

Raghav Sukhwal¹, Shivam Sharma², Dhruv Kumar³, Mr Amit Kumar⁴.

^{1,2,3},Department of Computer Science Application Sharda University Noida, Uttar Pradesh, India.

⁴Department Of Computer Science Engineering Sharda University Noida, Uttar Pradesh, India

ABSTRACT: For clients to avoid being charged for items they did not buy, credit card companies must be able to identify fraudulent credit card transactions. To overcome such challenges, Data Science and Machine Learning might be applied. This study uses Credit Card Fraud Detection to show how machine learning can be used to model a data collection. The Credit Card Fraud Detection Issue includes modelling previous credit card transactions using information from transactions that turned out to be fraudulent.

The model is then applied to assess the likelihood of fraud in a new transaction. Our objective is to eliminate erroneous fraud classifications while detecting all fraudulent transactions. Credit card fraud detection is an excellent illustration of classification. During this process, we focused on analysing and pre-processing large data sets as well as implementing multiple anomaly detection methods.

1. INTRODUCTION

A credit card allows the individual identified on it to charge products or services to his account, for which he will get regular invoices. It also carries identity information such as a signature or a photograph. The information on the card is currently read by automated teller machines (ATMs), store readers, banks, and online internet banking systems.

They have a unique card number, which is important. Both the plastic card's physical security and the privacy of the credit card number are necessary for its security.

A major rise in fraudulent activity has been brought on by the quick growth in credit card transactions. Credit card fraud is defined as theft or fraud that uses a credit card as a false source of funds during a transaction.

Statistical approaches and a range of data mining algorithms are frequently utilised to address this fraud detection issue. The bulk of credit card fraud detection systems employ artificial intelligence, meta learning, and pattern matching. Evolutionary algorithms called genetic algorithms search for more effective fraud detection techniques. The creation of an effective and secure electronic payment system is given top attention in order to establish whether a transaction is fraudulent or not.

This essay will examine credit card fraud and methods of detection. When someone uses another person's credit card for their own personal use without the owner's knowledge, it is considered credit card fraud. In such cases, it is exploited by fraudsters up until its available credit is depleted. We therefore need a solution that lowers the total credit card limit, which is more susceptible to fraud. Also, a genetic algorithm improves its results over time. The creation of an effective and secure electronic payment system is given top priority for detecting fraud.

II. LITERATURE REVIEW

A fraudulent act is one that is illegal or criminal and is done with the intention of obtaining money or other benefits. It is a deliberate action carried out in contravention of a law, regulation, or policy with the aim of obtaining illegal financial benefit.

A number of publicly accessible literatures on anomaly or fraud detection in this domain have already been published. A thorough study conducted by Clifton Phua and his colleagues revealed that data mining applications, automated fraud detection, and adversarial detection are some of the techniques employed in this field. Unusual methods, based on network reconstruction algorithm that allows creating representations of the divergence of one instance from a reference group, have been effective on medium-sized online transactions. One such method is hybrid data mining/complex network classification algorithm. No algorithm can reliably predict if a transaction is fake, making fraud detection a challenging task.

The following are characteristics of a good fraud detection system:

- It is important to appropriately identify the frauds.
- Frauds need to be found out right away.
- An honest transaction shouldn't be labelled as fraudulent.

III. PROBLEM IDENTIFICATION

The project's objective is to use machine learning algorithms to predict fraudulent credit card transactions. From both the bank's and the customer's perspectives, this is essential. The banks cannot afford to let fraudsters steal their clients' money. Since the bank is accountable for the fraudulent transactions, every fraud results in a.

The dataset includes transactions made by cardholders of European credit cards over the course of two days in September 2013. The dataset is much skewed, with frauds making up 0.172% of all transactions in the positive class. When creating the model, we must be mindful of the data imbalance and use a variety of algorithms to get the optimum model.





IV. FUNCTIONALITY

To detect and prevent credit card fraud, a number of machine learning and deep learning algorithms have been employed, including logistic regression, naive bays, decision trees, SVM, and random forest. By categorizing them into useful groups, this study surveys the machine learning algorithms used for credit card transaction fraud detection. Based on accuracy, precision, recall, etc., a thorough comparison of different methods is also made. In the end, thorough insights.

On the basis of training data, the Classification algorithm is a Supervised Learning technique that is used to categories new observations. A programmer that does classification divides new observations into several classes or groups, such as Yes or No, 0 or 1, etc., after learning from the dataset or observations provided. Targets, labels, or categories can all be used to describe classes. The outcome variable of classification, in contrast to regression, is a category rather than a value. The classification algorithm uses labelled input data, which means that it has input and the associated output, as it is a supervised learning technique.

Classification Algorithms

• Decision Tree Classifier

A supervised learning method called a decision tree can be used to tackle classification and regression issues, although it is most frequently used to solve classification issues.

Supervised Machine Learning techniques like decision trees involve continuously segmenting the data based on a particular parameter. Decision nodes and leaves are the two components that can be used to explain the tree.

Logistic Regression

A given collection of independent variables are utilized to predict the categorical dependent variable using the supervised learning technique known as logistic regression.

In a categorical dependent variable, the output is predicted via logistic regression. As a result, the result must be a discrete or categorical value. Either Yes or No, 0 or 1, etc., are possible. So rather than providing an exact result of 0 or 1, it provides probabilistic values that are in the range of 0 and 1.

• Random Forest Classifier

A popular supervised machine learning technique for solving classification and regression issues is random forest. Using the average for regression and the majority vote for classification, it builds decision trees from a variety of samples. The Random Forest Algorithm's ability to handle data sets with both continuous and categorical variables, as in regression and classification, is one of its key features. It offers better outcomes for classification challenges.

XGBoost Classifier

A supervised machine learning approach used for structured and tabular data is called XGBoost classifier. Both approaches fall under the umbrella of supervised machine learning. A gradient boosted decision tree implementation created for speed and performance is called XGBoost. An extreme gradient boost algorithm is XGBoost. Thus, it is a large machine learning method with numerous components. Large, intricate datasets are compatible with XGBoost. An ensemble modelling method is XGBoost.

	0.0		c	Code													•	Python :	(pyfort
	Export Export Export Export Export Export Export	numpy is ny pandas no p seadorn an skisarn matglotlib. Clib inline warmings s.411 comme	a ans aygber minted																
	datache																		
	Time	*1	47	*9	**	**	**	*7		w	921	¥22	¥23	124	¥25	¥76			
	• 40	1.359807	6472761	2.534547			6.462388	4.339994	0.058404	0.343767 _	0.018107	0.277836	6110474	6.066128				6.62105	1 1404
																	-0.0099		
							1,000409	0.741441		1.554854 -			0.000412	6.48%/91					2 1084
	4 Z0										0.009101				-0.204070				
[15]	top	mat data corr_feats	.corr() correat.l	ndex		-	-				-							
[15]	top_ plt. glt.	mat = data corr_featu figure(fig sns.heatu shape	a.com(ares = psize-(p(data) cornmat.i i0,20)) (top_corr	ndex _feature	HS].CONT	(),amoto	inst, cha	• *Refig	i r")									
[15] [5]: [6]:	<pre>1 corr top_ plt. g = cate. (2848)</pre>	mat - data corr_featu figure(fig sns.heatno shape e7, 31)	a.corr(res = psize(p(data) cornmat.i (0,20)) (top_corr	ndex _feature	s].corr	(),annot-1	nue, cea	, "Xeta	h*)							_		
[13] [5]: [6]:	t corr top_ plt. g = date. (2845 data.	mat = data corr_featu figurs(fig sns.heatnu shape #7, 31) describe()	.com(res = pize(p(deta) cormat.i H0,20)) (top_corr	ndex _featuri	es].corr	(),annot-1	insi, cea	o "Xer3	ir")									
[13] [5]: [6]: [6]:	top_ plt. g = data. (2849	met - deta corr_featu figurs(fig sns.heatu shape e7, 31) describe() T	a.corr(ares = psize:(p(data))) corrmat.i H0,20)) (top_corr	ndex _feature	H3].com	(),amot-l	rue, cma V3	9-78873 9-78873 9-78873	k")	5	16		٧7	ve		5	_	V21
[15] [5]: [6]: [6]:	top_ plt. g = date. (2845 date.	met = deta corr_featu figure(fig sns.heetne shape e7, 32) describe() T 25450/000	e.corr(ares = pize-(p(deta) Tene) cornnat.i H0,20)) (top_corr (top_corr v xxxsc.to=+0	ndex _feature 1 5 2.010	H].com H].com	(),#nnot-1	rus, cma V3 05 2.841	ф-116/13 9 11 11 11 11 11 11 11 11 11 11 11 11 1	17") V 22400/06-1	5	VS St(De+25)	2,1410/76	¥7 +05 21	V8	2.6450	V9 (0e+05)	- 284	V21 0/0++25
[13] (5]) (6]) (6])	top_ plt. E data. (2845 data.	met = deta corr_featu figurs(fig sns.heatnu shape e7, 32) describe() T 20400/000 \$4013.855	ares - pize-(p(deta)))))))))))))))))))) corrnat.3 (0,20)) (top_corr (top_corr 0,0100.00+0 0,0100.00+0 0,0100.00+0	ndex _featury 1 5 2,849 3 3,842	v2 5/06-15 5050-16	(),annot-1 2,8400/0e -4,245724e	rue, cha V3 05 2.841 15 2.81	¢-*Xd12i ¥4 \$0/3≠105 11134=15	<pre>w") v v v v v v v v v v v v v v v v v v v</pre>	5 5 2.540 5 2.540	V6 10/0+05 0130+15	2,5450.1%	v7 +05 2.0 =13 -1.	V8 1000-10+-05 103225516	2,6490	V9 /0e+05 40e+15	- - 2.5% - 1.42	V21 0./0+125 1122+16
[13] (5]) (6]; (6];	top_plt. E = (ats. (ats. dats. count mean std	mat = data corr_feat procession procession procession err, 32) describe() T 254507.000 94013.855 47400.140	a.com(ares = pize (p(data) here 2000 z 1975 1) correat.3 (40,20)) (top_corr (top_corr 2,0450.70+10 3,910430+10 3,910430+10 3,910430+10 3,910450+10	ndex _featury 1 5 2.840 5 5.852 0 1.651	V2 0/0+15 505e-16 205e-20	(),annot-1 2,8400/0e -4,781729e 1,514255e	V3 05 2.841 15 2.81 00 1.415	¢-*XeV3 ¥4 \$0/3+05 1118e-15 565e+00	<pre>w") v zs400/0+1 -1552103+1 1300047+1</pre>	5 5 2.840 5 2.840 6 1.332	V5 10/0+05 0130+15 2271c+00	2,5450./5e -1,699953 1,337094e	V7 +05 21 +15 -1. +00 1.1	V9 H30-16+-05 933285+-16 54353++00	2,5490	V9 /0e+05 40e-15 52e+00	- - 2.6% - 1.62 - 7.34	V21 0/0e+35 1120e-16 1240e-01
[13] (5]) (6]) (6])	data. count data. data. sid min	mat = data corr_feat figure(fig sns.heatm shape e7, 32) describe() T 25450/000 94013.555 47400.145 0.000	a.corr(pize (pize (p(deta)))))))))))))))))))) correat.1 (40,20)) (top_corr (top_corr (0,000/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,0100/0+0 0,000/0+000/0+0000000000	ndex _featury 1 5 2,840 5 5,852 0 1,051 1 -2271	v2 0/0e+05 006+16 006+00 573e+01	(),annot-1 2,8400/0e -4,261729e 1,514255e- 4,813559e-	V3 05 2.641 15 2.61 00 1.415 01 5.660	9-"%dr31 V4 80/3#+05 1118e-15 5053c+00 1171e+00	N") 22000/06-1 -1352/03e-1 1300047e-0 -133243e-0	5 5 2.840 5 2.840 6 1.333 9 -3456	V6 10/0±105 0130±15 1271c±00 2051c±01	2,8450./5e -1,659553 1,23704e 4,25574e	V7 +05 2.1 +00 1.1 +00 1.1	V8 1000-00+-00 0002050-16 94355-+00 016274+01	2,8480 - 3,9476 - 1,9496 - 1,9496	V9 (0+10) 40e-13 33c+00 52e+01	 - 2,840 - 1,477 - 7,340 3,440	V21 0/0e+35 1125e-56 1240e-01 008e+01
[15] (5]) (6]) (6])	count data. (2843 data. count mean std min 25%	mat - data corr_feats figure(fig sns.heatsu shape e7, 32) describe() T 20480/000 94913895 47405.40 6000 Scottag	e.corr(ures = psize() p(deta)))))))))))))))))))) cornwet.3 (top_corn)) (top_corn (top_corn (top_corn)) (top_corn (top_corn)) (top_corn (top_corn)) (top_corn)	ndex _featuri 1 5 2,840 5 3,840 0 1,055 1 -2275 1 5,940	v2 0.55+05 555+05 555+06 573+01 4796 01	(),annot-1 2,8400/0+ -4,781729e 1,514255e 4,81255e 8,50546e	V3 05 2.641 15 2.61 00 1.415 01 5.681 01 8.49	9-**kdr20 90/3#105 11158=15 5853:=400 11278=400 84556:01	N**) 2.5400.06+1 -1.552103+1 1.300.02+4 -1.031413+4 5.5559.74+2	5 5 2.840 5 2.840 0 1.333 0 -3.610 11 7.66	VS 10/0+05 0130+13 1271c+00 4051e+01 7554# 01	2,8480.7% -1,698953 1,23704e 4,25574e 5,548279	V7 +05 24 +05 11 +00 11 +01 -23	V8 H30/5+10 H3525+19 SH353+00 D1629+01 38325401 21629+01 38325/4 (1)	2.5490 - 3.5476 - 1.5454 - 5.4594	19 (0+10) 400-15 30:+00 30:+00 30:+01 30:+01	- 2.5% - 1.47 - 7.34 - 3.483 - 2.2%	V21 0/3+39 1120+9 1100+9 10000000000
[13] (5]) (6]) (6])	<pre>corr tog plt. glt. glt. glt. glt. glt. glt. glt. g</pre>	mat - data corr_feats figure(fig sns.heathu e7, 31) describe() T 20400/000 94013.000 44013.000 54013.000 54013.000 54013.000	*.corr(res = pize(p(deta)) ine 10000 2 1575 1 10000 5 10000 5 10000 5) corrnet.5 (top_corr (top_corr x4405.00+0 3.910440+1 \$50596+0 3.910440+1 1.550596+0 1.01754+0 0.101754+0	rdex _feature 5 2,849 5 5,655 1 -2,775 1 5,980 2 6,540	V2 10/0+15	(),annot-1 2,8400/0+ 4,781729e 1,514255e 4,812559e 8,505,65 1,7905,56	V3 05 2.841 15 2.81 00 1.415 01 5.680 01 4.156	g="36972 ¥4 80/3#105 1119=15 5853c=00 1171z=00 8405a 01 11532e 02	 Y 2.8400/06-1 1.5521036-1 1.5521036-1 1.5521036-1 5.4521036-1 5.4521036-1 	5 5 2.84 6 1.333 0 3.44 8 1.333 0 3.44 8 2.24	V6 80/04-05 91394-15 9271c+00 8051c+01 9554e 01 16/1e 01	2,5400,7% -1,690553 1,23704e 4,25574e 5,560756 4,0103056	¥7 ⊨15 23 ⊨15 41 ±10 11 ≠10 7 ≠10 7	V8 H007/0+105 8932854-16 943551+00 016279+01 08527940 01 2058744 01 2058744 01	2,54% - 2,54% - 1,54% - 1,54% - 5,54%	V9 40x105 40x105 30x400 30x401 31x401 1/be 02	- - 2.840 - 1.47 - 2.34 3.483 - 2.96 - 2.94	V21 0/0+25 1220-95 038+25 038+25 038+25 0494-01
[13] [5]: [6]: [6]:	corr top plt. E data. (2848 data. nd mean nd Sox 50% 75%	mat = dets corr_feats figure(fig sss.heets er, 31) describe() 7 20400/000 94013055 47400.400 5401305 60402.000 9401305 60402.000 9401305 60402.000	a.corr(ares = (psize (p(deta)) ime 19000 2 1975 1 1995 1 1995 1 1995 1 1995 1 1995 1 1995 1 1995 1 1995 1 1995 1 1996 1 1997) correst. (0,20) (top_corr 2010/0+10 0.0100/0+10 0.0100/0+10 0.0100/0+10 0.0100/0+10 0.0100/0+10 0.0100/0+10 0.0100/0+10	1 5 2,549 5 2,549 5 1,655 5 1,655 1 -2279 7 5,965 1 -2279 7 5,965 1 -2279 7 5,965 1 - 2279 7 5,965 1 - 2279 1 - 2079 1 - 20	18].com V2 0/0+15 5506+16 5538+61 4756 01 5556+62 2596+62	(),annot-1 2,8400/0+ 4,261724e 1,514255e 8,505,64 1,02715e 1,02715e	V3 05 2.841 05 2.841 05 3.641 01 4.65 01 4.15 01 1.56 00 7.43	9-13/2/20 V4 80/3+105 1110515 5055+50 5055+50 111712+00 111712+00 14155+02 34132+01	 V 2,640,76+1 4,552,102+1 4,552,102+1 4,552,102+1 5,452,942+2 5,452,942+2 5,452,942+2 5,452,942+2 6,1132,64-2 	5 5 244 5 244 6 1333 7 245 7 45 7 45 7 45 7 45 7 45 7 45 7 45 7	1/6 10/0+105 0130+15 12710+00 40530+01 7/54# 01 15/1+ 01 5643+-01	2,5450/56 -1,69953 1,237544 -4,155744 5,540/56 4,010305 5,700415	V7 105 21 105 21 100 11 100 17 100 7 100 7 1000 7 100 7 1000	V9 H50.19+10 193285+19 194353+00 215479+00 215494+02 213494+02 213494+02	224980 - 3.1696 - 1.3634 - 5.592 5.5925	V9 40e+05 40e+15 22e+01 52e+01 1/5e-02 90e-01	 2340 - 1.42 - 2.34 - 2.34 - 2.34 - 2.34 - 2.34 - 2.34 - 2.34 - 2.34 - 1.06	V21 0/3+35 3125+35 3125+35 315+555 315+555 315+555 315+5555 315+5555 315+55555 315+55555 315+5555555555
[13] [5]: [6]: [6]:	count deta. deta. deta. deta. deta. deta. sount mean ada sount sount mean sount soun	mat = dets corr_feats figure(fig sss.heets er, 31) describe() 7 20400/000 94013.555 47400.140 0.000 54013.555 47400.140 0.000 54013.555 47400.140 0.000 54013.555 47400.140 0.000 5400.140 0.000 5400.140 0.000 5400.1400.140 5400.1400.1400.1400.1400.1400.1400.1400.	a.corr(prize (prize ()) correst. 50,20)) (top_corr 2010/0=10 0.010040=1 0.010040=0 0.010040=0 0.010040=0 0.010040=0 0.010040=0 0.010040=0	1 1 2277 1 23 540 3 1455 1 2277 1 2577 2 5340 0 1655 3 540 0 1655 3 540 3 54	V2 V2 000+15 5066-16 5278+61 5356-52 2239-61 5356-52 2239-61	(),annot-1 2,8400/0+ 4,761726 1,514255e 8,701655 1,627156= 8,382558e	V3 V3 05 2.841 05 2.841 05 4.620 01 4.452 01 4.600 01 1.665 00 7.43 00 1.665	9-13630 V4 80/0+05 5000-00 1070+05 5000-00 1070+00 1070+00 1050+02 5300+01	N**) 2.5400.45+1 4.552103+1 4.552103+1 4.552103+1 4.557515-1 4.557515-1 4.557515-1 4.557515-1 4.557515-1 4.557515-1 4.55751-1 4.55751-1 5.400162+1 3.400162+1 3.400162+1	5 5 2.540 6 1.333 7 2.540 7 3.540 7 3.5400 7 3.54000 7 3.54000 7 3.54000000000000000000000000000000000000	V6 01/0+10 01/0+15 01/0+15 01/0+15 01/0+15 01/0+01 05/0+01 05/0+01	2,8480/06 -1,69953 1,23794e 4,25574e 5,540/09 4,010305 5,700410 1,20505c	V7 105 21 105 21 105 21 105 21 106 1 21 106 1 21 106 2 2 107 2 107 2 107 2 2 107 2 2 107 2	V9 450.0++05 1932255+54 54353+00 21527++01 253454+02 253454+02 203454+02	2.5480 -3.5476 -1.5436 6.4497 -5.5426 5.5713 1.5586	19 10e+05 52e+05 52e+05 11ae 01 11ae 01 900+01 900+01 900+01	 - 28% - 147 - 734 - 734 - 734 - 736 - 238 - 238 - 186 - 238	V21 0/0+15 1226-16 1226-01 108+19 108+108+1008+1008+1008+1008+10000000000

s can be seen, some of our predictors seem to be connected to the class variable. Despite this, there don't seem to be many meaningful links for so many different variables.

plt.see()	frauduk	ent vs Non-Frau	dulent					
250000								
200000 -								
¥ 150000 -								
300000 -								
50000 -								
			*					
fraud_percent deta_fraud_p	tage ('Class	Class 1:['tion-Fraud Datal rane(fro	dent', 'Frau	dulant'], 'Perc	ontago':[sormal	_share, fraud_	shane]}	
: fraud_percent deta_fraud_p sns.barplot(plt.title("N plt.show()	rage ("Class ercentage pd ercentage of fo	Cies Detairano(fri Percentage', c raudulent vs l	elent', 'Frau wd percentage fata data_frac Non-Fraudules!	dulant"], "Perc e) ud_parcentage) (")	ontago":[norma]	_share, fraud_	share]}	
fraud_percent deta_fraud_pe ars_harplot(plt_title("Pe plt_show() Pe	tage ("class preetage pd e-"class", y-" preetage of P preetage of Frau	Cies ":['tion-Fraud Datal rane(fr Parcentage', o raudulent, vs. 1 sdulent, vs. Non-F	alant', 'Frau nd percentag fata data_frav kon-Fraudulent raudulent	dulant"], "Perc x) ud_percentags) 1")	entage":[screal	_share, fraud_	shara]}	
fraud_parcent deta_fraud_p srs.harplot(plt.title("P plt.show() 200	rage - ("Class encentage - pd encentage of fr encentage of fra-	Class	alant', 'Frau wujjercentage fata-data_fra kon-Fraudulent fraudulent	dulant"], 'Perc e) ud_percentage) t"]	entage": [ecreal	_share, fraud_	share]}	
: fraud_percent deta_freed_p pit_title("P pit_title("P 20 80 80	tage - ("Class promises - pd r-"Class", - y-" promises - pd promises - p	Cass "1 ("Inse-Fried Parcentage", resolution: vs. 1 dulent vs. Non if	a aud percentage lata data fra kor-fraudulent raudulent	fulant'], 'Perc *) dipercentage) t')	entago": [norma]	_share, fraud_	share]}	
: fraud_percent deta_frewig_ps psc.lampics() pit.title("Pr pit.show() 200 200 200 200 200 200 200	tage - ("Class ercentage - pd i-"Class", y-" ercentage of fr intertage of Frau	Cass ': ['ton-Fraud Parcentage ', o raubulent vis 1 dulent vis Non 8	a nul persentag iata data fra kon frandulent randulent	dulant"], "Perc e) ad percentage) (*)	ontago" : [sorea]	_share, fraud_	shara]}	
: fraud_parcent dsta_fraud_parcent pst.title("Pp plt.title("Pp 200 80- 200 80- 200 80- 200 80-	tage - ("Class eccentage - pd (-"Class", y-" eccentage of fr mentage of frac	Cass	A Slant', 'Praw www.percenteg fata-data_fra kon-frawdulent readulent	dulant"], "Perc e) ad_percentage) (*)	ontago": [norma]	Johane, fraudj	share]}	
fraud_parcent deta_freed_p pit.lille("M pit.lille" pit.	tage - ("Class ercentage - pd er-Class", y-1 ercentage of fr ecentage of frac	Cass 7: ["toos-kraud Detail reso(fri Parcentage", resolution: vs. Non-B clutent vs. Non-B	A Slamt', 'Prose add percentage fata data (rea son - fraudulent readulent	fulant"], "Perc) ed_percentage) ")	entago": [soreal	Johane, fraudj	share]}	
fraud_parcent deta_freed_p sea_harplit(plt.lile("P plt.lile ao ao ao ao ao ao ao ao ao ao ao ao ao	tage - ("Class ercentage - pr- ercentage of P mentage of Frau	Cass	s Llant ', ' 'Prowe wid juercentag lata data dra dra fea foor-fraudulent 'audulent	fulant"], "Perc e) ed.percentage) (*)	entago": [sormal	share, fraud	share]}	

Distribution of classes with amount and time :- We came to know that fraud is maximum for only lower amount whereas for higher amount fraud is very less.









Split & Train Model

Confusion Matrices

• Decision Tree Classifier

•[39]:	import itertools									
	<pre>tree matrix = confusion matrix(y_test, dt_yhat, labels = [0,1])</pre>									
	<pre>tree_cm_plot = plot_confusion_matrix(tree_matrix,</pre>									
	<pre>classes = ['Non-Default(0)', 'Default(1)'],</pre>									
	normalize = False, title = 'Decision Tree')									
	<pre>plt.savefig('tree_cm_plot.png')</pre>									
	plt.show()									
	Confering Matrix of Decision Tex.									
	Confusion Matrix of Decision Iree									
	- 60000									
	Non-Default(0) - 50000 - 50000									
	2000									
	12 79 - 20000									
	Default(1)									
	- 10000									
	the the									
	and the second sec									
	trant to									
	Predicted label									
[40]-	confusion tree = confusion matrix(u test_dt ubst_labels = [0, 1])									
[0]-	of confusion_tree = confusion_matrix(y_test, dt_ynat, labels = [0, 1])									
1.000	The second se									
[41]:	The confusion_tree[1,1]									
	EP = confusion_tree[0,0]									
	EN = confusion_tree[0,1]									
[42]+	aniat/=Accuracy + = (TD+TH)/()ant(TD+TH+CH+CD))									
[1.	origit("Precision: - TP/float(TP#FP))									
	orint("Sensitivity : ", TP/float(TP+FN))									
	Accuracy : 0.999353950/31/211									
	Precision : 0.0494023033913979									
	Sensitivity : 0./11/11/11/11/									

[48]:	<pre>lr_matrix = confusion_matrix(y_test, lr_yhat, labels = [0,1]) lr_cm_plot = plot_confusion_matrix(lr_matrix,</pre>								
	Confusion Matrix of Logistic Regression								
	Positive(0) -	71045	46	- 60000 - 50000 - 40000					
	Negative(1) -	41	70	- 30000 - 20000 - 10000					
9]:	confusion_lr	Producted Predicted	Habel atrix(y_test	, lr_yhat, labels = [0, 1])					
æ]:	TP = confusion TN = confusion FP = confusion FN = confusion	_lr[1,1] _lr[0,0] _lr[0,1] _lr[1,0]							
i1]:	print("Accurat print("Precis: print("Sensit:	:y : ", (TP+T ion : ", TP/f ivity : ", TP	N)/float(TP+ loat(TP+FP)) /float(TP+FN	TN+FN+FP))))					
	Accuracy : 0. Precision : 0	998778124209	9941 2069						

RandomForest Classifier

•

•

[92]:	from sklearn.onsemble import RandomforestClassifier
[53]:	<pre>rf = RandomforestClassifier(max_depth = 4) rf.fl(Q_train, y_train) rf.fl(Q_train, y_train) rf.fl(train)</pre>
[54]:	print("Accuracy = {}".format(accuracy_score(y_test, rf_yhat)))
	Accuracy = 0.9993258616331002
[55]:	
[55]:	array([[71081, 10], [38, 73]], dtype-int64)
[56]:	<pre>ard pinz, undering, schedup, titley, unseller = fries, unsp = pit.ex.line); title = (content for article (content)); title = (content for article (content)); title = (content for article (content)); title = (content for article (content); title (content); title (content); title (content for article (content, content = co); title (content); title (content for article (content, content = co); title (content, content for article (content, content = co); title (content, content for article (content, content); title (content, content, content = content, c</pre>
	plt, slabel('Predicted label')
	rf_metrix = confusion_metric/t_text; rf_yhrt, lables = {5,1} rf_metrix = confusion_metric/t_metrix, classes = { Trailise(0); "impaties());], metrix= * Trailise(0); "impaties());], metrix= * Trailise(0); "impaties();], plt.immodel former classifier") plt.immodel former classifier Poster(); "Total 10 Poster(); "Total 10 Poster(
	Image: sequence (3) 30 7) 20000 Image: sequence (3) 30 20000 20000 Image: sequence (3) 30 30 30000 Image: sequence (3) 30 30 30000 Image: sequence (3) 30 30 30000
[58]:	confusion rf = confusion matrix(y test, rf yhat, labels = [0, 1])
[59]:	TP + confusion_rf[i_r] Th + confusion_rf[i_r] TP + confusion_rf[i_r] TP + confusion_rf[i_r] TP + confusion_rf[i_r] TP + confusion_rf[i_r]
[69]:	print("Accuracy : ", (IP-IB)/finit(IP-IB-IR-IP)) print("Precision : ", IP/finit(IP-IB-IR-IP)) print("Sensitivity : ", IP/finit(IP-IR))
	Accuracy : 0.9993258616333002 Precision : 0.8795180722893566 Semillivity : 0.85056750570577

XGBoost Classifier



BLOCK DIAGRAM & DESCRIPTION



V. RESULTS

We checked for the accuracy scores after fitting out training data into the model and we received an accuracy of 99.95 percent which is better that the accuracy received from other classification algorithms.

Algorithms	Accuracy	Precision	Sensitivity
Decision Tree Classifier	99.935%	84.946%	71.171%
Logistic Regression	99.877%	60.344%	63.063%
Random Forest	99.932%	87.951%	65.765%
XG Boost Classifier	99.957%	96.551%	75.675%

VI. CONCLUSION AND FUTURESCOPE

Without a question, credit card fraud is a form of criminaldeception. This article outlined the most frequent types offraud, as well as how to detect them, and examined recentresearch findings in the field. This paper also includes a detailed explanation of how machine learning can be used to improve fraud detection findings, as well as the algorithm, pseudocode, explanation, and experimentation results. While the method achieves a precision of over 99.6%, when only a tenth of the data set is taken into account, it only achieves a precision of 28%. When the complete dataset is given into the system, however, the precision increases to 33%. Due to the massive imbalance, such a high percentage of accuracy is to be expected.

VII. REFERENCES

[1] https://en.wikipedia.org/wiki/Logistic_regression

[2] https://stackoverflow.com/questions/22721060/mat%20plotlib-unexpected-gridspecbehavior

[3] https://www.sciencedirect.com/science/article/pii/S%20187705092030065X

[4]https://www.kaggle.com/code/gpreda/credit-%20card-fraud-detection-predictive-%20models/notebook