CREDIT CARD FRAUD DETECTION

A COURSE PROJECT REPORT

Submitted by

SANDEEP REDDY (RA2011027010165)
SHASHI AKSHITH (RA2011027010172)
SIVANESH G (RA2011027010177)

Under the guidance of

Dr. A. Shanthini

In partial fulfilment for the Course

of

Data science (18CSE396T)

in

DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS

SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
KATTANKULATHUR - 603 203

November, 2022

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project titled "**CREDIT CARD FRAUD DETECTION"** is the bonafide work of **SANDEEP REDDY (RA2011027010165), SHASHI AKSHITH (RA2011027010172), SIVANESH G(RA2011027010177),** who carried out the project work under my supervision.

SUPERVISOR

Dr. A. Shanthini

Associate Professor

Department of Data Science and Business Systems

SRM Institute of Science and Technology

Kattankulathur – 603 203

HEAD OF THE DEPARTMENT

Dr. M. LAKSHMI

Professor & Head

Department of Data Science and Business Systems

SRM Institute of Science and Technology

Kattankulathur – 603 203

ABSTRACT

Credit card frauds are easy and friendly targets. E-commerce and many other online sites have increased the online payment modes, increasing the risk for online frauds. Increase in fraud rates, researchers started using different machine learning methods to detect and analyze frauds in online transactions. The main aim of the paper is to design and develop a novel fraud detection method for Streaming Transaction Data, with an objective, to analyze the past transaction details of the customers and extract the behavioral patterns. Where cardholders are clustered into different groups based on their transaction amount. Then using sliding window strategy, to aggregate the transaction made by the cardholders from different groups so that the behavioral pattern of the groups can be extracted respectively. Later different classifiers are trained over the groups separately. And then the classifier with better rating score can be chosen to be one of the best methods to predict frauds. Thus, followed by a feedback mechanism to solve the problem of concept drift . In this paper, we worked with European credit card fraud dataset. Due to cashless Transaction every people use ATM card and credit card for transaction, so fraud can also be increase. Billions of dollars of loss are caused every year by fraudulent credit card transactions. The design of efficient fraud detection algorithms is key for reducing these losses, and more and more algorithms rely on advanced machine learning techniques to assist fraud investigators.

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor** and **Course Coordinator** for their constant encouragement and support.

We are highly thankful to my Course project Faculty Dr. **Dr. A. Shanthini , Associate Professor, Department of Data Science and Business Systems,** for his assistance, timely suggestion and guidance throughout the duration of this course project.

We extend our gratitude to our **HoD, Dr. M. Lakshmi, Professor, Department of Data Science and Business Systems,** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

TABLE OF CONTENTS

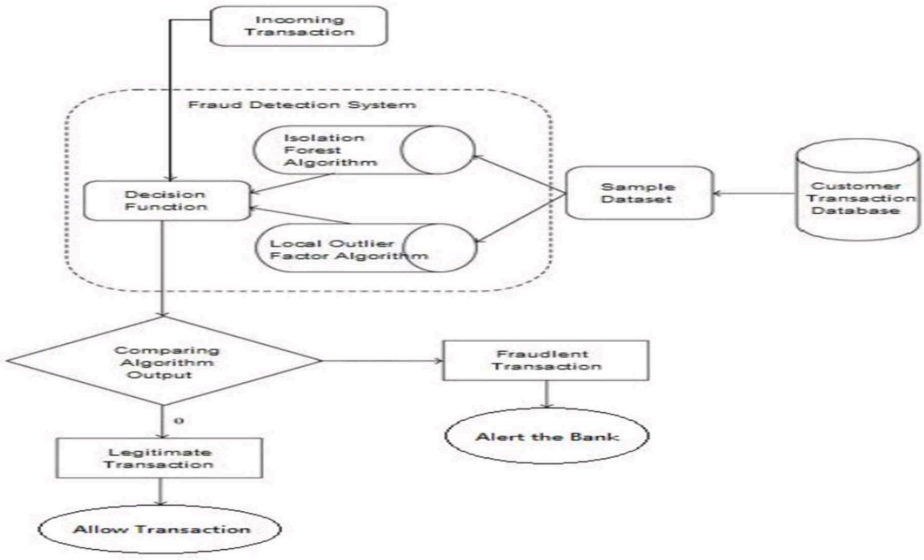
2. INTRODUCTION

1.1 Scenario Description

'Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account. Necessary prevention measures can be taken to stop this abuse and the behavior of such fraudulent practices can be studied to minimize it and protect against similar occurrences in the future. In other words, Credit Card Fraud can be defined as a case where a person uses someone else's credit card for personal reasons while the owner and the card issuing authorities are unaware of the fact that the card is being used. Fraud detection involves monitoring the activities of populations of users in order to estimate, perceive or avoid objectionable behavior, which consist of fraud, intrusion, and defaulting. This is a very relevant problem that demands the attention of communities such as machine learning and data science where the solution to this problem can be automated. This problem is particularly challenging from the perspective of learning, as it is characterized by various factors such as class imbalance. The number of valid transactions far outnumber fraudulent ones. Also, the transaction patterns often change their statistical properties over the course of time.

Fraud detection methods are continuously developed to defend criminals in adapting to their fraudulent strategies. These frauds are classified as:

- Credit Card Frauds: Online and Offline
- Card Theft • Account Bankruptcy
- Device Intrusion
- Application Fraud
- Counterfeit Card
- Telecommunication Fraud



3. LITERATURE SURVEY

- Fraud act as the unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule or policy with an aim to attain unauthorized financial benefit. Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already and are available for public usage.
- A comprehensive survey conducted by Clifton Phua and his associates have revealed that techniques employed in this domain include data mining applications, automated fraud detection, adversarial detection. In another paper, Suman, Research Scholar, GJUS&T at Hisar HCE presented techniques like Supervised and Unsupervised Learning for credit card fraud detection. Even though these methods and algorithms fetched an unexpected success in some areas, they failed to provide a permanent and consistent solution to fraud detection.
- A similar research domain was presented by Wen-Fang YU and Na Wang where they used Outlier mining, Outlier detection mining and Distance sum algorithms to accurately predict fraudulent transaction in an emulation experiment of credit card transaction data set of one certain commercial bank.
- Outlier mining is a field of data mining which is basically used in monetary and internet fields. It deals with detecting objects that are detached from the main system i,e. the transactions that aren't genuine. They have taken attributes of customer's behaviour and based on the value of those attributes they've calculated that distance between the observed value of that attribute and its predetermined value.
- Unconventional techniques such as hybrid data mining/complex network classification algorithm is able to perceive illegal instances in an actual card transaction data set, based on network reconstruction algorithm that allows creating representations of the deviation of one instance from a reference group have proved efficient typically on medium sized online transaction. There have also been efforts to progress from a completely new aspect. Attempts have been made to improve the alert feedback interaction in case of fraudulent transaction.
- In case of fraudulent transaction, the authorized system would be alerted and a feedback would be sent to deny the ongoing transaction. Artificial Genetic Algorithm, one of the approaches that shed new light in this domain, countered fraud from a different direction It proved accurate in finding out the fraudulent transactions and minimizing the number of false alerts.

4. REQUIREMENTS

4.1 Requirement Analysis

- Dataset of Credit Card and their Hex Code values
- Python Libraries (OpenCV, Pandas)
- LOGISTIC REGRESSION

4.2Hardware Requirement

RAM: 4GB and Higher

Processor: Intel i3 and above

Hard Disk: 500GB: Minimum

4.3 Software Requirement

OS: Windows or Linux

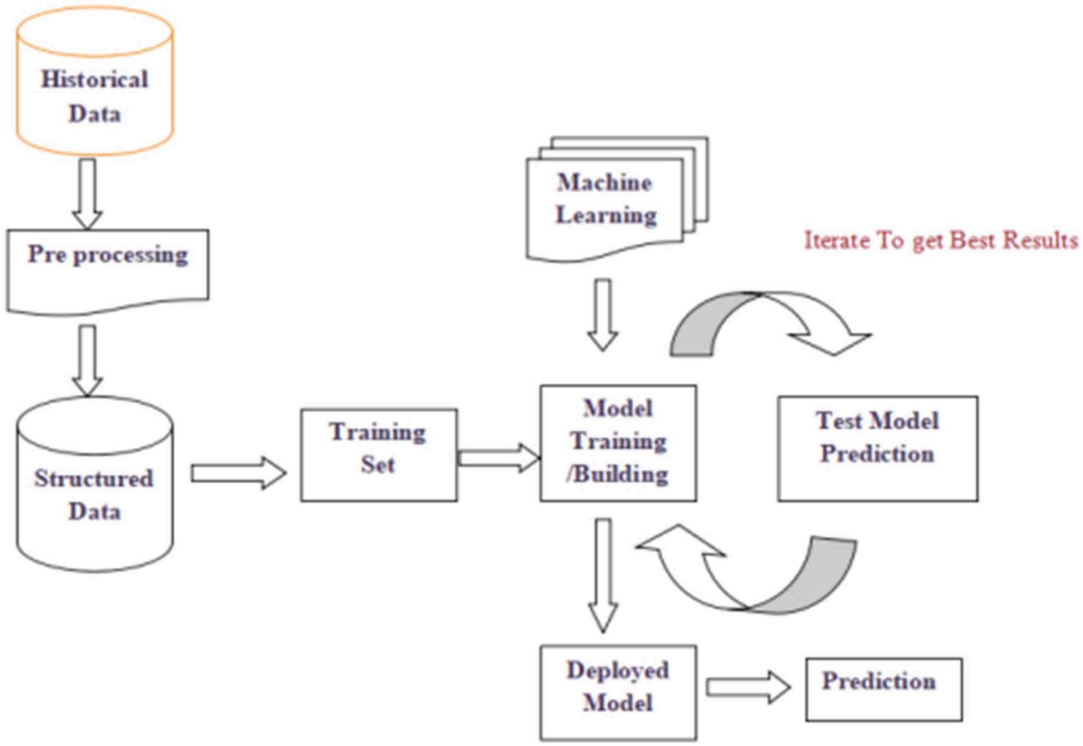
Python IDE: python 2.7.x and above

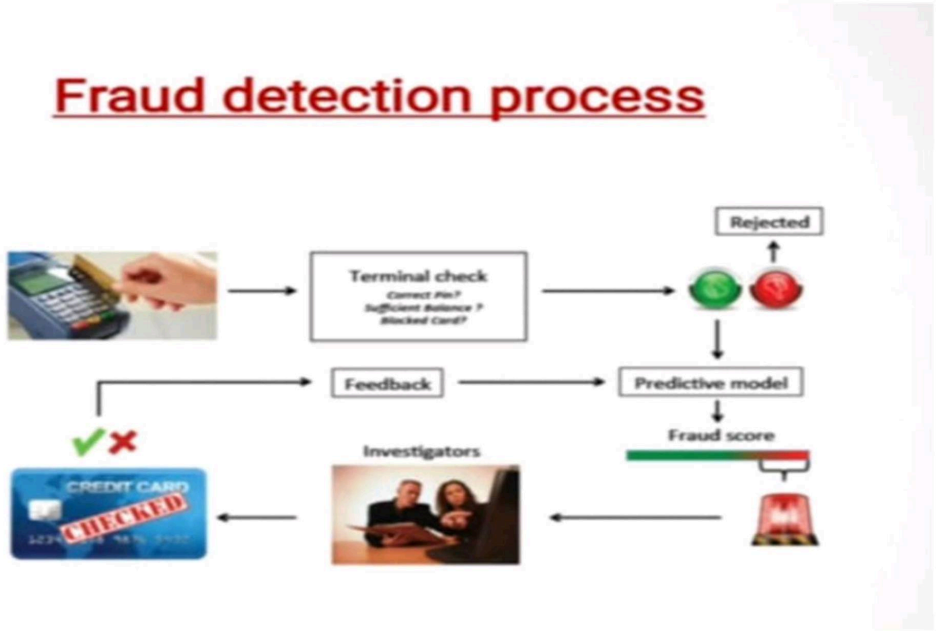
Jupyter Notebook

Setup tools and pip to be installed for 3.6 and above

Language : Python

5. ARCHITECTURE & DESIGN





6. DATA SET DESCRIPTION

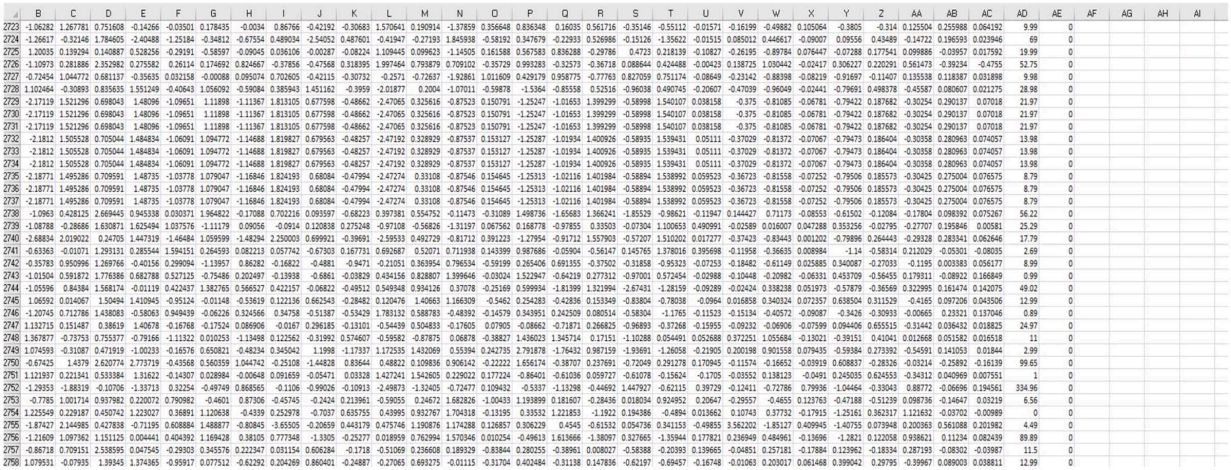
The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions**.**

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data

https://www.kaggle.com/code/renjithmadhavan/credit-card-fraud-detection-using-python

	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	AA	AB	AC	AD	AE
1	Time	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	V26	V27	V28	Amount	Class
2	0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.128539	-0.18911	0.133558	-0.02105	149.62	0
3	0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.16717	0.125895	-0.00898	0.014724	2.69	0
4	1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679	0.909412	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	0
5	1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.647376	-0.22193	0.062723	0.061458	123.5	0
6	2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.20601	0.502292	0.219422	0.215153	69.99	0
7	2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.105915	0.253844	0.08108	3.67	0
8	4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.750137	-0.25724	0.034507	0.005168	4.99	0
9	7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	0
10	7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.373205	-0.38416	0.011747	0.142404	93.2	0
11	9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.451773	0.203711	-0.24691	-0.63375	-0.12079	-0.38505	-0.06973	0.094199	0.246219	0.083076	3.68	0
12	10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.251367	-0.12948	0.04285	0.016253	7.8	0
13	10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.76731	-0.49221	0.042472	-0.05434	9.99	0
14	10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.161135	-0.35499	0.026416	0.042422	121.5	0
15	11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.548265	0.104094	0.021491	0.021293	27.5	0
16	12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.23275	-0.23556	-0.16478	-0.03015	58.8	0
17	12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.129394	15.99	0
18	12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.364298	-0.38226	0.092809	0.037051	12.99	0
19	13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	0
20	14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	0
21	15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.332931	-0.22038	0.022298	0.007602	5	0
22	16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	0
23	17	0.962496	0.328461	-0.17148	2.109204	1.129566	1.696038	0.107712	0.521502	-1.19131	0.724396	1.69033	0.406774	-0.93642	0.983739	0.710911	-0.60223	0.402484	-1.73716	-2.02761	-0.26932	0.143997	0.402492	-0.04851	-1.37187	0.390814	0.199964	0.016371	-0.01461	34.09	0
24	18	1.166616	0.50212	-0.0673	2.261569	0.428804	0.089474	0.241147	0.138082	-0.98916	0.922175	0.744786	-0.53138	-2.10535	1.12687	0.003075	0.424425	-0.45448	-0.09887	-0.8166	-0.30717	0.018702	-0.06197	-0.10385	-0.37042	0.6032	0.108556	-0.04052	-0.01142	2.28	0
25	18	0.247491	0.277666	1.185471	-0.0926	-1.31439	-0.15012	-0.94636	-1.61794	1.544071	-0.82988	-0.5832	0.524933	-0.45338	0.081393	1.555204	-1.39689	0.783131	0.436621	2.177807	-0.23098	1.65018	0.200454	-0.18535	0.423073	0.820591	-0.22763	0.336634	0.250475	22.75	0
26	22	-1.94653	-0.0449	-0.40557	-1.01306	2.941968	2.955053	-0.06306	0.855546	0.049967	0.573743	-0.08126	-0.21575	0.044161	0.033898	1.190718	0.578843	-0.97567	0.044063	0.488603	-0.21672	-0.57953	-0.79923	0.8703	0.983421	0.321201	0.14965	0.707519	0.0146	0.89	0
27	22	-2.07429	-0.12148	1.322021	0.410008	0.295198	-0.95954	0.543985	-0.10463	0.475664	0.149451	-0.85657	-0.18052	-0.65523	-0.2798	-0.21167	-0.33332	0.010751	-0.48847	0.505751	-0.38669	-0.40364	-0.2274	0.742435	0.398535	0.249212	0.274404	0.359969	0.243232	26.43	0
28	23	1.173285	0.353498	0.283905	1.133563	-0.17258	-0.91605	0.369025	-0.32726	-0.24665	-0.04614	-0.14342	0.97935	1.492285	0.101418	0.761478	-0.01458	-0.51164	-0.32506	-0.39093	0.027878	0.067003	0.227812	-0.15049	0.435045	0.724825	-0.33708	0.016368	0.030041	41.88	0
29	23	1.322707	-0.17404	0.434555	0.576038	-0.83676	-0.83108	-0.2649	-0.22098	-1.07142	0.868559	-0.64151	-0.11132	0.361485	0.171945	0.782167	-1.35587	-0.21694	1.271765	-1.24062	-0.52295	-0.28438	-0.32336	-0.03771	0.347151	0.559639	-0.28016	0.042335	0.028822	16	0
30	23	-0.41429	0.905437	1.727453	1.473471	0.007443	-0.20033	0.740228	-0.02925	-0.59339	-0.34619	-0.01214	0.786796	0.635954	-0.08632	0.076804	-1.40592	0.775592	-0.94289	0.543969	0.097308	0.077237	0.457331	-0.0385	0.642522	-0.18389	-0.27746	0.182687	0.152665	33	0
31	23	1.059387	-0.17532	1.26613	1.18611	-0.786	0.578435	-0.76708	0.401046	0.6995	-0.06474	1.048292	1.005618	-0.542	-0.03991	-0.21868	0.004476	-0.19355	0.042388	-0.27783	-0.17802	0.013676	0.213734	0.014462	0.002951	0.294638	-0.39507	0.081461	0.02422	12.99	0
32	24	1.237429	0.061043	0.380526	0.761564	-0.35977	-0.49408	0.006494	-0.13386	0.43881	-0.20736	-0.92918	0.527106	0.348676	-0.15254	-0.21839	-0.19155	-0.11658	-0.63379	0.348416	-0.06635	-0.24568	-0.5309	-0.04427	0.079168	0.509136	0.288858	-0.0227	0.011836	17.28	0
33	25	1.114009	0.085546	0.493702	1.33576	-0.30019	-0.01075	-0.11876	0.188617	0.205687	0.082262	1.133556	0.626699	-1.49278	0.520788	-0.67459	-0.52911	0.158256	-0.39875	-0.14571	-0.27383	-0.05323	-0.00476	-0.03147	0.198054	0.565007	-0.33772	0.029057	0.004453	4.45	0
34	26	-0.52991	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.28672	-0.48469	0.87249	0.851636	-0.57175	0.100974	-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.046949	0.208105	-0.18555	0.001031	0.098816	-0.5529	-0.07329	0.023307	6.14	0
35	26	-0.52991	0.873892	1.347247	0.145457	0.414209	0.100223	0.711206	0.176066	-0.28672	-0.48469	0.87249	0.851636	-0.57175	0.100974	-1.51977	-0.28438	-0.31052	-0.40425	-0.82337	-0.29035	0.046949	0.208105	-0.18555	0.001031	0.098816	-0.5529	-0.07329	0.023307	6.14	0
36	26	-0.53539	0.865268	1.351076	0.147575	0.43368	0.086983	0.693039	0.179742	-0.28564	-0.48247	0.8718	0.853447	-0.57182	0.102252	-1.51999	-0.28591	-0.30963	-0.4039	-0.82374	-0.28326	0.049526	0.206537	-0.18711	0.000753	0.098117	-0.55347	-0.07831	0.025427	1.77	0

	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	AA	AB	AC	AD	AE
37	26	-0.53539	0.865268	1.351076	0.147575	0.43368	0.086983	0.693039	0.179742	-0.28564	-0.48247	0.8718	0.853447	-0.57182	0.102252	-1.51999	-0.28591	-0.30963	-0.4039	-0.82374	-0.28326	0.049526	0.206537	-0.18711	0.000753	0.098117	-0.55347	-0.07831	0.025427	1.77	0
38	27	-0.24605	0.473267	1.695738	0.262411	-0.01087	-0.61084	0.793937	-0.24725	0.138879	-0.40101	-0.81205	-0.18352	-0.63011	-0.28622	-0.33705	-0.42811	-0.0306	-0.50457	0.35518	0.045721	-0.1946	-0.33513	-0.07818	0.392784	-0.03116	0.198037	-0.17535	-0.20362	30.49	0
39	27	-1.45219	1.765124	0.611669	1.176825	-0.44598	0.246826	-0.25757	1.092472	-0.60752	0.047156	0.783727	1.096386	-0.26809	0.768648	-0.52437	-0.80882	0.710386	-0.11837	0.979288	0.008713	0.08228	0.325782	-0.06911	0.020962	-0.04467	-0.24344	0.14918	0.120557	1.8	0
40	29	0.99637	-0.12259	0.546819	0.70658	0.13456	1.156995	-0.29456	0.407429	0.337863	-0.40815	0.682646	1.25372	0.391917	-0.03751	0.795302	-1.66821	1.322995	-2.83004	-1.46343	-0.20346	-0.0763	0.108867	0.162231	-0.57562	0.109795	0.373813	0.050552	0.005106	20.53	0
41	29	1.11088	0.168717	0.517144	1.325407	-0.19157	0.019504	-0.03185	0.11762	0.017665	0.044865	1.345075	1.28634	-0.25227	0.274458	-0.81039	-0.58701	0.087451	-0.55047	-0.15475	-0.19012	-0.03771	0.095701	-0.0482	0.232115	0.606201	-0.3421	0.03677	0.00748	6.54	0
42	32	1.249055	-0.62473	-0.71059	-0.9916	1.429973	3.692977	-1.09021	0.967291	0.850149	-0.30708	-0.45625	0.229981	-0.01691	-0.22085	0.362418	0.315222	-0.51227	0.118995	0.57472	0.097853	-0.00629	0.0092	-0.12946	1.11297	0.500382	1.196549	-0.04822	0.005094	29.89	0
43	32	-2.00887	2.198527	0.144242	1.159432	-0.81517	0.182288	-0.61711	1.530817	-0.58683	0.129876	0.658479	1.265049	-0.20652	1.013603	-0.54941	-0.70803	0.869643	-0.09566	0.944395	0.02874	0.094917	0.294983	0.011081	0.015249	0.034211	-0.23614	0.128291	0.117986	2.35	0
44	33	-0.60788	1.031345	1.74045	1.232106	0.418592	0.119168	0.850893	-0.17627	-0.2435	0.148455	-0.387	0.398299	0.481917	-0.36544	0.235545	-1.34781	0.504648	-0.7984	0.75971	0.254325	-0.08733	0.258315	-0.26478	0.118282	0.173508	-0.21704	0.094312	-0.03304	14.8	0
45	33	-0.93573	0.170416	2.746261	-1.07796	-0.30559	0.011577	-0.29618	0.402776	-0.04047	-0.85205	1.552109	0.953561	0.41543	-0.2339	0.452336	0.844368	-0.82274	0.338822	-1.21423	-0.00542	0.401212	1.064864	-0.15833	0.295505	-0.25937	0.754195	0.046664	0.093948	9.1	0
46	34	-0.76226	0.127768	2.650056	-1.87566	-0.89336	-0.26871	-0.41986	0.146233	-0.86964	-0.03865	-0.68612	-0.50291	1.380826	-1.03912	0.652842	1.898748	-0.45841	-0.5598	-0.2951	0.359393	0.43335	1.260403	-0.4047	0.094946	0.480976	-0.09068	0.319223	0.1356	15.99	0
47	34	1.138316	0.056956	0.649419	0.873062	-0.46847	-0.41019	-0.0139	-0.07244	0.306788	-0.26995	-0.0026	1.124304	0.744207	-0.18835	-0.07566	-0.70919	0.381219	-1.37208	-0.2737	-0.07835	-0.16422	-0.2474	0.059405	0.456286	0.361004	0.274414	-0.0025	0.017109	21.34	0
48	34	-0.29154	0.445575	1.249752	-1.73574	0.085756	-0.12192	0.407716	0.095309	0.815902	-1.49119	-0.84619	0.056533	-0.05895	0.151923	1.982595	-0.4433	-0.31825	0.064787	0.613505	-0.03352	-0.06491	-0.12045	-0.15653	-0.80021	-0.00062	-0.8352	0.131001	0.062896	18.95	0
49	34	0.201686	0.497483	1.373913	0.571405	-0.63066	-0.53966	-0.07627	-0.91711	0.270008	-0.48028	-0.51253	0.680906	0.089923	-0.15741	-0.58882	-0.47791	0.225389	-0.66868	-0.0236	-0.2223	0.718961	-0.17188	-0.16621	0.776471	0.81846	0.442547	0.142757	0.219238	9.99	0
50	35	1.386397	-0.79421	0.778224	-0.86471	-1.06413	0.351296	-1.19145	0.052686	-0.3044	0.576517	-1.63111	0.04256	2.047898	-0.73934	1.456222	-0.27205	-0.93201	1.926532	-0.65994	-0.27303	-0.22873	-0.12352	-0.13103	-0.92967	0.181379	1.194928	0.000531	0.019911	30.9	0
51	35	-1.06324	1.418191	1.086673	1.24144	0.002306	0.045902	0.514121	0.241252	-0.1545	0.280673	-0.53195	0.589858	0.555087	-0.11636	0.165886	-1.23721	0.638414	-0.77607	0.692806	0.235009	-0.05723	0.314153	-0.12986	0.114284	-0.02781	-0.25994	0.118754	-0.00896	20.22	0
52	35	1.199356	0.129953	0.863585	1.002635	-0.78376	-0.88468	-0.04074	-0.20807	0.392478	-0.24818	-0.09536	1.314131	1.091681	-0.40935	-0.57799	-0.58128	0.212343	-0.89318	-0.02065	-0.07262	-0.04247	0.198474	-0.03301	1.01329	0.559098	0.401818	-0.00586	0.017936	0.99	0
53	36	-1.00493	-0.98598	-0.03804	3.710061	-6.63195	5.122103	4.371691	-2.00687	-0.27874	-0.23087	0.145155	-0.06316	-0.79959	-0.34196	-0.93053	0.51051	0.092428	0.823984	1.190398	-0.00198	1.393406	-0.38167	0.969719	0.019445	0.570923	0.333278	0.857373	-0.07554	1402.95	0
54	36	-1.16942	1.158314	1.4068	0.860189	-0.10381	0.122035	0.264451	-0.10877	-0.18198	0.659593	0.456317	-0.09776	-0.34616	0.210182	2.345229	-1.17265	0.860019	-1.15422	0.858768	-0.12056	0.024498	-0.12015	0.212986	0.087536	-0.94653	0.2267	-0.8101	0.234957	7.99	0
55	36	1.095525	-0.11609	1.397912	1.497547	-1.04912	0.072839	-0.7238	0.287532	0.996327	-0.14914	-0.50829	-0.02784	-1.32156	-0.0624	0.505751	-0.2093	0.207537	-0.47133	-0.7363	-0.27503	-0.03323	0.093262	0.089498	0.361261	0.250963	-0.37828	0.081024	0.034227	2.09	0
56	37	1.295668	0.341483	0.081505	0.566746	-0.11046	-0.76632	0.073155	-0.1683	0.071837	-0.28104	-0.54953	0.049093	0.30395	-0.31675	1.134897	0.63591	-0.19976	-0.12205	0.004516	-0.06598	-0.32361	-0.92978	0.063809	-0.19357	0.287574	0.127881	-0.02373	0.0252	0.99	0
57	38	0.158332	0.872687	0.965525	1.802127	0.139022	0.110863	0.559079	-0.02907	-0.39491	0.032674	-0.50092	0.631916	0.754694	-0.04766	0.348218	-1.49353	0.757231	-0.68437	1.303359	0.089838	0.11627	0.691685	0.121113	0.116142	-1.11273	-0.30337	0.327533	0.255621	21.76	0
58	39	-0.55406	0.215728	0.844265	-0.75307	0.034848	-0.5586	1.090401	-0.20387	0.262394	-1.35471	-0.09488	0.953746	0.836964	0.144752	1.15661	-1.93302	0.741349	-0.51361	1.567345	0.383283	0.212134	0.702338	-0.0057	0.147346	0.245652	-0.44046	0.14713	0.139468	120.96	0
59	39	-1.33088	0.26754	-0.16847	-0.70123	3.281972	3.21639	-0.05939	0.89543	-0.01778	-0.16575	-0.76505	-0.01564	-0.35663	-0.0412	-0.7351	0.030533	-0.77807	-0.01516	-0.15282	-0.11388	-0.15104	-0.31173	-0.37479	1.014886	0.353992	-0.51033	0.159033	0.228708	13.84	0
60	40	1.110692	0.081942	0.406063	1.29074	-0.16981	0.120955	-0.08085	0.174373	0.152071	0.065267	1.02455	0.793137	-1.01504	0.422995	-0.70663	-0.48846	0.058531	-0.38363	-0.07407	-0.21985	-0.05602	-0.0116	-0.07048	0.004345	0.603929	-0.32742	0.02989	0.004944	13.84	0
61	41	1.154312	0.265462	0.384871	0.575007	-0.21748	-0.39152	-0.08149	0.062789	-0.26058	-0.16168	2.052928	1.130826	0.189485	-0.07079	0.619345	0.318583	0.090232	-0.33145	-0.34472	-0.09958	-0.19321	-0.55769	0.169492	0.186863	0.089252	0.093626	-0.00963	0.019668	2.67	0
62	41	0.986063	-0.20296	-0.49277	0.407691	0.30566	-0.23053	0.585028	-0.20822	-0.2475	-0.19225	-0.62832	0.430308	0.811248	0.371566	1.026079	-0.0285	-0.30755	-1.09965	0.010739	0.263505	-0.30587	-1.21656	-0.0776	-0.74134	0.286881	0.200347	-0.0752	0.027271	169.05	0
63	41	1.138759	-1.19295	1.407131	-0.33007	-2.0695	-0.24218	-1.30664	0.10451	0.134628	0.493931	-0.89519	-0.1827	0.146081	-0.58661	0.797189	-0.89172	-0.07921	1.541588	-0.98359	-0.29931	-0.1562	-0.03057	-0.01972	0.433753	-0.02952	1.141241	-0.00861	0.041564	96.94	0
64	41	1.145524	0.575068	0.194008	2.598192	-0.09221	-1.04443	0.531588	-0.24189	-0.89629	0.757952	-0.44894	-0.66086	-1.30852	0.788864	0.320294	0.295404	-0.28788	-0.45145	-1.01145	-0.19105	0.011106	-0.1197	-0.07651	0.69132	0.633984	0.048741	-0.05319	0.016251	34.13	0
65	42	-0.24936	0.399227	0.068009	-1.06062	2.410399	3.736574	0.316446	0.672296	0.01514	-0.2695	-0.49512	-0.16151	-0.27421	-0.27619	-0.17241	-0.11896	-0.55	-0.14805	0.771762	0.107676	-0.25766	-0.61676	-0.15667	1.031135	0.120617	0.310494	-0.2883	-0.22396	32.51	0
66	42	-0.52267	1.009923	0.27647	1.475289	-0.70701	0.355243	1.559849	-0.39958	-0.47981	0.516352	0.03433	0.589226	1.42691	0.02051	1.651336	-0.96134	0.24636	-0.40754	0.917399	-0.01302	0.172401	1.011543	0.069666	0.15782	-1.10922	-0.30237	0.31817	0.31691	243.66	0
67	44	-0.89999	0.136255	1.883665	-0.209	1.051441	1.905241	0.241423	0.647631	-0.05347	-0.50417	1.254599	-0.23663	-2.51759	0.384574	0.343832	-1.37056	0.958568	-1.60724	-0.64726	-0.12173	-0.0815	-0.01693	-0.14771	-1.38462	-0.02435	0.412659	-0.10678	-0.19048	21.55	0
68	44	-0.71476	0.514969	1.821676	0.616434	0.848776	-0.11194	1.505617	-0.79817	0.244757	0.264554	-1.10811	-0.35986	-0.16347	-0.784	-0.36715	-0.65312	-0.5043	-0.26056	0.435251	0.107322	-0.22008	-0.01784	-0.43194	-0.13046	0.337159	-0.44004	-0.67453	-0.47322	59.9	0
69	44	-0.9489	0.248414	2.956914	2.81375	0.145539	-0.02735	0.133702	-0.30753	-0.12524	1.03494	-0.53265	-0.18988	0.081222	-0.93905	0.081187	-0.12978	-0.17435	-0.09029	0.222435	-0.05619	-0.08365	0.41609	0.207537	0.716064	-0.60231	-0.06423	-0.31506	-0.27246	0.75	0
70	44	0.92706	-0.32368	0.387585	0.544474	0.246787	1.650358	-0.42758	0.615371	0.226278	-0.2255	1.944545	1.38516	-0.47926	0.314166	0.214167	-1.33372	1.002869	-2.11931	-1.05304	-0.18539	-0.04051	0.079359	0.096632	-0.99257	0.085096	0.377447	0.036096	-0.00596	45.71	0
71	46	-1.92321	-0.87048	2.32017	1.988776	0.417091	-0.38001	0.472139	-0.55733	-0.64908	1.410889	-0.51774	-0.9852	-0.40059	-0.83066	0.338009	0.029924	0.37115	-1.05432	1.890364	-0.36923	-0.68649	-0.77927	1.086055	0.518863	-0.36371	3.065576	-0.58902	-0.39611	35	0
72	46	1.006589	-0.07111	0.347614	1.329684	-0.19324	0.155418	0.008574	0.146538	0.103844	0.03183	1.070981	0.940665	-0.72947	0.380115	-0.72913	-0.4934	0.043556	-0.40529	-0.09258	-0.09137	-0.02024	-0.01888	-0.12097	0.027382	0.593864	-0.33469	0.021368	0.01508	67.97	0



	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	AA	AB	AC	AD	AE	AF	AG	AH	AI
2723	-1.06282	1.267781	0.751608	-0.14266	-0.03501	0.178435	-0.0034	0.86766	-0.42192	-0.30683	1.570641	0.190914	-1.37859	0.356648	0.836348	0.16035	0.561716	-0.35146	-0.55112	-0.01571	-0.16199	-0.49882	0.105064	-0.3805	-0.314	0.125504	0.255988	0.064192	9.99	0				
2724	-1.26617	-0.32146	1.784605	-2.40488	-1.25184	-0.34812	-0.67554	0.489034	-2.54052	0.487601	-0.41947	-0.27193	1.845938	-0.58192	0.347679	-0.22933	0.526986	-0.15126	-1.35622	-0.01515	0.085012	0.446617	-0.09007	0.09556	0.43489	-0.14722	0.196593	0.023946	69	0				
2725	1.20035	0.139294	0.140887	0.528256	-0.29191	-0.58597	-0.09045	0.036106	-0.00287	-0.08224	1.109445	0.099623	-1.14505	0.161588	0.567583	0.836288	-0.29786	0.4723	0.218139	-0.10827	-0.26195	-0.89784	0.076447	-0.07288	0.177541	0.099886	-0.03957	0.017592	19.99	0				
2726	-1.10973	0.281886	2.352982	0.275582	0.26114	0.174692	0.824667	-0.37856	-0.47568	0.318395	1.997464	0.793879	0.709102	-0.35729	0.993283	-0.32573	-0.36718	0.088644	0.424488	-0.00423	0.138725	1.030442	-0.02417	0.308227	0.220291	0.561473	-0.39234	-0.4755	52.75	0				
2727	-0.72454	1.044772	0.681137	-0.35635	0.032158	-0.00088	0.095074	0.702605	-0.42115	-0.30732	-0.2571	-0.72637	-1.92861	1.011609	0.429179	0.958775	-0.77763	0.827059	0.751174	-0.08649	-0.23142	-0.88398	-0.08219	-0.91697	-0.11407	0.135538	0.118387	0.031898	9.98	0				
2728	1.102464	-0.30893	0.835635	1.551249	-0.40643	1.056092	-0.59084	0.385943	1.451162	-0.3959	-2.01877	0.2004	-1.07011	-0.59878	-1.5364	-0.85558	0.52516	-0.96038	0.490745	-0.20607	-0.47039	-0.96049	-0.02441	-0.79691	0.498378	-0.45587	0.080607	0.021275	28.98	0				
2729	-2.17119	1.521296	0.698043	1.48096	-1.09651	1.11898	-1.11367	1.813105	0.677598	0.48662	-2.47065	0.325616	-0.87523	0.150791	-1.25247	-1.01653	1.399299	-0.58998	1.540107	0.038158	-0.375	-0.81085	-0.06781	-0.79422	0.187682	-0.30254	0.290137	0.07018	21.97	0				
2730	-2.17119	1.521296	0.698043	1.48096	-1.09651	1.11898	-1.11367	1.813105	0.677598	0.48662	-2.47065	0.325616	-0.87523	0.150791	-1.25247	-1.01653	1.399299	-0.58998	1.540107	0.038158	-0.375	-0.81085	-0.06781	-0.79422	0.187682	-0.30254	0.290137	0.07018	21.97	0				
2731	-2.17119	1.521296	0.698043	1.48096	-1.09651	1.11898	-1.11367	1.813105	0.677598	0.48662	-2.47065	0.325616	-0.87523	0.150791	-1.25247	-1.01653	1.399299	-0.58998	1.540107	0.038158	-0.375	-0.81085	-0.06781	-0.79422	0.187682	-0.30254	0.290137	0.07018	21.97	0				
2732	-2.1812	1.505528	0.705044	1.484834	-1.06091	1.094772	-1.14688	1.819827	0.679563	0.48257	-2.47192	0.328929	-0.87537	0.153127	-1.25287	-1.01934	1.400926	-0.58935	1.539431	0.05111	-0.37029	-0.81372	-0.07067	-0.79473	0.186404	-0.30358	0.280963	0.074057	13.98	0				
2733	-2.1812	1.505528	0.705044	1.484834	-1.06091	1.094772	-1.14688	1.819827	0.679563	0.48257	-2.47192	0.328929	-0.87537	0.153127	-1.25287	-1.01934	1.400926	-0.58935	1.539431	0.05111	-0.37029	-0.81372	-0.07067	-0.79473	0.186404	-0.30358	0.280963	0.074057	13.98	0				
2734	-2.1812	1.505528	0.705044	1.484834	-1.06091	1.094772	-1.14688	1.819827	0.679563	0.48257	-2.47192	0.328929	-0.87537	0.153127	-1.25287	-1.01934	1.400926	-0.58935	1.539431	0.05111	-0.37029	-0.81372	-0.07067	-0.79473	0.186404	-0.30358	0.280963	0.074057	13.98	0				
2735	-2.18771	1.495286	0.709591	1.48735	-1.03778	1.079047	-1.16846	1.824193	0.68084	-0.47994	-2.47274	0.33108	-0.87546	0.154645	-1.25313	-1.02116	1.401984	-0.58894	1.538992	0.059523	-0.36723	-0.81558	-0.07252	-0.79506	0.185573	-0.30425	0.275004	0.076575	8.79	0				
2736	-2.18771	1.495286	0.709591	1.48735	-1.03778	1.079047	-1.16846	1.824193	0.68084	-0.47994	-2.47274	0.33108	-0.87546	0.154645	-1.25313	-1.02116	1.401984	-0.58894	1.538992	0.059523	-0.36723	-0.81558	-0.07252	-0.79506	0.185573	-0.30425	0.275004	0.076575	8.79	0				
2737	-2.18771	1.495286	0.709591	1.48735	-1.03778	1.079047	-1.16846	1.824193	0.68084	-0.47994	-2.47274	0.33108	-0.87546	0.154645	-1.25313	-1.02116	1.401984	-0.58894	1.538992	0.059523	-0.36723	-0.81558	-0.07252	-0.79506	0.185573	-0.30425	0.275004	0.076575	8.79	0				
2738	-1.0963	0.428125	2.669445	0.945338	0.930371	1.964822	-0.17088	0.702216	0.093597	-0.68223	0.397381	0.554752	-0.11473	-0.31089	1.498736	-1.65683	1.366241	-1.85529	-0.98621	-0.11947	0.144427	0.71173	-0.08553	-0.61502	-0.12084	-0.17804	0.098392	0.075267	56.22	0				
2739	-1.08788	-0.28686	1.630871	1.625494	1.037576	-1.11179	0.09056	-0.0914	0.120838	0.275248	-0.97108	-0.56826	-1.31197	0.067562	0.168778	-0.97855	0.33503	-0.07304	1.100653	0.490991	-0.02589	0.016007	0.047288	0.353256	-0.02795	-0.27707	0.195846	0.00581	25.29	0				
2740	-2.68834	2.019022	0.24705	1.447319	-1.46484	1.059599	-1.48294	2.250003	0.699921	-0.39691	-2.59533	0.492729	-0.81712	0.391223	-1.27954	-0.91712	1.557903	-0.57207	1.510202	0.017277	-0.37423	-0.83443	0.001202	-0.79896	0.264443	-0.29328	0.283341	0.062646	17.79	0				
2741	-0.63363	-0.01071	1.293131	0.285544	1.594151	0.264593	0.082213	0.057742	-0.67303	0.167731	0.692687	0.52071	0.711938	0.143399	0.987686	-0.05904	-0.56147	0.145765	1.378016	0.395698	-0.11958	-0.36635	0.008984	-1.14	-0.58314	0.212029	-0.05301	-0.08035	2.69	0				
2742	-0.35783	0.950996	1.269766	-0.40156	0.299094	-1.13957	0.86282	-0.16822	-0.4881	-0.9471	-0.21051	0.363954	0.796534	-0.59199	0.265406	0.691355	-0.37502	-0.31858	-0.95323	-0.07253	-0.18482	-0.61149	0.025885	0.340087	-0.27033	-0.1195	0.003383	0.056177	8.99	0				
2743	-1.01504	0.591872	1.776386	0.682788	0.527125	-0.75486	0.202497	-0.13938	-0.6861	-0.03829	0.434156	0.828807	1.399646	-0.03024	1.522947	-0.64219	0.277312	-0.97001	0.572454	-0.02988	-0.10448	-0.20982	-0.06331	0.453709	-0.56455	0.179311	-0.08922	0.166849	0.99	0				
2744	-1.05596	0.84384	1.568174	-0.01119	0.422437	1.382765	0.566527	0.422157	-0.06822	-0.49512	0.549348	0.934126	0.37078	-0.25169	0.599934	-1.81399	1.321994	-2.67431	-1.28159	-0.09289	-0.02424	0.338238	0.051973	-0.57879	-0.36569	0.322995	0.161474	0.142075	49.02	0				
2745	1.06592	0.014067	1.50494	1.410945	-0.95124	-0.01148	-0.53619	0.122136	0.662543	-0.28482	0.120476	1.40663	1.166309	-0.5462	0.254283	-0.42836	0.153349	-0.83804	-0.78038	-0.0964	0.016858	0.340324	0.072357	0.638504	0.311529	-0.4165	0.097206	0.043506	12.99	0				
2746	-1.20745	0.712786	1.438083	-0.58063	0.949439	-0.06226	0.324566	0.34758	-0.51387	-0.53429	1.783132	0.588783	-0.48392	-0.14579	0.343951	0.242509	0.080514	-0.58304	-1.1765	-0.11523	-0.15134	-0.40572	-0.09087	-0.3426	-0.30933	-0.00665	0.23321	0.137046	0.89	0				
2747	1.132715	0.151487	0.38619	1.40678	-0.16768	-0.17524	0.086906	-0.0167	0.296185	-0.13101	-0.54439	0.504833	-0.17605	0.07905	-0.08662	-0.71871	0.266825	-0.96893	-0.37268	-0.15955	-0.09232	-0.06906	-0.07599	0.094406	0.655515	-0.31442	0.036432	0.018825	24.97	0				
2748	1.367877	-0.73753	0.755377	-0.79166	-1.11322	0.010253	-1.13498	0.122562	-0.31992	0.574607	-0.59582	-0.87875	0.06878	-0.38827	1.436023	1.345714	0.17151	-1.10288	0.054491	0.052688	0.372251	1.055684	-0.13021	-0.39151	0.41041	0.012668	0.051582	0.016518	11	0				
2749	1.074593	-0.31087	0.471919	-1.00233	-0.16576	0.650821	-0.48234	0.345042	1.1998	-1.17337	1.172535	1.432069	0.55394	0.242735	2.791878	-1.76432	0.987159	-1.93691	-1.26058	-0.21905	0.200198	0.901558	0.079435	-0.59384	0.273392	-0.54591	0.141053	0.01844	2.99	0				
2750	-0.67425	1.4379	2.620774	2.773719	-0.43568	0.560359	1.044742	-0.25108	-1.44828	0.83644	0.48822	0.109836	0.906142	-0.22222	1.656174	-0.38707	0.237691	-0.72049	0.291278	0.170945	-0.11574	-0.16652	-0.03919	0.608837	-0.28326	-0.03214	-0.25892	-0.16139	99.65	0				
2751	1.121937	0.221341	0.533384	1.31622	-0.14307	0.028984	-0.00648	0.091659	-0.05471	0.03328	1.427241	1.542605	0.229022	0.177224	-0.86401	-0.61036	0.059727	-0.61078	-0.15624	-0.1705	-0.03552	0.138123	-0.0491	0.245035	0.624553	-0.34312	0.040969	0.007551	1	0				
2752	-1.29353	-1.88319	-0.10706	-1.33713	0.32254	-0.49749	0.868565	-0.1106	-0.99026	-0.10913	-2.49873	-1.32405	-0.72477	0.109432	-0.5337	-1.13298	-0.44692	1.447927	-0.62115	0.39729	-0.12411	-0.72786	0.79936	-1.04464	-0.33043	0.88772	-0.06696	0.194561	334.96	0				
2753	-0.7785	1.001714	0.937982	0.220072	0.790982	-0.4601	0.87306	-0.45745	-0.2424	0.213961	-0.59055	0.24672	1.682826	-1.00433	1.193899	0.181607	-0.28436	0.818034	0.924952	0.20647	-0.29557	-0.4655	0.123763	-0.47188	-0.51239	0.098736	-0.14647	0.03219	6.56	0				
2754	1.225549	0.229187	0.450742	1.223027	0.36891	1.120638	-0.4339	0.252978	-0.7037	0.635755	0.43995	0.932767	1.704318	-0.13195	0.33532	1.221853	-1.1922	0.194386	-0.4894	0.013662	0.10743	0.37732	-0.17915	-1.25161	0.362317	1.121632	-0.03702	-0.00989	0	0				
2755	-1.87427	2.144985	0.427838	-0.71195	0.608884	1.488877	-0.80845	-3.65505	-0.20659	0.443179	0.475746	1.190876	1.174288	0.126857	0.306229	0.4545	-0.61552	0.054736	0.341153	-0.49855	3.562202	-1.85127	0.409945	-1.40755	0.073948	0.200363	0.561088	0.201982	4.49	0				
2756	-1.21609	1.097362	1.151125	0.004441	0.404392	1.169428	0.38105	0.777348	-1.3305	-0.25277	0.018959	0.762994	1.570346	0.010254	-0.49613	1.613666	-1.38097	0.327665	-1.35944	0.177821	0.236949	0.484961	-0.13696	-1.2821	0.122058	0.938621	0.11234	0.082439	89.89	0				
2757	-0.86718	0.709151	2.538595	0.047545	-0.29303	0.345576	0.222347	0.031154	0.606284	-0.1718	-0.51069	0.236608	0.189329	-0.83844	0.280255	-0.38961	0.008027	-0.58388	-0.20393	0.139665	-0.04851	0.257181	-0.17884	0.123962	-0.18334	0.287193	-0.08302	-0.03987	11.5	0				
2758	1.079531	-0.07935	1.39345	1.374365	-0.95917	0.077512	-0.62292	0.204269	0.860401	-0.24887	-0.27065	0.693275	-0.01115	-0.31704	0.402484	-0.31138	0.147836	-0.62197	-0.69457	-0.16748	-0.01063	0.203017	0.061468	0.399042	0.29795	-0.39967	0.089003	0.038811	12.99	0				

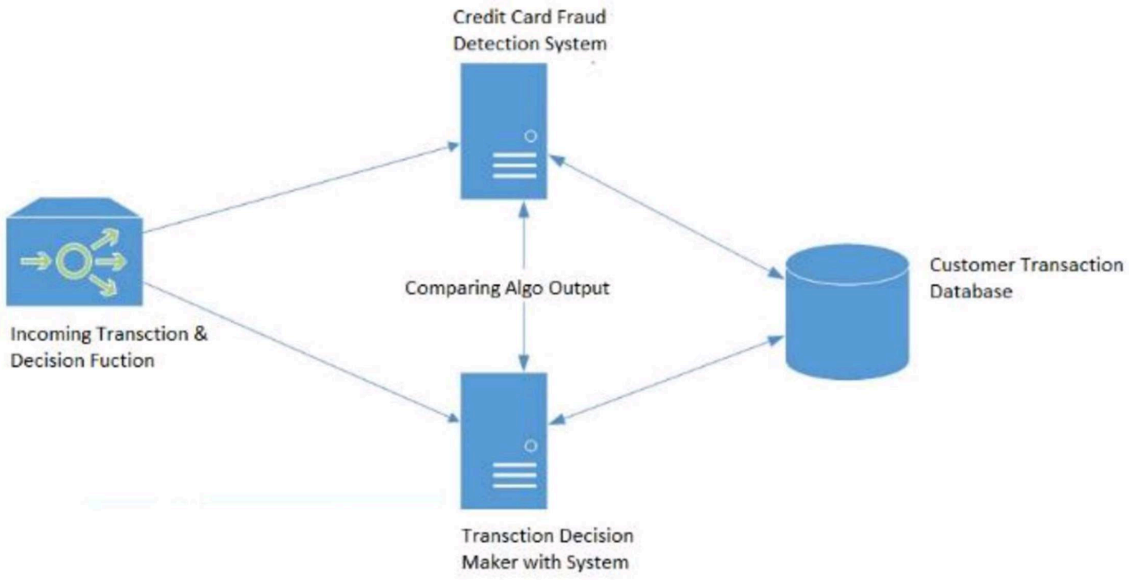
We are using this dada set to detect the credit card frauds happening in the whole system.

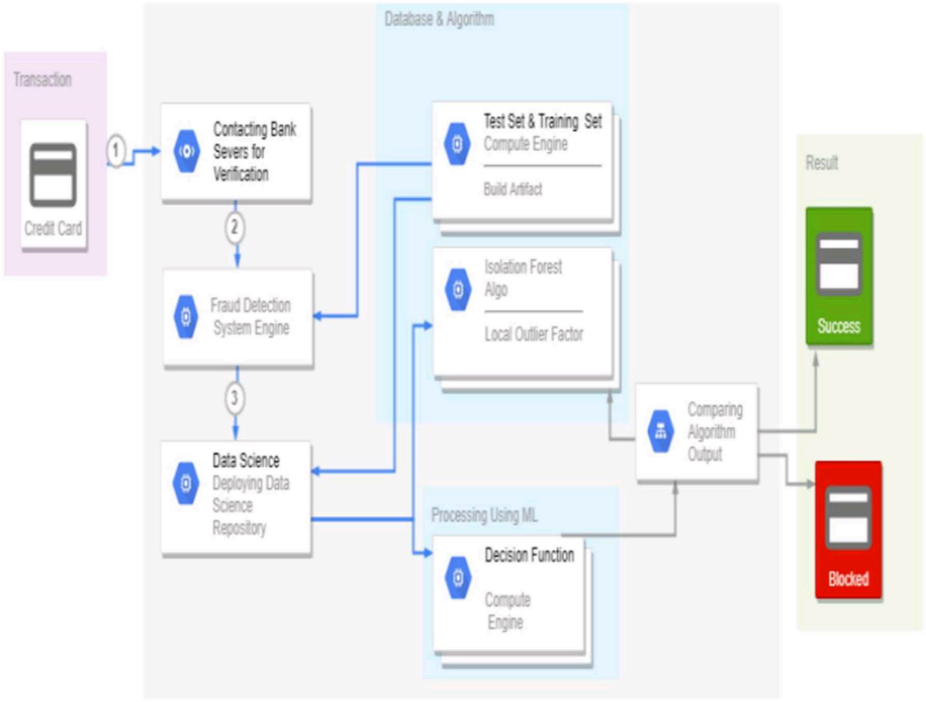
7. METHOD/ALGORITHM/MODEL USED

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college. These binary outcomes allow straightforward decisions between two alternatives.

A logistic regression model can take into consideration multiple input criteria. In the case of college acceptance, the logistic function could consider factors such as the student's grade point average, SAT score and number of extracurricular activities. Based on historical data about earlier outcomes involving the same input criteria, it then scores new cases on their probability of falling into one of two outcome categories.





The approach that this paper proposes, uses the latest machine learning algorithms to detect anomalous activities, called outliers. First of all, we obtained our dataset from Kaggle, a data analysis website which provides datasets. Inside this dataset, there are 31 columns out of which 28 are named as v1-v28 to protect sensitive data. The other columns represent Time, Amount and Class.

8.IMPLEMENTATION

Define project goals, measurement metrics and assign resources

- What are the fraud cases that we wanted to identify?
- What kind of analytics techniques have already been implemented to combat fraud?
- What are the key measurement metrics that we wanted to focus on when assessing the effectiveness of our fraud detection system?

Identify proper data sources

The common data sources for detecting fraud includes:

- client profile
- risk profile
- product usage
- billing data

Develop the data engineering, transformation and modelling pipelines

- For the data engineering pipeline, we need to ingest and merge the data from different sources, aggregate the data based on business metrics, and set up batch processes.
- For the data transformation pipeline, the main goal is to improve the data quality, deal with data issues such as missing & incorrect data and convert the data so that it could be fed into machine learning models.
- For the machine learning model pipeline, we focus on building and comparing diversified ML models based on key business metrics. A module for automated model accuracy testing and re-training is a necessity in the production environment to avoid model drifting issue.

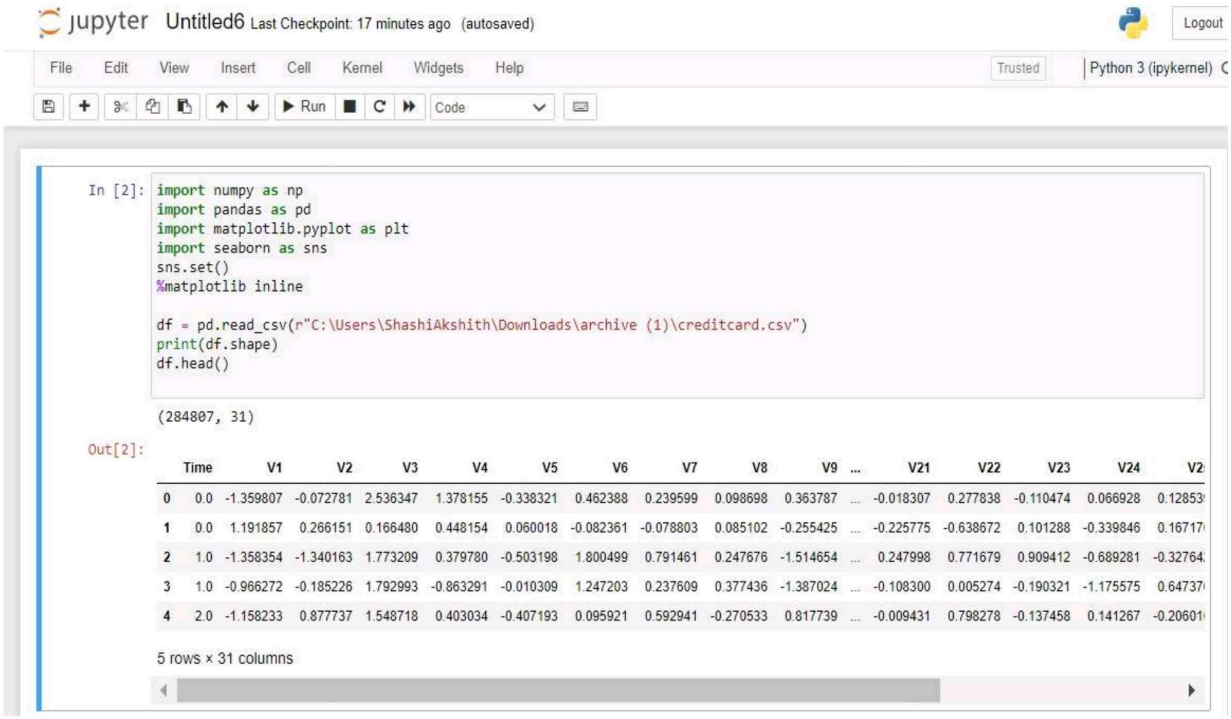
Integrate the model into the case management system

The final step is to incorporate our best performing ML model into the case management system. We can rank the risk level of individual case based on the risk score that we generated. Then, a list of highly suspicious cases will be sent and assigned to relationship managers for further review through the case management system.

9. RESULTS AND DISCUSSION

The code prints out the number of false positives it detected and compares it with the actual values. This is used to calculate the accuracy score and precision of the algorithms. The fraction of data we used for faster testing is 10% of the entire dataset. The complete dataset is also used at the end and both the results are printed. These results along with the classification report for each algorithm is given in the output as follows, where class 0 means the transaction was determined to be valid and 1 means it was determined as a fraud transaction. This result matched against the class values to check for false positives.

Before proceeding with the Random Under Sampling technique we have to separate the original data frame. Why? for testing purposes, remember although we are splitting the data when implementing Random Under Sampling or Oversampling techniques, we want to test our models on the original testing set not on the testing set created by either of these techniques. The main goal is to fit the model either with the that were under sample and oversample (in order for our models to detect the patterns), and test it on the original testing set. Random Under Sampling is basically consists of removing data in order to have a more balanced dataset and thus avoiding our models to overfitting.

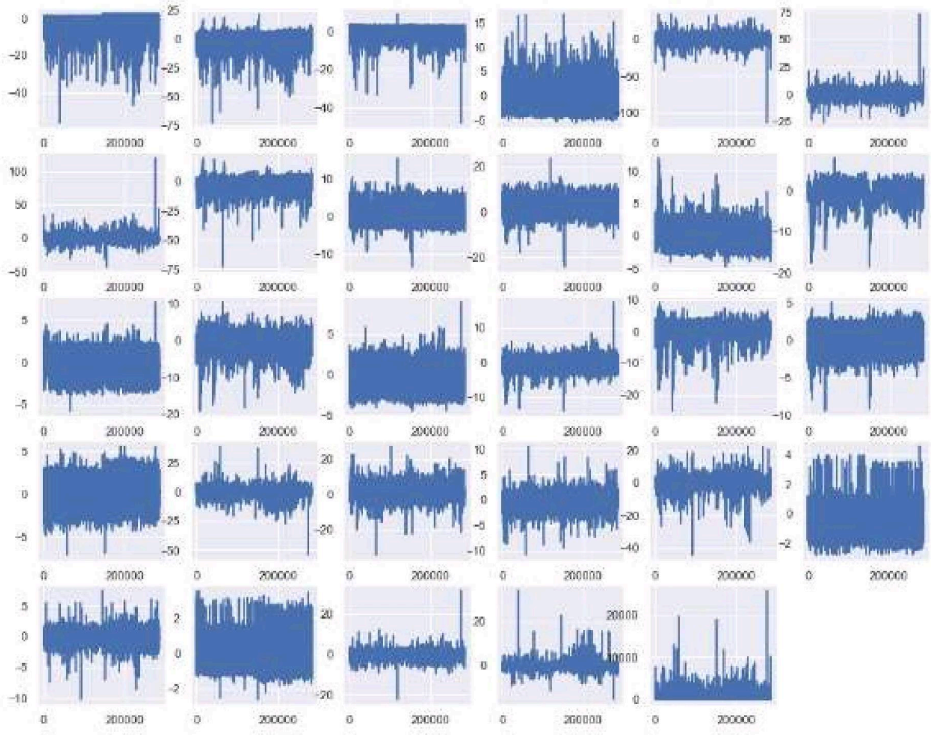


In [5]:
```
class_names = {0:'Not Fraud', 1:'Fraud'}
print(df.Class.value_counts().rename(index = class_names))
```

```
Not Fraud    284315
Fraud           492
Name: Class, dtype: int64
```

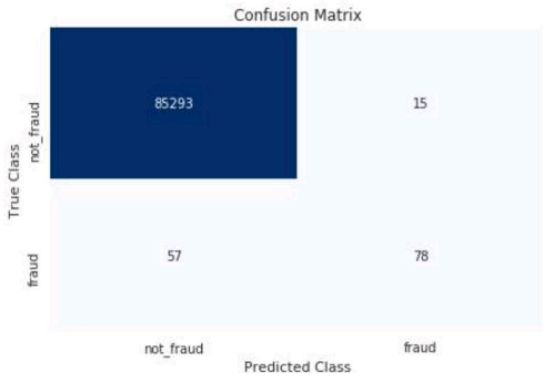
In [6]:
```
fig = plt.figure(figsize = (15, 12))

plt.subplot(5, 6, 1) ; plt.plot(df.V1) ; plt.subplot(5, 6, 15) ; plt.plot(df.V15)
plt.subplot(5, 6, 2) ; plt.plot(df.V2) ; plt.subplot(5, 6, 16) ; plt.plot(df.V16)
plt.subplot(5, 6, 3) ; plt.plot(df.V3) ; plt.subplot(5, 6, 17) ; plt.plot(df.V17)
plt.subplot(5, 6, 4) ; plt.plot(df.V4) ; plt.subplot(5, 6, 18) ; plt.plot(df.V18)
plt.subplot(5, 6, 5) ; plt.plot(df.V5) ; plt.subplot(5, 6, 19) ; plt.plot(df.V19)
plt.subplot(5, 6, 6) ; plt.plot(df.V6) ; plt.subplot(5, 6, 20) ; plt.plot(df.V20)
plt.subplot(5, 6, 7) ; plt.plot(df.V7) ; plt.subplot(5, 6, 21) ; plt.plot(df.V21)
plt.subplot(5, 6, 8) ; plt.plot(df.V8) ; plt.subplot(5, 6, 22) ; plt.plot(df.V22)
plt.subplot(5, 6, 9) ; plt.plot(df.V9) ; plt.subplot(5, 6, 23) ; plt.plot(df.V23)
plt.subplot(5, 6, 10) ; plt.plot(df.V10) ; plt.subplot(5, 6, 24) ; plt.plot(df.V24)
plt.subplot(5, 6, 11) ; plt.plot(df.V11) ; plt.subplot(5, 6, 25) ; plt.plot(df.V25)
plt.subplot(5, 6, 12) ; plt.plot(df.V12) ; plt.subplot(5, 6, 26) ; plt.plot(df.V26)
plt.subplot(5, 6, 13) ; plt.plot(df.V13) ; plt.subplot(5, 6, 27) ; plt.plot(df.V27)
plt.subplot(5, 6, 14) ; plt.plot(df.V14) ; plt.subplot(5, 6, 28) ; plt.plot(df.V28)
plt.subplot(5, 6, 29) ; plt.plot(df.Amount)
plt.show()
```



In [13]:

```
class_names = ['not_fraud', 'fraud']
matrix = confusion_matrix(y_test, pred)
# Create pandas dataframe
dataframe = pd.DataFrame(matrix, index=class_names, columns=class_names)
# Create heatmap
sns.heatmap(dataframe, annot=True, cbar=None, cmap="Blues", fmt = 'g')
plt.title("Confusion Matrix"), plt.tight_layout()
plt.ylabel("True Class"), plt.xlabel("Predicted Class")
plt.show()
```



1. True Positive Rate, which can be defined as the number of fraudulent transactions that are even classified by the system as fraudulent.

2. True Negative Rate, which can be defined as the number of legitimate transactions that are even classified as legitimate by the system.

3. False Positive Rate, which can be defined as a number of the legal transactions which are wrongly classified as fraud.

4. False Negative Rate is defined as the transactions that are fraud but are wrongly classified as legal.

For a financial institution dealing with identifying fraud, Sensitivity and F1 - Score might be more important metrics. F1- Score represents a more balanced result as it is the harmonic mean between Precision and Recall. Sensitivity is more important in the sense that we are more interested in identifying fraud than than identifying legitimate customers.

In [14]:
```
from sklearn.metrics import f1_score, recall_score
f1_score = round(f1_score(y_test, pred), 2)
recall_score = round(recall_score(y_test, pred), 2)
print("Sensitivity/Recall for Logistic Regression Model 1 : {recall_score}".format(recall_score
= recall_score))
print("F1 Score for Logistic Regression Model 1 : {f1_score}".format(f1_score = f1_score))
```

```
Sensitivity/Recall for Logistic Regression Model 1 : 0.58
F1 Score for Logistic Regression Model 1 : 0.68
```

10. CONCLUSION AND FUTURE ENHANCEMENT

Credit card fraud is without a doubt an act of criminal dishonesty. This article has listed out the most common methods of fraud along with their detection methods and reviewed recent findings in this field. This paper has also explained in detail, how machine learning can be applied to get better results in fraud detection along with the algorithm, pseudocode, explanation its implementation and experimentation results. While the algorithm does reach over 99.6% accuracy, its precision remains only at 28% when a tenth of the data set is taken into consideration. However, when the entire dataset is fed into the algorithm, the precision rises to 33%. This high percentage of accuracy is to be expected due to the huge imbalance between the number of valid and number of genuine transactions. Since the entire dataset consists of only two days' transaction records, its only a fraction of data that can be made available if this project were to be used on a commercial scale. Being based on machine learning algorithms, the program will only increase its efficiency over time as more data is put into it.

While we couldn't reach out goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project. More room for improvement can be found in the dataset. As demonstrated before, the precision of the algorithms increases when the size of dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves.

11. REFERENCES

[1] “Credit Card Fraud Detection Based on Transaction Behavior -by John Richard D. Kho, Larry A. Vea” published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017.

[2] CLIFTON PHUA1, VINCENT LEE1, KATE SMITH1 & ROSS GAYLER2 “ A Comprehensive Survey of Data Mining-based Fraud Detection Research” published by School of Business Systems, Faculty of Information Technology, Monash University, Wellington Road, Clayton, Victoria 3800.

[3] “Survey Paper on Credit Card Fraud Detection by Suman” , Research Scholar, GJUS&T Hisar HCE, Sonepat published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014.

[4] “Research on Credit Card Fraud Detection Model Based on Distance Sum – by Wen-Fang YU and Na Wang” published by 2009 International Joint Conference on Artificial Intelligence.

[5] “Credit Card Fraud Detection through Parenclitic Network Analysis By Massimiliano Zanin, Miguel Romance, Regino Criado, and SantiagoMoral” published by Hindawi Complexity Volume 2018, Article ID 5764370, 9 pages.

[6] “Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy” published by IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, VOL. 29, NO. 8, AUGUST 2018