MEF UNIVERSITY

FRAUD DETECTION IN THE BITCOIN EXCHANGE MARKET

Capstone Project

Hüseyin Namlı

İSTANBUL, 2017

MEF UNIVERSITY

FRAUD DETECTION IN THE BITCOIN EXCHANGE MARKET

Capstone Project

Hüseyin Namlı

Advisor: Asst. Prof. Levent Güntay

İSTANBUL, 2017

EXECUTIVE SUMMARY

FRAUD DETECTION IN THE BITCOIN EXCHANGE MARKET
Hüseyin Namlı

Advisor: Asst. Prof. Levent Güntay

SEPTEMBER, 2017, 30 pages

The trading volume and financial assets of Bitcoin are growing up, while the popularity of Bitcoin world increasing continuously in recent years. In parallel, the market becomes an attraction center for malicious people. In this paper we show how machine learning models can be used for taking preventive actions against fraud attempts in Bitcoin markets. The small sample size of transactions is a big problem for machine learning, however selecting right algorithms and suitable parameter ranges can result in accurate fraud predictions. The prediction of fraud detection involves both numerical and categorical variables; hence classification algorithms can give better results than regression methods. We use a proprietary Bitcoin transaction database supplied by Koinim.com and with 500,000 transactions and 18 fraudulent users. We observed that Classification and Regression Algorithm (C&RT) made better predictions than the Logistic Regression. Also Random Trees improve results of C&RT. We find that the financial volume of transactions users create in their first days is highly correlated with the possibility of being a hit-and-run fraudster. Additionally, the volume of their first transaction is crucial for fraud detection in Bitcoin market. Machine learning also help us to decide the critical threshold for Bitcoin markets, in setting volume limitation for users' transactions in their first days. Koinim.com confirms that, the fraud threshold value estimated by Random Tree model is fairly successful in minimizing fraud loses.

Key Words: Random Trees, Fraud Detection, Bitcoin, Machine Learning, CA&RT, Logistic Regression

ÖZET

BİTCOİN BORSALARI İÇİN DOLANDIRICILIK ÖNLEME PROJESİ

Hüseyin Namlı

Tez Danışmanı: Yrd. Doç. Levent Güntay

EYLÜL, 2017, 30 sayfa

Son yıllarda popülaritesi hızla artan Bitcoin dünyasında, piyasalardaki islem hacmi ve bu hacminin finansal karşılıkları da aynı hızla büyümektedir. Buna paralel olarak sektör kötü niyetli kişiler için de cazibe merkezi haline gelmektedir. Bu noktada bankacılık ve finans dünyasındaki diğer araçlarda oluğu gibi, Bitcoin borsalarında da, dolandırıcılıkla mücadele kapsamında makine öğrenmesi kullanılması mümkündür. Dolandırıcılık örneklerinin sayılarının azlığı, makine öğrenmesi için hala büyük sorun olsa da, doğru algoritma ve uygun parametre seçimleri ile güzel sonuçların elde edilebildiğini görmekteyiz. Dolandırıcılık verisi hem nicel hem de nitel değişkenler içeridiği için, karar ağaçlarının, regresyon yöntemlerine göre daha başarılı tahminlerde bulunduğunu söyleyebiliriz. Özellikle bir sınıflandırma ve regresyon metodu olan C&RT algoritmasının Lojistik Regresyon'a göre oldukça iyi tahminlerde bulunduğu gözlemlendi. Yeni geliştirilmiş metodlardan sayılabilecek ve daha ileri düzey bir algoritma olan Rastgele Ağaçlar algoritmasının ise C&RT ile elde edilen sonuçları daha da geliştirebildiğini gördük. Vur-kaç tipi dolandırıcılıklarda kullanıcıların ilk günlerinde hatta ilk işlemlerinde yarattıkları finansal hacimin, dolandırıcılık tespitinde kullanılabilecek belirgin bir özellik olduğu söylenebilir. Bitcoin borsalarında kullanıcıların ilk günlerindeki işlemlerinde bir hacim kısıtı getirilmek istenirse, makine öğrenmesi modelleri ile bu eşiğin ne olması gerektiği konusunda da çıkarımlar yapılabilir. Koinim.com sitesinin verileri ile yapılan uygulama sonuçlarına göre, Rastgele Ağaçlar algoritmasnın öngördüğü eşiklerin finansal zararı minimize etme konusunda da oldukça başarılı olduğunu söyleyebiliriz.

Anahtar Kelimeler: Rastgele Ağaçlar, Dolandarıcılık Tespit, Bitcoin, Makine Öğrenmesi, C&RT, Lojistik Regresyon

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

1. INTRODUCTION

1.1. Bitcoin

Bitcoin, a digital currency which is based on encoding and decentralized peer-to-peer network technologies, is arisen at 2009. All Bitcoin transactions are enrolled in common registry called blockchain without the influence of any central authority. The registration process is called mining and to mine a new Bitcoin requires sheer volume of processor power. The consistency of those transactions is ensured with encoding technologies as well. There are more than 16 million registered Bitcoins in blockchain which cost over 20 billion dollars. Bitcoin transactions (buy and sell) take place on trade-in web sites called Bitcoin markets. Each subscriber has a private key and public key provided by trade-in markets which enables user to create Bitcoin related transactions. Private Key represents the secure wallet of trader and the Bitcoins in that wallet are only accessible with private keys. Public key enables traders to send Bitcoins to their wallets. Making an analogy with banking system, public key is similar to IBAN and private key is similar the internet banking access information. Koinim.com is one of those markets.

1.2. Koinim.com

Koinim.com is the first Turkey located Bitcoin market which is founded at 2013. It had reached more than 35,000 users at March 2017. Koinim.com provides TL and Bitcoin wallets for users and makes profit by getting commission from both sell and buy transactions. Traders can transfer amount to their TL wallets and can buy Bitcoins buy using that amount.

1.3. Research Problem

Bitcoin transactions have no point of return in order to the used technology which means there is no way to rollback a Bitcoin transfer to Bitcoin wallet. Also anybody can create a Bitcoin wallet by setting up software in the local computer and use it anonymously. Because of those features preventing usage of Bitcoin by fraudsters is one of the most significant focuses of the sector. Those problems have a reflection to koinim.com in two ways: Internet & phone fraudsters, bank account thefts.

Internet & phone fraudsters convince their victims to send amount to their TL wallets by using several fraud methods. Even though victims send those amounts knowingly, when they realize that they are defrauded they might think koinim.com is connected with the crime.

Bank account thefts who steal money from other people's bank accounts send TL to their own TL wallets and buy Bitcoin with those amounts which does not actually belong to them.

There are several fraud preventive mechanism used by koinim.com but those ones who can pass over this actions still cause some legal and financial problems to company. Company relief the victimization and undertake the legal proceedings against the real victims. That situation cost money losses and lawyer expense to company.

According to Sahin et al. (2013), there are two approaches to avoid fraudulent behaviors. The first is to prevent them proactively before they happen and the other one is using fraud detection mechanism to detect fraudulent activities after they happen.

This project will focus on the hit-and-run type of frauds which is sharply different from other fraud types. Hit-and-run fraudsters are being detected and kicked from the system in several days manually in current situation. Those kinds of fraudsters have typical behaviors from this aspect that need to be deeply investigated independently.

1.4. Our Goal

Overall scope of this project is to find solution proposals to koinim.com in order to predict fraud behaviors before they actually happen by analyzing fraudulent customer data.

2. PREPARATORY

This section describes data and provides the estimation methodology. We describe estimation methods, tools and techniques and explain the raw data and data transformation stages.

2.1. Methodology

Subsequent sections describe tools and techniques for each step of the project model in detail. We give a brief summary about these steps in this section.

First, since the data is provided as a MySQL database dump by the company, MySQL is used to get familiar with the raw data. For feature creation, data cleansing, imputation phases MySQL is used as well. MySQL is used for all kind of data manipulation steps and the final data is exported to a csv file to be given as input to machine learning algorithms.

After manipulation operations, deeper exploration on the final data set is done by the help of Tableau software. Almost all figures in section 3 are generated with that software.

In the modelling phase machine learning algorithms are used. "SPSS Modeler" software is used to build ML models and observe the results.

2.2. Raw Data

The company has provided the whole dump of customer and transactional databases with encrypted data in it. Data belongs to a snapshot of a specific date with four-year historic data between January 2013 and March 2017. There were more than 34,000 customers with over 500,000 transactions for that specific date range. Next we describe the Dataset and variables provided by the company.

2.2.1. Demographic User Data

- Subscriber ID
- Subscription Date
- Last Transaction Date
- Gender

- Age
- City

2.2.2. Transaction Data

- TL withdrawal
- TL deposit
- Bitcoin withdrawal
- Bitcoin deposit
- Lightcoin withdrawal
- Lightcoin deposit
- Transaction Date
- Transaction ID

2.2.3. Order Data

- Order type (buy / sell)
- Price (TL)
- Order date

2.3. Data Cleaning

About 20% of registered users have transactionals, the rest of the customers only opened an account. Further, some demographic info is not required for registration. So we met many missing values for demographic information. Only about 6,800 of 34,000 single subscribers have transactional movement. Customers with no transactions cannot involve in any fraudulent behavior, so we ignore data for these users.

The demographic information "gender" has been also ignored in model experiments since it was missing for more than %60 of customers. Making imputation is an option to get rid of missing values but very complex rules need to be applied to impute demographic features such as gender in which there is no guarantee to predict correct values. So gender is generated as a feature but ignored in the models.

The "age" feature is calculated by using the birthday of users. For only 8 customers whose birthdays are missing the values are imputed as "NA". Although age is numerical, it does not have linear relationship with the results. It is better to make a segmentation transformation to age variable according to best practices. For this purpose, a numeric to categorical transformation has been applied on age. Age is divided into 4 segments as very "young", "young", "middle age" and "old".

Another imputation is applied for "country" variables. There are two different countries as "Türkiye" and "Cyprus" in the data set and more 99% is "Türkiye". Therefore imputation for null values is done as "Türkiye".

2.4. Feature Creation

Some features are provided in raw format in customer databases. They can be used in the model after some transformations and selections. The transactional databases contain transactional movements of subscribers. The information in those databases need to be denormalized. Some analytic and aggregation functions are used by the help of MySQL to create the new features.

Feature creation stage consists of several steps. According to the results of many experiments in modelling phases some additional features are created to improve the models. We will try to explain the feature creation order in this section.

Customers can give orders to buy or sell certain amounts of bitcoin or lightcoin via koinim.com web site. So they have coin and TL balances stored in the system. After every coin transaction those balances are updated.

Firstly, averages and counts are calculated from the transaction tables by analyzing bitcoin/lightcoin investment or withdrawal movement.

Transactional tables also include each transactionSo it is possible to reach similar aggregations for specific time periods. In data exploration phase it is recognized that, hit-and-run fraudsters are usually supposed to make their movements in the beginning of their lifecycles. Some additional features related with the first days and transactions of subscribers are created to improve the feature quality. Descriptions for all features are mentioned in **Appendix A**. But there is an important note that we want mention at this point. "Tenure" and "is_active" features are used in exploration phases and provided useful

clues about the data. But they are not used in final models because the aim of the project is to detect fraudulent behavior before ore as soon as they happen.

2.5. Machine Learning for Fraud Detection

Pozollo (2015) argues that Machine Learning plays a key role in Data Driven Fraud Detection Systems as has the same role for many daily application and scientific subjects.

2.5.1. Supervised Algorithms

According to Bolton and Hand (2001), in supervised Algorithms, fraud detection cases are based on differentiating fraudulent and non-fraudulent behaviors. Data should contain samples from both cases in order to detect patterns of two different forms and to decide the borders between areas. Also supervised methods do not work well in detection of new kind of frauds.

Sanjeev et al. (2012) say that Supervised algorithms can be classified as traditional statistical classification methods, rule-based methods, and recent development of power tools.

Logistic regression, linear discriminant analysis (LDA) and fisher discriminant analysis (FDA) are some examples for traditional methods. Mahmoudi and Duman (2014) say that LDA divides th input region into decision areas and those bounderies are used to decide classifications. Decision trees can be categorized as rule based methods. Finally, neural networks, support vector machines and random forest are recently developed advanced algorithms. In modelling phase we will build models with at least one method for each type and estimation results.

2.5.1. Unsupervised Algorithms

Unsupervised algorithms in fraud detection do not require fraudulent transaction samples. The purpose is to define a norm for usual behaviors and detect transactional behaviors located out of boundaries of normal behavioral area.

According to Bolton and Hand (2001), although unsupervised methodology has the ability of detecting new kinds of frauds which are not observed earlier as an advantage over supervised methodology, supervised algorithms are more popular in fraud detection

applications. Supervised Algorithms will be out of scope of this study since their processing complexity and costs are extremely high.

3. Exploratory Data Analysis

Basic exploration steps are used during feature creation steps to increase the informativeness of variables. Deeper exploration is done on the final data set by the help of Tableau software. A brief description of final dataset can be seen in the figure 1.

Field	Measurement	Values
avg_and_first_ratio	Continuous	[0.0,35018.5]
number_of_tx_in_first_day	Continuous	[1.0,185.0]
sum_tl_in_first_day	Continuous	[0.0,5.00051599E8]
created_by_id	Continuous	[12.0,34239.0]
first_tx_date	Continuous	[2013-12-02,2017-03-01]
first_tx_in_day	Continuous	[0.0,1184.0]
first_tx_amt	Continuous	[1.595E-4,17898.0]
first_tx_kurus	Continuous	[0.0,4210000.0]
tenure	Continuous	[0.0,1265.0]
avg_trading_volume	Continuous	[0.0,4069840.0]
id_number_verified	Continuous	[0.0,1.0]
number_of_active_bank_account	Continuous	[0.0,9.0]
number_of_inactive_bank_account	Continuous	[0.0,12.0]
total_bitcoin_buy_order_cnt	Continuous	[0.0,83819.0]
total_light_coin_buy_order_cnt	Continuous	[0.0,2285.0]
total_light_coin_sell_order_cnt	Continuous	[0.0,1502.0]
total_bitcoin_sell_order_cnt	Continuous	[0.0,60835.0]
first_tx_type	Flag	S/B
first_tx_currency	Flag	LTC/BTC
is_fraud	Flag	1.0/0.0
is_active	Flag	1.0/0.0
deactivation_date	Nominal	"42009.460266203707","42019.631967...
age	Nominal	NA,middleage,old,very_young,young
city	Nominal	ADANA,ADIYAMAN,AFYONKARAHÄ°S...

Figure 1: Final Data Set

3.1. Distributions

The target feature is "is_fraud" and "ceated_by_id" represent the record id. Apart from these there are 6 categorical and 16 numerical variables. There are 6825 rows in the final dataset and only 18 of them are flagged as fraudsters. The main issue that we tried to figure out in the scope of this project is the fewness of target data. This will cause many problems in exploring the data and executing the models which we will mention in further chapters.

Before we discuss exploration results, note that koinim.net detects and closes the fradulant customer's accounts manually. Hence all fraudulent rows are also flagged as

"is_active=0". This information will be relevant later in our exploration phase. There are 98 inactive customers, so we will explore 18 fraudsters into 98 inactive customers instead of exploring in 6825 rows. As a result, fraudsters will be more visible in our plots.

We start with some simple bar charts to see the distributions of variables. In Figure 2 we see the relative infrequency of of inactive customers including fraudsters.

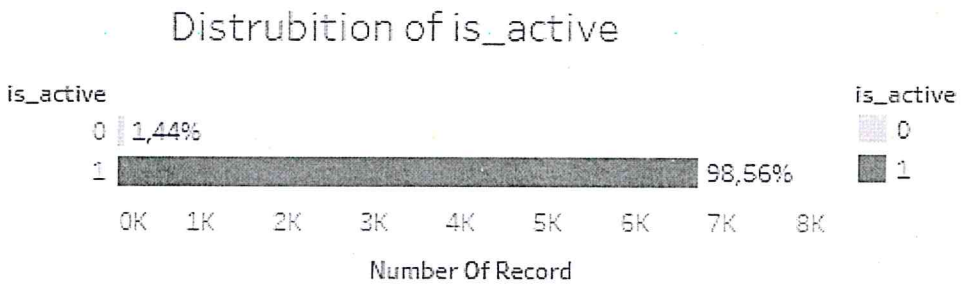

Figure 2: Distrubition of Active Users

1.44% of customers are inactive and 18,4% of those inactive customers are fraudsters. The distrubution can be seen in Figure 3.

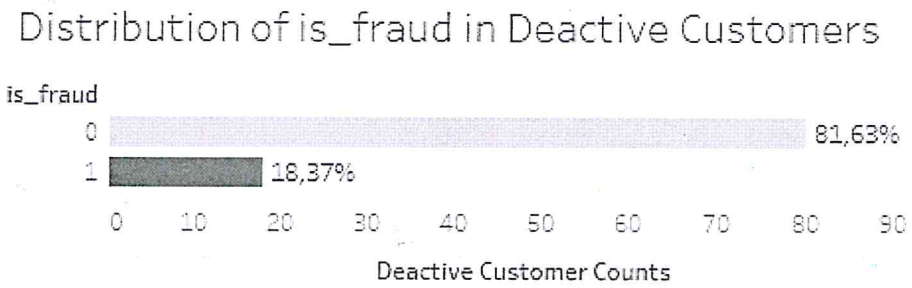

Figure 3: Distrubition of Inactive Users

In Figure 4, customer fraud cases based on their age segment is displayed.

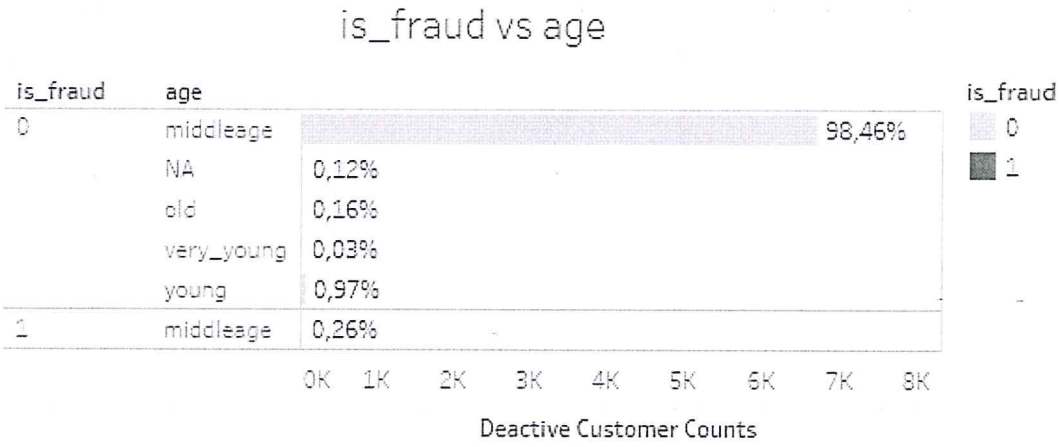

Figure 4: Distrubition of Deactive Users by Age Segment

All fraudsters are in middle age, because more than 98% of customers are in middle-aged segment. This graph only shows that bitcoin and relevant technologies are points of interest for people between age 25 and 46.

The feature tenure is highly correlated with is_fraud feature for inactive customers. As we see in the Figure 5, the max tenure for fraudsters is 15. In fact, this is the expected behavior because we are analyzing hit-and-run types of fraud cases and most customers make their fraudulent transactions not long after they first open their account cycle. As soon as they are detected, the admins of the site end and deactivate the fraudster's accounts. Remember that we already mentioned in previous sections how tenure feature is created, for inactive users deactivation date is the end of their lifecycle. Another insight from this figure is hit-and-run fraudsters mostly make their move in their first 2 days. 10 out of 18 fraud cases are detected and the fraudster accounts are terminated in this manner.

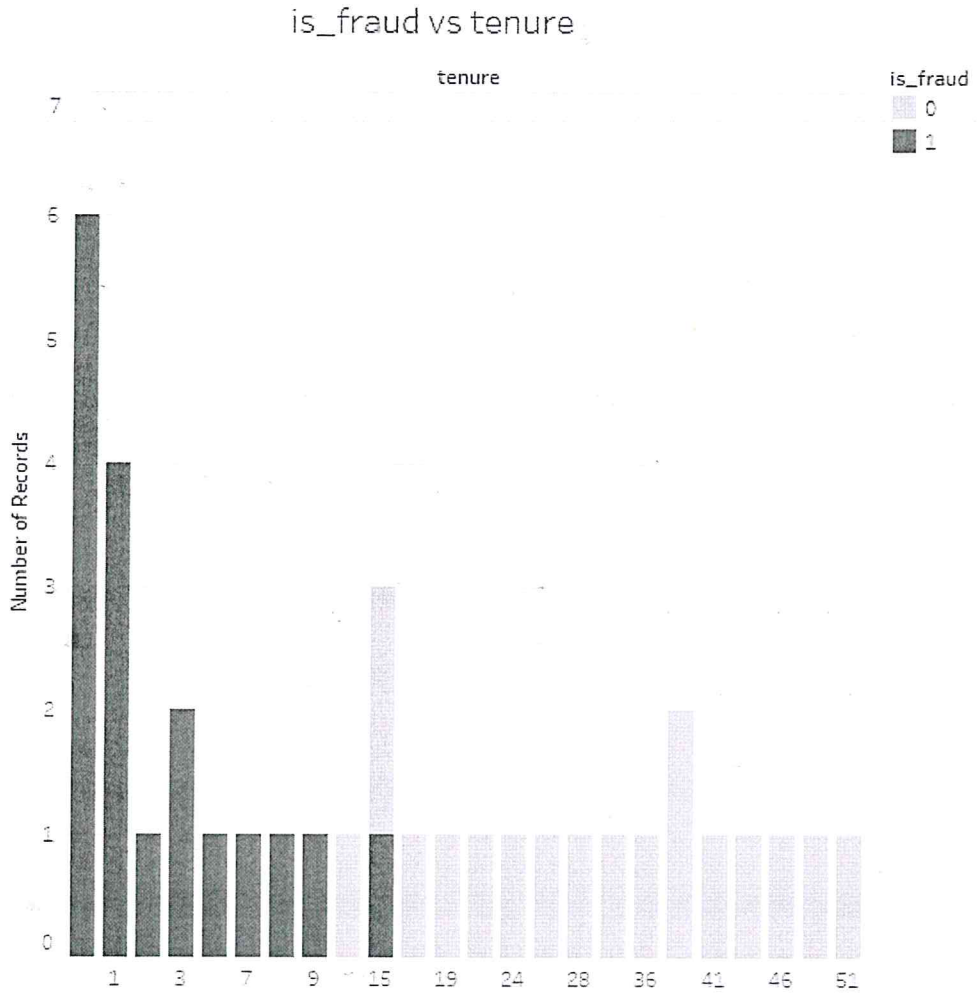

Figure 5: Distrubition of Deactive Users by Tenure

3.1. Clusters

Tenure is quite informative about detecting the fraud cases, but it is a forward-looking variable. That is, it is determined after fraud is detected for fraudsters eventually, hence using this feature in our model will not be correct. We need to detect fraudsters without using tenure to detect fraud cases before they happen.

While we won't use tenure as a predictor in our model, Figure 6 assisted us in investigating customers' first transactions deeply. Next we try to find a pattern by using firts_tx_kurus. It indicates the kuruş volume of the user's first transaction after signing up with the koinim.com site.

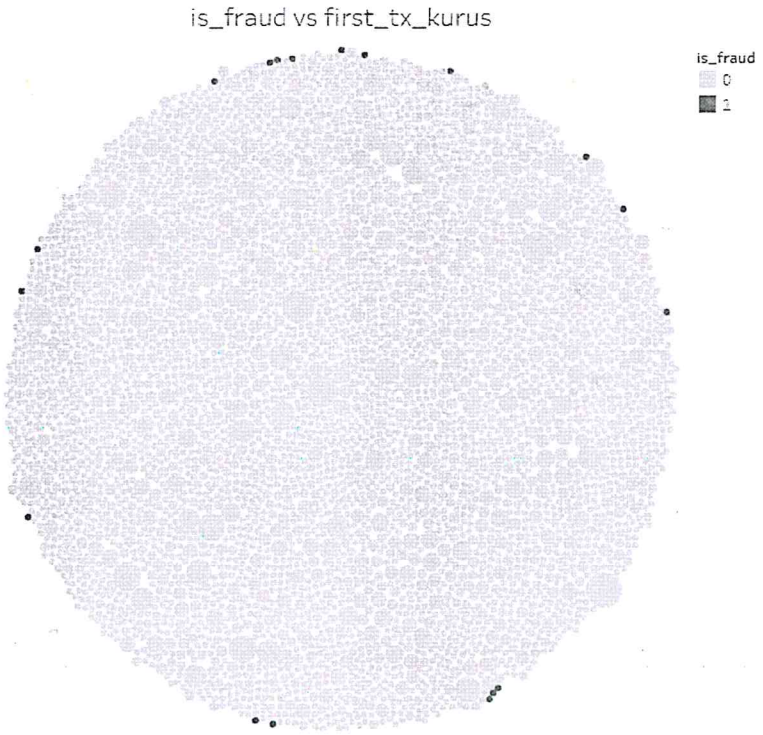

Figure 6: Behavior in first Transaction

In the center of the big circle the first_tx_kurus value is zero, as we moved towards the borders its value is increasing. There is no filter on this figure; every single sample point is represented. We say that all red points are on the peripheral of the bi circle which means hit-and-run fraudsters do transactions with big volumes as expected. This exploration will be a key observation later in model building phase.

Figure 7 looks at the same graph after excluding active users.

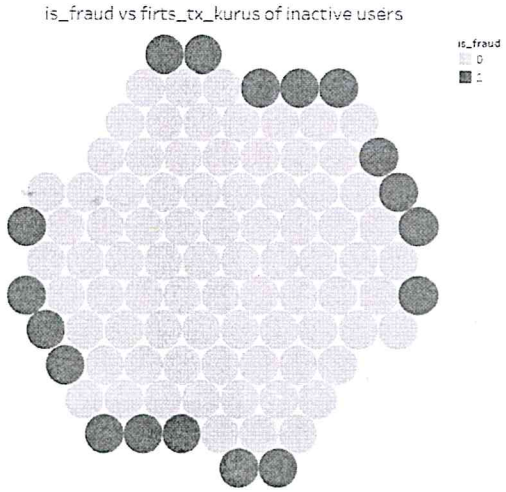

Figure 7: Behavior of Inactive Users in first Transaction

We observe the same behavior in filtered heat map. We can say that fraudsters extremely high volume of first transactions. There are other yellow points in the borders so let's try to face wrap our insight based on first transaction type which is a categorical feature.

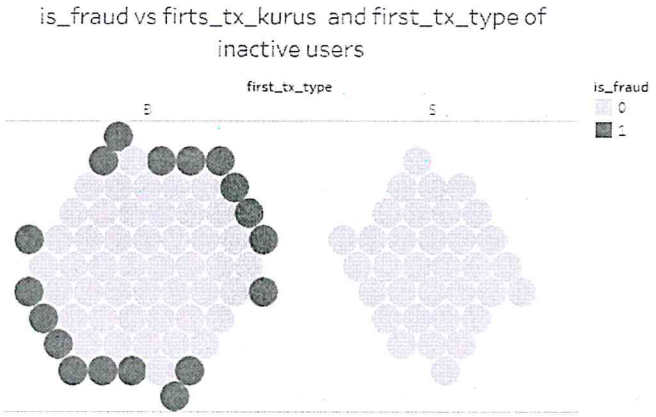

Figure 8: Inactive Users in First Transaction by First Transaction Type

In Figure 8, there is no red point if a customer's first transaction type is "S". That means fraudsters do not give sell order in their first transaction. Probably it is because of they are using some people's money to buy bitcoins. In the left part of the figure there are still yellow points near the boundaries, but we are sure that features related with customers' first transactions are related with their fraud status. At this point we let machine learning algorithms to find out the exact relationship.

4. MODELLING

4.1. Logistic Regression

Logistic regression is a simple and traditional learning algorithm for solving a classification problem. The data contains both numeric and categorical variables with a categorical target. In preparatory section, it is already mentioned that there are use cases of this algorithm in fraud detection according to literature research.

In the first run the predictor feature are selected as number_of_tx_in_first_day, sum_tl_in_first_day, first_tx_in_day, first_tx_type, first_tx_amt, first_tx_kurus, first_tx_currency, avg_trading_volume, id_number_verified, age, tenure. The outcomes of exploration phase are considered in feature selection, but as this is the first model experience for our dataset, we include almost all variables.

.

Case processing summary is given in Table 1

		N	Marginal Percentage
is_fraud	0.0	3363	99,7%
	1.0	11	0,3%
first_tx_type	B	2268	67,2%
	S	1106	32,8%
first_tx_currency	BTC	3171	94,0%
	LTC	203	6,0%
age	middleage	3336	98,9%
	NA	3	0,1%
	old	5	0,1%
	very_young	1	0,0%
	young	29	0,9%
Valid		3374	100,0%
Missing		0	
Total		3374	
Subpopulation		3373[a]	

Table 1: Case Processing Summary for Logistic Regression

Coincidence matrixes will be the most important output criteria during the all modelling phase, because we are trying to solve classification problem and the success criteria will based on how many users are predicted as correct classes they are actually in.

In Figure 9, output values coincidence matrix, confidence report and evaluation metrics are shown.

Results for output field is_fraud

Individual Models

Comparing SL-is_fraud with is_fraud

'Partition'	1_Training		2_Testing	
Correct	3,374	100%	3,447	99.88%
Wrong	0	0%	4	0.12%
Total	3,374		3,451	

Coincidence Matrix for SL-is_fraud (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,363	0
1.000000	0	11

'Partition' = 2_Testing	0.000000	1.000000
0.000000	3,440	4
1.000000	0	7

Confidence Values Report for SLP-is_fraud

'Partition' = 1_Training: SL-is_fraud always correct. No confidence report

'Partition' = 2_Testing	
Range	0.78 - 1.0
Mean Correct	1.0
Mean Incorrect	0.93
Always Correct Above	1.0 (0% of cases)
Always Incorrect Below	0.78 (0% of cases)
99.88% Accuracy Above	0.0
2.0 Fold Correct Above	1.0 (0% of cases)

Evaluation Metrics

'Partition'	1_Training		2_Testing	
Model	AUC	Gini	AUC	Gini
$L-is_fraud	1.0	1.0	0.999	0.999

Figure 9: Outcomes for First Try in Logistic Regression

The overall accuracy seems perfect within this model, but in fraud detection cases the main problem is that fraudulent customer number is too less. In order to have a concrete idea, type I and type II errors should be analyzed.

The training set provides a model with zero error, in testing phase the model successfully detects all fraudulent costumer (0 type I error) and there is only 4 type I errors which means 4 customers are flagged as fraudulent while they are not.

The result look good, but we have a problem about the predictor feature tenure. For now, fraud cases are being detected manually and fraudulent customers are being kicked as soon as they are detected. Most fraud cases happen in the right after customer account opening, the tenure is less the 15 days for all fraudulent customers. So let's check the predictor importance matrix and see if tenure is an effective feature;

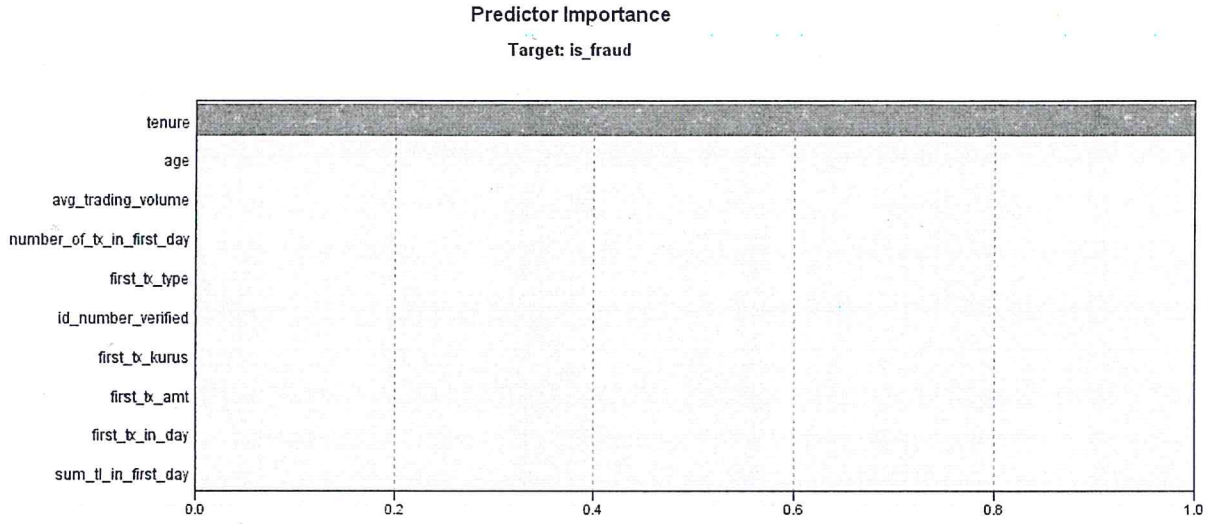

Figure 10: Predictor Importance for First Try in Logistic Regression

We in Figure 10 that the model owes its success to tenure feature which is generated after detection for real customers. We need to see if the model has the predictive power without using tenure.

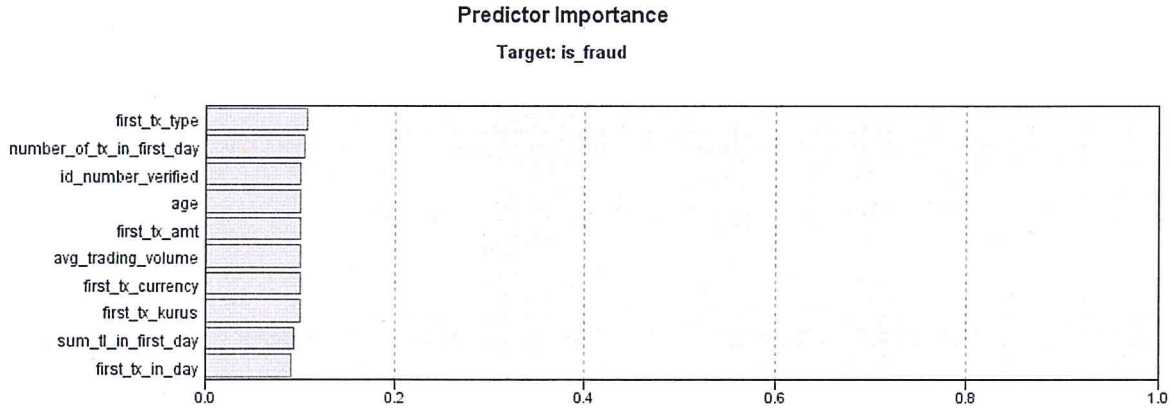

Figure 11: Predictor Importance Without Tenure Try in Logistic Regression

Figure 11 shows the predictor importance table without success. It is more evenly distributed now. We look at the confidence matrix to see if we have similar predictive rate in Figure 12.

Results for output field is_fraud

Individual Models

Comparing SL-is_fraud with is_fraud

'Partition'	1_Training		2_Testing	
Correct	3,364	99.7%	3,442	99.74%
Wrong	10	0.3%	9	0.26%
Total	3,374		3,451	

Coincidence Matrix for SL-is_fraud (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,362	1
1.000000	9	2

'Partition' = 2_Testing	0.000000	1.000000
0.000000	3,442	2
1.000000	7	0

Figure 12: Outcomes of Logistic Regression Without Tenure

The overall accuracy is still above 99 % but that does not means the model is successful. The success of accuracy comes from detecting non-fraudulent customers. But our focus is to find fraudulent cases, that is the model sensitiviy. And the model failed to find any fraud cases in test set.

We rerun model with different methods such as stepwise, forwards, backwards and backwards stepwise. We can say that, by excluding the most meaningful numeric variable (tenure) from the model we broke the logistic regression algorithm. This result shows that we should try other modeling algorithms that can handle numeric and categorical variables together.

4.2. Classification and Regression Trees (CART)

CART is successful to handle outliers. The disadvantages of CART is that it can be split on a single variable but we have a single target feature in our case which means we can use CART without being affected from that issue.

When we run CART with default parameters the stopping rules of CART prevent any tree growth. The reason is that we have 6825 sample points and for only 18 fraud cases (is_fraud = "1") which comprises of 0.26 % of observations. Similarly, in the training set there are 3364 observations and fraud cases make 0.33 % of the trainin set. The default values are 2% for "minimum records in parent branch" and 1% for "minimum records in child branch". Since there are only 18 fraud cases, we should tune these 2 parameters to make possible a single leaf may contain a single sample. Minimum number of records in

the child branch should be small enough to fulfill that condition. So let's values given in Table 2 for these parameters;

Minimum records in parent branch(%):	0.2
Minimum records in child branch(%):	0.1

Table 2: CART Parameters

According to the coincidence matrix given in Figure 13, there are some false positive errors but no false negative errors. We can say that we have a reasonably good result with %99.94 overall accuracy.

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,361	2
1.000000	0	11
'Partition' = 2_Testing	**0.000000**	**1.000000**
0.000000	3,441	3
1.000000	0	7

Figure 13: Coincidence Matrix For CART

The result seems successful but we may overfitting problem because of the tenure feature. Next we review the the tree construction shown in Figure 14.

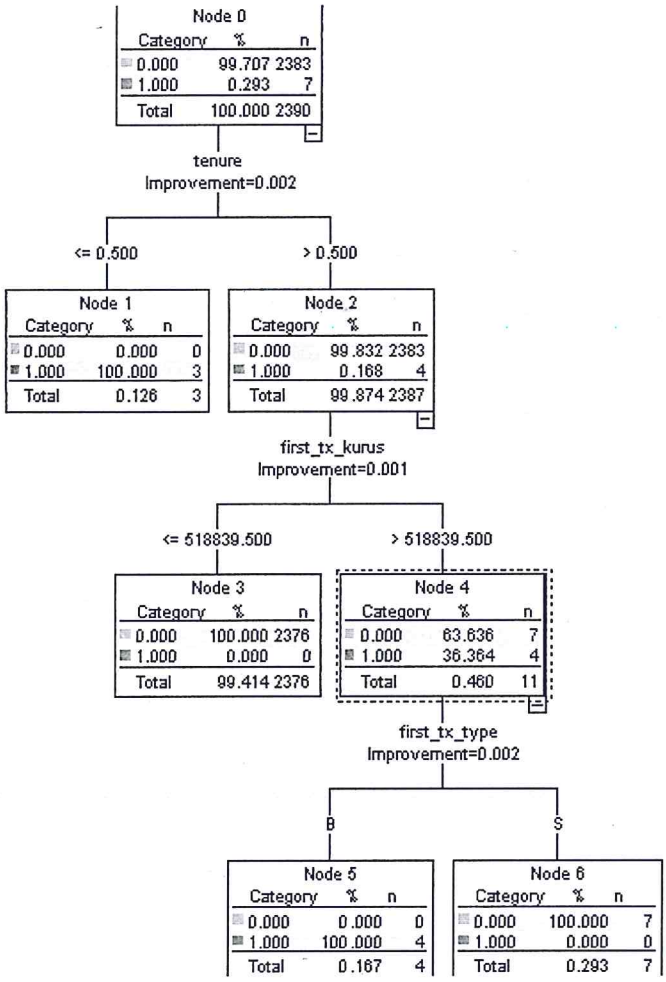

Figure 14: Tree Construction For CART

Tenure is the first decision point of the tree as expected. It is not the most important feature as shown in Figure 15, but is still included in our model..

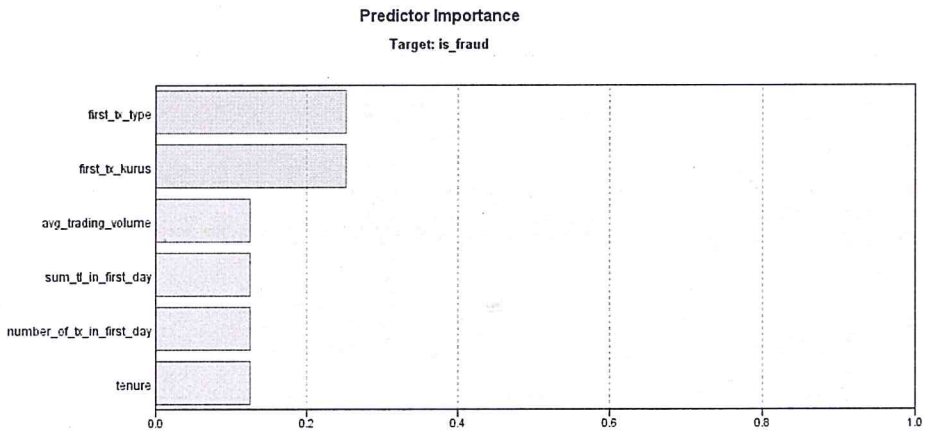

Figure 15: Predictor Importance with Tenure

Good news is that there 3 other features with same importance as tenure which does not appear in the tree structure. We are able to obtain good results after excluding tenure. We build a model without tenure and by tuning parameters as given in Figure 16.

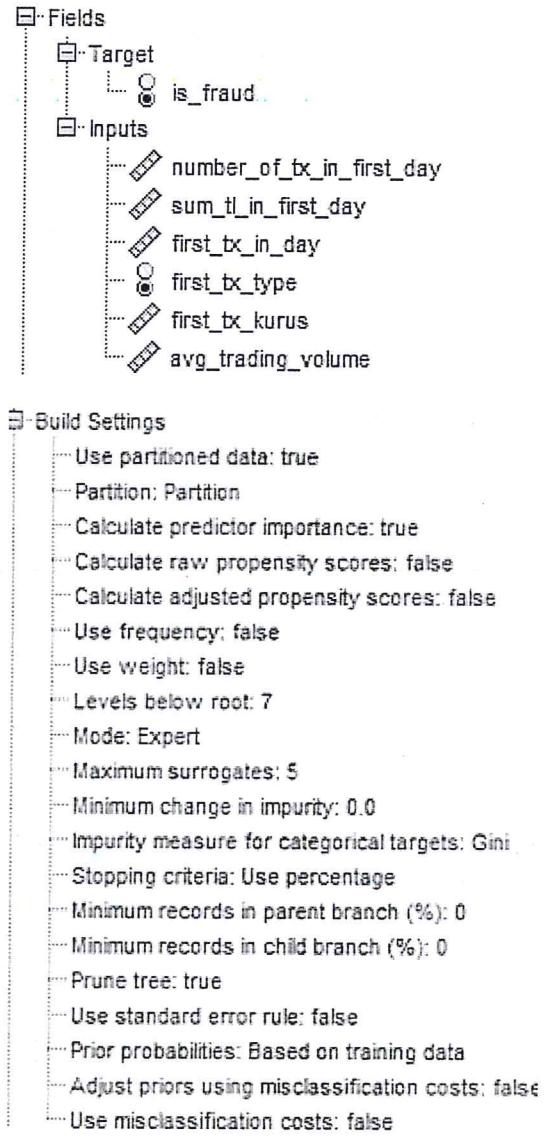

Figure 16: Input Parameters For CART

The most predictive model parameters from previous model runs with tenure are also included as well. The only change is that we excluded tenure as an additional predictor.

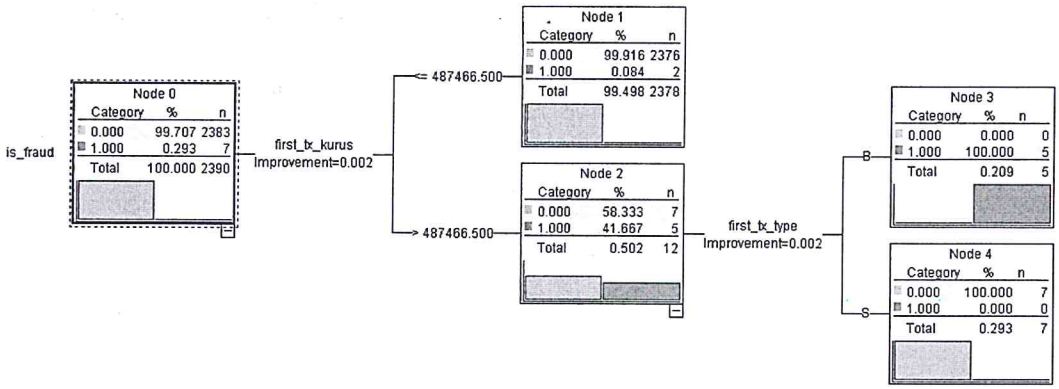

Figure 17: Tree Construction For CART without Tenure

The bar chart presentation of tree structure provides brief conclusions about the results of the model. We know that we have there is 7 fraudulent costumers in our training set and only 5 of them are determined in 100% pure leaf. The result seems reasonable but there are still errors. We need to check some other output assets to realize what is missing and how can we fix it.

Coincidence Matrix for SR-is_fraud (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,362	1
1.000000	2	9
'Partition' = 2_Testing	**0.000000**	**1.000000**
0.000000	3,442	2
1.000000	0	7

Figure 18: Coincidence Matrix For CART without Tenure

There are some unexpected results in the coincidence matrix. Although there are 2 false negative errors in model built on training set, there are no missing fraudulent costumers when building model with training data. Analyzing the matrix for training partition we can say the overall accuracy is quite high (over 99%). The result is reasonable for detecting non-fraudulent customers; there is only 1 mistake for more than 3,000 samples. But model failed to detect 2 fraudulent cases of 11 in total, which means the accuracy for fraudulent samples is about **81.8%.** This success could be a coincidence. In tree structure it is obviously clear that in second step there is an impure leaf with 2376 non-fraudulent and 2 fraudulent samples. The purity ratio is high but still not 100%. The problem in the model can be due to that observation.

CART algorithm has a parameter called "minimum change in impurity" which can tune the stop condition. By reducing this value it would be possible to create new leafs from node 2, but we already set that parameter to the minimum value and the purity is still too high to let the tree growth.

The default value of "impurity measure for categorical targets" is "GINI". We also tried to build models with two other methods which are "ordered" and "twoing". But the result didn't change. Assuming these are the best results that we can obtain with CART, we look at the output metrics of the model beginning with predictor importance graph in Figure 19.

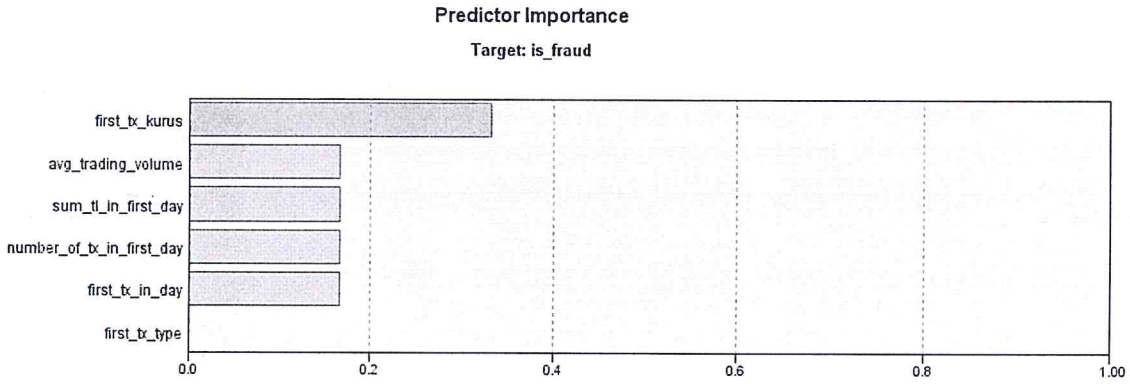

Figure 19: Predictor Inportance of CART Model

According to the model the most important feature is the kuruş volume in the first transaction. Generally, customers invest low amount of money in their first days until they experiment with the website. However, those customer who invest bigger amounts in their very first day are risky regarding their fraud potential.

Another crucial output is the comparison table for actual and predicted values. Bear in mind that, since our target variable has too few positive values, the overall accuracy does not inform us about the success of the model.

'Partition'	1_Training		2_Testing	
Correct	3,371	99.91%	3,449	99.94%
Wrong	3	0.09%	2	0.06%
Total	3,374		3,451	

Figure 20: Summary for CART Model

Such as in our example, the overall accuracy is 99% but the accuracy for fraudulent samples was calculated as 81.8% in previous sections. We also report confidence values in the report below.

'Partition' = 1_Training	
Range	0.857 - 0.999
Mean Correct	0.998
Mean Incorrect	0.952
Always Correct Above	0.999 (0% of cases)
Always Incorrect Below	0.857 (0% of cases)
99.91% Accuracy Above	0.0
2.0 Fold Correct Above	0.999 (0% of cases)

'Partition' = 2_Testing	
Range	0.857 - 0.999
Mean Correct	0.998
Mean Incorrect	0.857
Always Correct Above	0.857 (99.74% of cases)
Always Incorrect Below	0.857 (0% of cases)
99.94% Accuracy Above	0.0
2.0 Fold Correct Above	0.857 (100% of cases)

Figure 21: Confidence Report for CART Model

6698 samples of total 6779 observations are calculated with almost 100% confidence. Values for 26 samples are calculated with 89% confidence and finally 19 of them are scored with 86% confidence.

4.3. Random Forest

We have run one regression and one tree algorithm so far. CART is a tree algorithm that uses classification and regression together. CART gave a better result than Logistic Regression. We can say that the accuracy value we obtained from CART (80%) is quite enough for fraud detection problems, yet it is can still be improved.

Since it is observed good result with a standard tree algorithm, I believe we can improve the accuracy score with Random Forest which is an advance tree algorithm combining several tree results.

First of all we run our model with all features including "is_active" and "tenure" features. As a characteristic of Random Forest the algorithm will try different decision rules. Top 5 decision rules are given at below table.

Top Decision Rules for 'is_fraud'

Decision Rule	Most Frequent Category	Rule Accuracy	Ensemble Accuracy	Interestingness Index
(first_tx_in_day > 1.0) & (first_tx_kurus > 283000.0)	0.0	1.000	1.000	1.000
(first_tx_kurus <= 283000.0)	0.0	1.000	1.000	1.000
(sum_tl_in_first_day <= 5161.0) & (first_tx_type = {B}) & (first_tx_amt > 0.7413295)	0.0	1.000	1.000	1.000
(first_tx_type = {S}) & (first_tx_amt > 0.7413295)	0.0	1.000	1.000	1.000
(first_tx_in_day > 1.0) & (tenure <= 98.0)	0.0	1.000	1.000	1.000

Table 3: Top 5 Rule for Non-Fraud Cases

All rules give perfect accuracies. We wonder whether this is an over fitting issue. We may have the same problems that in the CART and LR algorithms. We need a deeper investigation of this issue. Only one of the rules includes tenure and other 4 rules also gives the same perfect accuracy. Excluding tenure may have a small effect on the accuracy but not a dramatic one as we experienced with LR. We look at the confidence table below to see the real effect of "tenure" and if overall accuracy represents the figure for type I and type II errors as well.

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,361	2
1.000000	0	11
'Partition' = 2_Testing	**0.000000**	**1.000000**
0.000000	3,441	3
1.000000	1	6

Figure 22: Coincidence Matrix with Tenure Included

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,363	0
1.000000	1	10
'Partition' = 2_Testing	**0.000000**	**1.000000**
0.000000	3,444	0
1.000000	2	5

Figure 23: Coincidence Matrix with Tenure Excluded

Figure 22 and Figure 23 emphasize two points. First, excluding tenure does not affect the overall accuracy, but increases false negative errors and minimize false positive errors. Second, we don't observe any dramatic decrease in accuracy which means we still have a good working model without "tenure". The input parameters for tuned Random Forest model (tenure excluded) are given in figure 24.

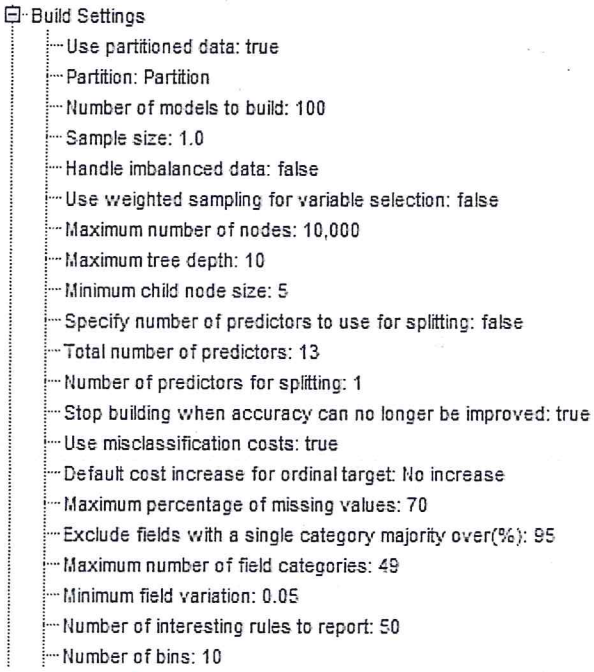

Figure 24: Build Settings for Random Forest Model

And predictor fields are given in figure 25.

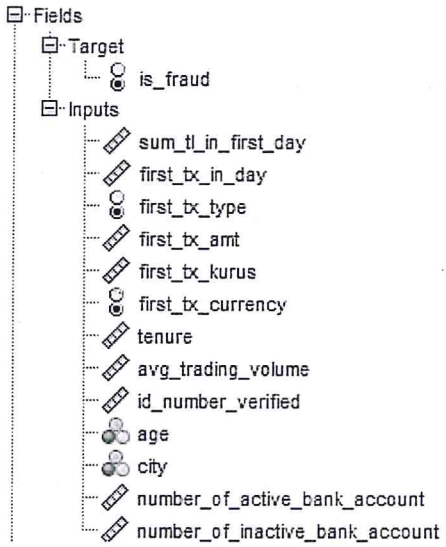

Figure 25: Field Settings for Random Forest Model

Another capability of RF is that the decision rules for different categories of target feature may differ. This is what we see in our analysis. Most frequently occuring category values were 0 for all top 5 decision rules. This is an expected outcome because more than 99% of our dataset is non-fraudulent customers. Next we check the rule for fraudulent cases.

Top Decision Rules for Target Category '1.0'

Decision Rule	Most Frequent Category	Rule Accuracy	Ensemble Accuracy	Interestingness Index
(first_tx_in_day <= 0.0) & (first_tx_currency = {BTC}) & (first_tx_in_day <= 1.0) & (first_tx_amt > 0.7413295)	1.0	0.063	0.984	0.005

Table 4: Decision Rule for Fraud Cases

There is a single rule to determine class 1. According to RF all fraudulent customers generate transactions more than 0.74 digital coins (could be bitcoin or light coin) in their first 2 days after activation. If it is their first day, then they create transaction for bitcoin but not light coin. "first_tx_in_day" represents how many days after activation a customer makes his first transaction.

Rule accuracy seems very low but this output metric explains how often the rule is executed and as our class 1 has a very low ratio (0,2%) in the overall data that is the expected behavior. Also ensemble accuracy is very high. Ensemble accuracy can be stated as the accuracy value of the class. Therefore, we argue that the model gives highly better prediction results for fraudulent cases.

As we see in Figure 24 we have only 2 incorrect predictions for our test data as in Coincidence matrix shows. Although results look good, missing 2 out of 7 fraudulent cases means that model still can be improved.

$RC-is_fraud is an output metric that indicates the confidence level of the prediction. Confidence intervals for both testing and training sets are shown in upper figure. "Always Correct Above" and "Always Incorrect Below" metrics also shows that, when the model is incorrect it indicates a prediction with low confidence. Thus, by just looking at the confidence levels of a row, it is possible to find or guess possible mistakes, after deploying to live environments. Next we look at an example from the output. There

are only 3 predictions with confidence level below 0.667. They are all false positive cases, namely fraudsters as predicted by the model but non-fraudsters actually. Either the model is incorrect or these customers are potential fraudsters but somehow did not commit fraud..

Results for output field is_fraud

Individual Models

Comparing SR-is_fraud with is_fraud

'Partition'	1_Training		2_Testing	
Correct	3,373	99.97%	3,449	99.94%
Wrong	1	0.03%	2	0.06%
Total	3,374		3,451	

Coincidence Matrix for SR-is_fraud (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,363	0
1.000000	1	10

'Partition' = 2_Testing	0.000000	1.000000
0.000000	3,444	0
1.000000	2	5

Confidence Values Report for SRC-is_fraud

'Partition' = 1_Training	
Range	0.556 - 1.0
Mean Correct	0.999
Mean Incorrect	0.556
Always Correct Above	0.556 (99.97% of cases)
Always Incorrect Below	0.667 (0.03% of cases)
99.97% Accuracy Above	0.0
2.0 Fold Correct Above	0.556 (100% of cases)

'Partition' = 2_Testing	
Range	0.667 - 1.0
Mean Correct	0.999
Mean Incorrect	0.778
Always Correct Above	0.889 (99.39% of cases)
Always Incorrect Below	0.667 (0% of cases)
99.94% Accuracy Above	0.0
2.0 Fold Correct Above	0.889 (100% of cases)

Evaluation Metrics

'Partition'	1_Training		2_Testing	
Model	AUC	Gini	AUC	Gini
$R-is_fraud	1.0	1.0	0.999	0.999

Figure 26: Summary of Random Forest Model

Although there is 1 false negative case for 11 sample points in the training set, there is 2 false negatives for for 7 sample points in the testing set. So the accuracy of testing is 71% while it was 83% for training set. This could have been caused by an overfitting situation due to using highly correlated features together into the model.

The paramaters "sum_tl_in_first_day" and "first_tx_amount" are highly correlated because when a customer makes a high volume transaction in his first transaction that directly should be added to the sum of his total TL volume in that day. The number of active / inactive bank account features are not related with the issue, because those are not

real demographic information, instead they are some information stored in the site based on customer declaration. Next we investigate when we throw all those features into the model. We will use the same tuned parameters as previous runs. Figure 27 shows the predictors.

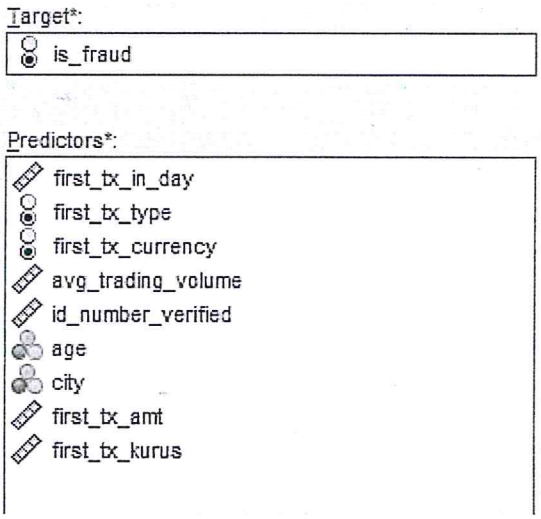

Figure 27: Final Predictors of Random Forest Model

The new coincidence matrix in Figure 28 show that we totally got rid of false negatives.

Coincidence Matrix for SR-is_fraud (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	3,360	3
1.000000	0	11
'Partition' = 2_Testing	**0.000000**	**1.000000**
0.000000	3,441	3
1.000000	0	7

Figure 28: Final Results of Random Forest Model

We finally succeeded to find all fraudsters without leakage and only flagged 6 non-fraudulent customers as fraudsters in both training and testing sets. The overall accuracy is 99% to detect fraudulent costumers.

5. CONCLUSIONS

5.1. Summary of Contributions

Fraud detection is an unbalanced class distributed problem. The low number of fraudulent class compared to the sampe of all transactions constitutes a problem in fraud detections. There are several methods to deal with this problem. Some of them are analyzed and applied to the models run our dataset As a result, we can say that decision based algorithms have higher accuracy to predict the fraudsters than regression based algorithms.

Logistic regression failed to make good predictions after excluding a certain predictor feature called tenure. Tree based algorithms improve the prediction results by tuning the model.

Finally, we can say that the best result is provided by random forest which is an advanced and recently developed tree based machine learning algorithm. According to the RF model, if a user makes transaction in the first 2 days after signing up to the system and submits a buy order for more than 0.74 bitcoin, this is potentially a fraudulent transaction. This result makes sense because most bitcoin traders first try to experiment with the web site and then decide whether the site is safe for transactions or not. During the first 10 days after signing up the user explores the features of the platform and gains experience about the bitcoin market. When a user submits a large transaction at a very short time after the signup he may be spending someone else's money or the user may be forced to buy some bitcoins by blackmailing. Another insight that the results provides is that the lightcoin users have been have never commited fraud by now. No fraudulent cases have been observed in sell transactions for bitcoin.

5.2. Added Value for Koinim.com

With this study the company can have a better fraudsters what are the conditions to name a transaction as fraudulent. There are several actions to be taken to prevent or detect fraudulent behaviors. For initial subscriptions there could be limits per days or transactions for everybody. The model outputs can direct the company about what that limit should be and when it should be increased for each user.

5.3. Future Objective

This project carries out a user based analysis to detect fraudulent Bitcoin transactions. Transactional data of users are processed with analytical models. We find that transactional behaviors can help in detecting the fraudsters. As a next step, we analyze transactions directly and a transaction can be marked as risky as soon as it happens. Then several precautionary actions such as blocking or parking the transaction can be executed. To implement such a solution in the future, the company should further develop the source systems that creates the user data. The raw data can be processed real or near real time. Streaming and complex event processing techniques should be investigated for that purpose.

DATA DICTIONARY

No	Column Name	Role	Description
1	avg_and_first_ratio	predictor	Ratio between the average volume of all transactions and volume of first transaction
2	number_of_tx_in_first_day	predictor	Number of Transactions in the day when the first transaction is done
3	sum_tl_in_first_day	predictor	Total money volume of the day when the first transaction is done
4	created_by_id	recırd_id	Indicates the user id
5	first_tx_date	predictor	Date of first transaction
6	first_tx_in_day	predictor	Indicates how long it took to make a transaction after signing in to to site
7	first_tx_type	predictor	Indicates the first transaction type of user. Buy (B) / Sell (S)
8	first_tx_amt	predictor	Indcates the Bitcoin or Ligthcoin amount of first transacion
9	first_tx_kurus	predictor	Indicates the money amount in kuruş currency of the first transaction
10	first_tx_currency	predictor	Indicates the digial currency type of first transaction. Bitcoin(BTC) or Lightcoin(LTC)
11	deactivation_date	predictor	Date of deactivation
12	tenure	used in exploration not used in modelling	Total number of days that a user stayed active in the site. This feature is nıt used in to modelling because after this feature is stated to its final value after fraud happens.
13	is_fraud	target	1 for fraudsters 0 for clean users.
14	avg_trading_volume	predictor	Average money volume in kuruş per transaction
15	is_active	used in exploration not used in modelling	active status of users. 0 for deactive, 1 for active.
16	id_number_verified	predictor	indacates if the id number is verified
17	age	predictor	age
18	city	predictor	location
19	number_of_active_bank_account	predictor	number of active bank account registered in the site
20	number_of_inactive_bank_account	predictor	number of inactive bank account registered in the site
21	total_bitcoin_buy_order_cnt	predictor	total bitcoin buy transaction count
22	total_light_coin_buy_order_cnt	predictor	total lightcoin buy transaction count
23	total_light_coin_sell_order_cnt	predictor	total lightcoin sell transaction count
24	total_bitcoin_sell_order_cnt	predictor	total bitcoin sell transaction count

REFERENCES

Sahin, Y., Bulkan S. And Duman E. "A Cost-Sensitive Decision Tree Approach for Fraud Detection." *Expert Systems with Applications*, vol. 40, no. 15, 2013, pp. 5916–5923., doi:10.1016/j.eswa.2013.05.021.

Pozzolo, A. D.(2015). Adaptive Machine Learning for Credit Card Fraud Detection (Unpublished doctoral thesis). Université Libre De, Brusels, Belgium

Jha, S., Guillen, M. and Westlandl,J.C. "Employing Transaction Aggregation Strategy to Detect Credit Card Fraud." *Expert Systems with Applications*, vol. 39, no. 16, 2012, pp. 12650–12657., doi:10.1016/j.eswa.2012.05.018.

Mahmoudi, N., & Duman, E. "Detecting Credit Card Fraud by Modified Fisher Discriminant Analysis." *Expert Systems with Applications: An International Journal*, Pergamon Press, Inc., dl.acm.org/citation.cfm?id=2803265.

Richard J Bolton and David J Hand. "Unsupervised profiling methods for fraud Detection". *Credit Scoring and Credit Control* VII, pages 235–255, 2001.