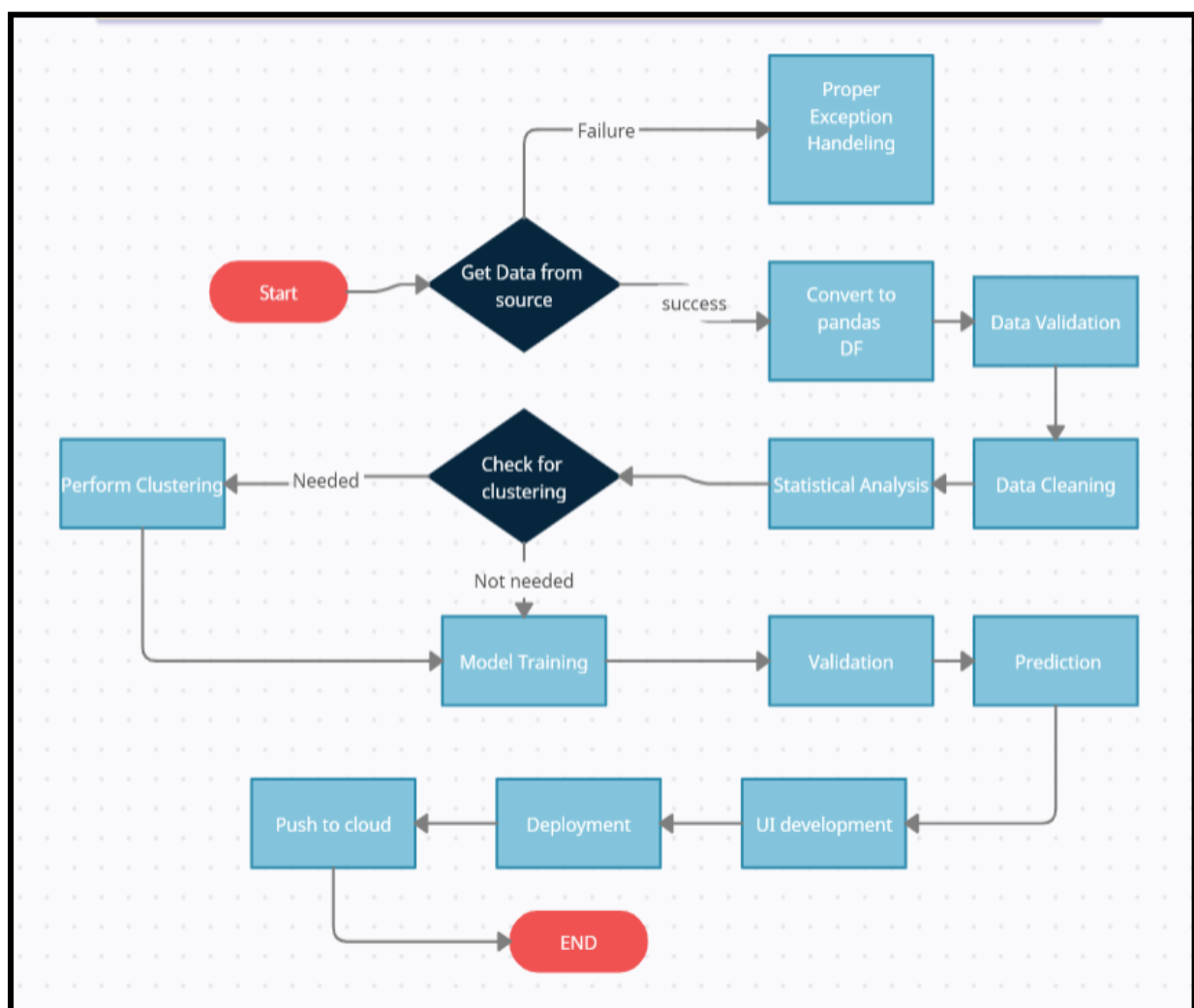
Front End	HTML/CSS/
Backend	Python Flask
Database	Cassandra
Deployment	AWS

Proposed Solution:-

To prevent customers from fraud transactions.

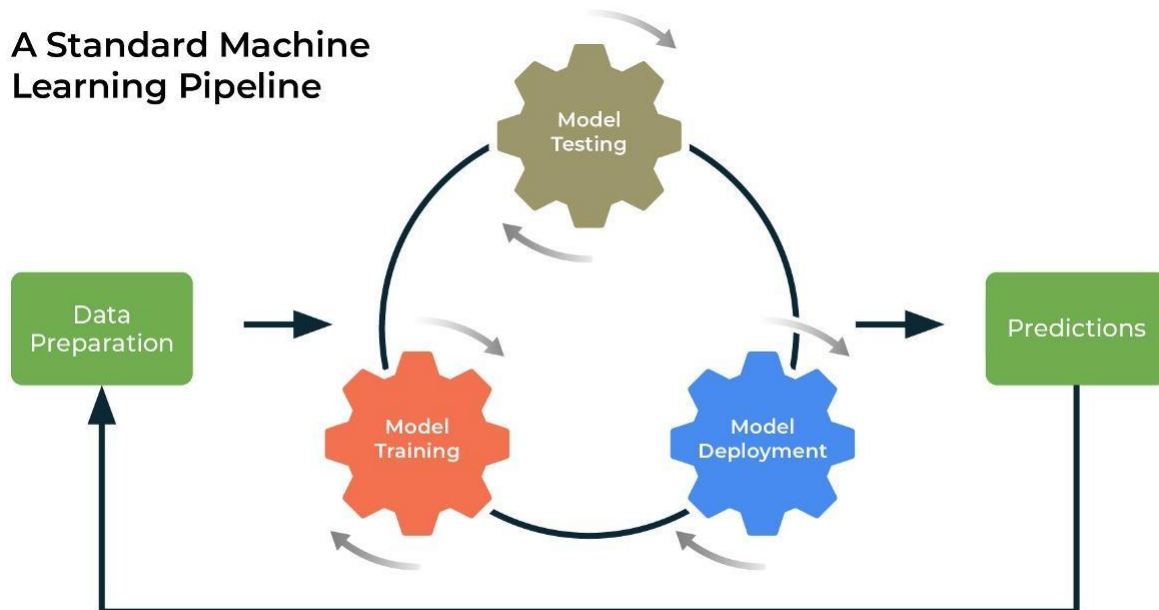
1. **Baseline Model: Random Forest**, since this is a classification problem.
2. **Actual model: Random Forest.**

Model training/validation workflow



3 User I/O workflow

A Standard Machine Learning Pipeline



4 Exceptional scenarios

Exception	Mitigation	Module
1.1	First Draft	Agamdeep Singh
1.8	Last Draft- Added Workflow chart	Agamdeep Singh

5 Key performance indicators (KPI)

- Time and workload reduction using the Random Forest model.
- Comparison of accuracy of model prediction.