View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Aaltodoc Publication Archive

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Syed Azeem Akhter

Using machine learning to predict potential online gambling addicts.

Master's Thesis
Espoo, October 2, 2017

Supervisor: Professor Aristides Gionis
Advisor: Markku Mantere

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: English	Syed Azeem Akhter		
Title: Using machine learning to predict potential online gambling addicts.			
Date:	October 2, 2017	**Pages:**	45
Major:	Computer Science (Big Data and Large-Scale Computing)	**Code:**	T-79
Supervisor:	Professor Aristides Gionis		
Advisor:	Markku Mantere		

Betting addicts on the gambling websites are difficult to identify because online gambling is by nature different from real gambling. This thesis attempts to identify potential gambling addicts in an online gambling website X using machine learning models. The models are based on user's usage history on the website. The usage data is collected for each user from the site using JavaScript. The data is then analyzed and stored in a database. Machine learning models are then trained using Support Vector Machines with the data of users who are by definition problem gamblers. The system then makes a prediction for all active users based on their recent usage history. The final results include an automated system for daily learning and prediction of potential problem gamblers who show early signs of gambling addiction.

Keywords:	machine learning, classification, pathelogical gambling, gambling addiction
Language:	English

Acknowledgements

I would like to thank my thesis supervisor Aristides Gionis for continiously steering me in the right direction during the course of this thesis. It has been an utmost pleasure working under his supervision.

I would also like to thank Markku Mentere and Joni Trunen from Frosmo Oy for allowing me to work on this thesis topic and providing support in the work done for this thesis.

Lastly, I must express my profound gratidue to my parents and the rest of my family, my spouse, Sarah Haider, and my friends in Finland and Pakistan for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Helsinki, October 2, 2017

Syed Azeem Akhter

Contents

Chapter 1

Introduction

1.1 Motivation

Gambling addiction or more formally called problem gambling or pathological gambling is an old impulse control disorder with many side effects. Online gambling is diagnosed similarly as regular gambling problem at the casinos but it has its own problems, which are not present or can be easily diagnosed in the regular gambling.

For example if someone plays for long hours regularly at casinos people will start to notice their excessive presence and consider that as a sign of addiction. However, with online gambling people can play using their computer and mobile phone with an Internet connection wherever they are which makes it much more addictive than regular gambling without anyone noticing them. Also with no control over the environment where gambling is done online, people can play under the influence of drugs or alcohol and end up spending much more money.

With these unique problems of online gambling in sight it is very important to identify problem gamblers on the gambling sites and also early identify those users who have a tendency to be problem gamblers later on.

Countries where online gambling is completely regularized, gambling companies are required to have information about responsible gaming on the website/application. They have to allow users to control their gambling habits from a separate section. The online gambling company X where the data for this research is collected from also has a separate responsible gambling section. Users can limit their gambling habits in multiple ways. It allow users to have deposit limits, get reality check popups when they have gambled for some time and self exclude from the site after which they are not able to use the site for certain duration or permanently. Despite these features for re-

sponsible gambling, users show addictive behavior on the site, some without using the responsible gaming feature at all.

With these in mind it is very important to find out problem gamblers and potential problem gamblers on the betting site so that they can be informed about the addiction and given help if needed.

1.2 Main Contributions and Research Question

This thesis tries to answer the question whether it is possible to find potential gambling addicts on a high traffic gambling site using predictive analysis machine learning methods. The work done as part of this thesis include tracking user data, building a data pipeline for storing and aggregating user data. Machine learning methods were later used to classify users as addicts and non addicts based on their usage history on the site over time. The thesis was done as a special project in Frosmo, a software company based in Helsinki, Finland. Frosmo tools were used in tracking user data as well as technical guidance from Frosmo's senior developers for tackling scalability problems.

1.3 Why Machine Learning?

Finding problem gamblers is a deterministic task. People who play more than a threshold amount or spend more than a threshold time can be considered problem gamblers on the site. With the meaning of problem gamblers already defined it is easy to find those from the usage data. The bigger problem is the early identification of users who might become problem gamblers later on or in other words those users who have not passed the threshold yet but still have an overall profile of gamblers.

We have used the usage history for those addicted by definition for training the algorithm for the classification task. The classifier is then used to predict all the users based on their usage history for last month. The data for model creation is collected and summarized monthly for model creation and the predictor predicts everyday.

1.4 Structure of the Thesis

Chapter 2 reviews the relevant literature related to the potential gambling and methods used in the implementation of the system.

In order to predict problem gamblers on the site We collected the usage data related to their gambling habits of all users on site X over 4 months period with over 30000 users. The exact parameters which are tracked and how they are tracked are defined first briefly in Chapter 3 and in more detail in Section 1 of Chapter 4.

Tracking, cleaning, analyzing and storing data for a high traffic site as X needs to be both efficient and scalable. Section 3 and 4 of Chapter 4 defines how the problems of scalability and eficciency were solved through a custom log analyzer and using an indexed relational database. Furthermore, Section 4.4 has a detailed information about data analysis and visualization.

We tried Naive Bayes and Support Vector Machines (SVM) algorithms for prediction but found linear SVM to have the highest accuraccy, precision and recall from the usage data. Chapter 4 has the detailed implementation of feature engineering, machine learning modeling and evaluation of different models.

The result section has a brief summary of the work done, results found from data analysis and predictive analysis system, and recommendations for the future work for solving this or similar problems.

Chapter 2

Literature Review

2.1 Addiction

Aviel Goodman et al. defined addiction as a behaviour that produces pleasure and a relief from internal discomfort. It usually comes with a pattern that is characterised by constant failure to control the behavior and continuation of the behavior despite experiencing negative effects [9]. Several other researchers have defined addiction similarly as a harmful behaviour which is hard to hard to stop engaging in [10]. Addiction causes medical, social and psychological harm. The fact that it hampers an individual's ability to make a decision and violates their freedom of choice, means that its appropriate to consider it a psychiatric disorder.

Addiction typically involves initial exposure to a stimulus followed by behaviors seeking to repeat the experience. After a number of repetitions of the behaviour-stimulus sequence, the addiction becomes established. The character and severity of the addiction may change over time, and it may be punctuated by attempts by the sufferer to abstain or regain control in some cases, sufferers will achieve recovery for a sustained period or even permanently [20].

2.2 Online Gambling

Multiple studies have assessed the prevalence of gambling on the internet. Online gambling in UK has increased from 1% in 2001 to 7% in 2007 [13]. Another study done by Griffiths & Wood, in 2007 suggests that 8% of UK kids aged 12-15 years play national lottery online [18].

To summarize, different studies have shown different results but researchers agree that Online gambling is on the rise and in general 2-6% of the popula-

tion gambles online [13].

2.2.1 Gambling Addiction

The prevalence of gambling addiction is 2-3% [5]. The essential feature of gambling addiction is recurrent and persistent gambling behaviour. It results in disruptive personal, family or work life [5]. Aviel Goodman, has listed the top characteristics of gambling addicts which include frequent gambling, frequent gambling of larger amount of money, betting for a longer amount of time than intended, needing to increase the amount and size of bets than intended, restlessness if unable to gamble etc [9].

Daria J et al argued that online gambling addicts have the same kind of characteristics as normal addicts [13]. The prevalence of gambling addiction has increased a great deal over the time period.

2.3 Tracking User Data

Modern web applications use sophisticated user interfaces as compared to in the past which was simple and straight forward. Additionally, more applications are using clientside frontend for application logic and use server to load and save data. Earlier web applications strictly followed the HTTP procotol's request/response paradigm. These developments pose a problem when it comes to obtaining feedback about the usage of web applications [6].

The data left by many interactive applications in the server's log file is minimal and not sufficient for extracting detailed information about the actual usage of the application. For example, how much time each user spent on each webpage or in what order a web form was filled.

For these reason we need more sophisticated methods of tracking user data. There are many commercial tools available for tracking user's interaction on the website and visualizing the data in multiple different forms. HotJar, google analytics etc are the most commonly used tools for this purpose.

2.3.1 Frosmo Tools

Frosmo is a startup based in Helsinki Finland. Their core focus is improving UX/UI of ecommerce websites for better conversions and more [3]. They have several in house tools for ecommerce data tracking and recommendations. Frosmo ask their customers to insert a script tag to the html body and their

developers modify scripts in the script tag to interact with their customer websites using JavaScript.

Since the work done for this thesis was for one of Frosmo customers, the data tracking from the website was done with javascript using Frosmo tools. They tracks products on their customers websites for product recommendations and other things. We modified the product tracking with the guidance of Frosmo's core product team to track user behavior data on the online betting website X.

2.4 Machine Learning

The formal definition of Machine learning according to Thomas M. Mitchell, 2007 [14].

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience T."

In laymen's terms Machine learning is the ability of a computer to learn from experiences (data) and generalize the solution over all kinds of datasets. Machine learning applications are useful for non-deterministic problems where the problems are either too complex or too big to be broken down in to step by step instructions. Some machine learning applications are spam detection, predicting stock values, finding patterns or hidden structures in a structured data set etc. The method of learning from previous examples or experiences is a sub field of Artificial Intelligence called Supervised learning.

The two main applications of supervised machine learning are classification and regression. Classification is used to predict which class a record belong to whereas regression is used to predict a real numerical value.

Machine learning can also be used to reveal a hidden class structure in unstructured data, or it can be used to find dependencies in a structured data to make predictions. We have used the latter for predicting potential gambling addicts on the gambling site X.

2.5 Predictive Analysis

Predictive analytics is the act of predicting future events and behaviors present in previously unseen data, using a model built from similar past data [15]. It has applications in wide variety of domains such as finance, healthcare, academics and gambling industry etc [4]. The application and methods of predictive analysis in all these fields is similar. A machine learn-

ing algorithm finds the relation between different properties of the data to build a model. The resulting model is able to predict one of the properties of future unseen data [2].

Creating a prediction model from previously known dataset is called training. The data used for training a model is called training data or training set. Once the model is created it is tested against a dataset which is not part of training set to test the efficiency and effectiveness of the model. The data used for testing the model is called test data or test set. The model is then fine tuned and reiterated multiple times to find the best accuracy score for the test set. Once the model is efficient enough for test set, it can be said that it is generalized for any un seen data and can be used on production.

The reason for splitting the data in two different sets (training and testing) is to avoid overfitting. If we use all the data for training then the model will be highly efficient for training set but will perform poorly on unseen data [17].

We have used predicitive analysis to learn the relationship of different properties of a betting addict, and then used the model to predict for all users if they are potential betting addicts or not. The implementation and system overview part tells more about the methodolgy and implementation.

2.6 Classification and Selected Methods

There are hundreds of different classification algorithms which learn from a dataset and predict whether a record belongs to a class or not. Each algorithm has the same task i.e predicting a dependant variable. They are based on different mathematical methods with their own weaknesses and strengths. We have used two different algorithms, Support vector machines with linear kernel and Naive bayes classifier for the usecase and found SVM to be performing better.

2.6.1 Support Vector Machines

Support Vector Machines (SVMs) as originally proposed by Vladimir Vapnik [19] within the area of statistical learning theory and structural risk minimization, have demonstrated to work successfully on various classification and forecasting problems. SVMs have been used in many pattern recognition and regression estimation problems and have been applied to the problems of dependency estimation, forecasting and constructing intelligent machines [8]. In Multi Layer Perceptron (MLP) classifiers, the weights are updated during the training phase for which the total sum of errors among the network out-

puts and the desired output is minimized. The performance of the network strongly degrades for small data sizes, as the decision boundaries between classes acquired by training are indirect to resolute and the generalization ability is dependent on the training approach. In contrast to this, in SVM the decision boundaries are directly determined from the training data set for which the separating margins of the boundaries can be maximized in feature space. A SVM is a maximum fringe hyperplane that lies in some space and classifies the data separated by non-linear boundaries which can be constructed by locating a set of hyperplanes that separate two or more classes of data points. After construction of the hyperplanes, the SVM discovers the boundaries between the input classes and the input elements defining the boundaries (support vectors). From a set of given training samples labeled either positive or negative, a maximum margin hyperplane splits the positive or negative training sample, as a result the distance between the margin and the hyperplane is maximized. If there exist no hyperplanes that can split the positive or negative samples, a SVM selects a hyperplane that splits the sample as austerely as possible, while still maximizing the distance to the nearest austerely split examples.

Mathematically it can be defined as the following:
We are given a training set of n data points.

$$(\vec{x_1}, y_1), ...(\vec{x_n}, y_n)$$

Where y_i are either 1 or -1 indicating the class in which the data element $\vec{x_i}$ belongs. SVM tries to find the 'maximum margin hyperplane' that divides the group of points $\vec{x_i}$ for which $y_i = 1$ from the group of $\vec{x_i}$ for which $y_i = -1$ so that the distance between the hyperplane and the nearest point from either group is maximized. A hyperplane can be written as the set of points $\vec{x}$ satisfying:

$$\vec{w}.\vec{b} - b = 0$$

Where $\vec{w}$ is the normal vector to the hyperplane.

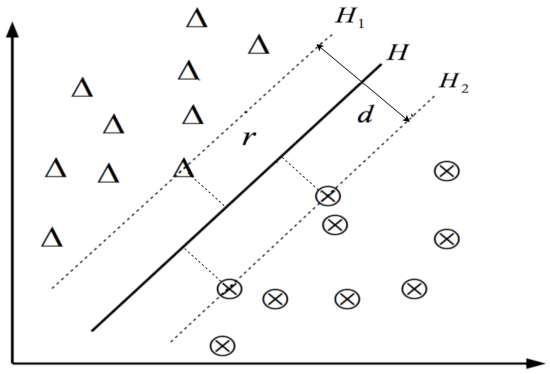

Figure 2.1: Linearly Separable Samples Indicated in a Hyperplane

Figure 2.1 indicates a linearly separable hyper plane, where there are two groups of data points represented by '⊗' and '△'. There may be possibility of an infinite no. of hyper planes but in the described figure, only one hyper plane represented by solid line optimally separates the sample points and is situated in between the maximal margins.

2.6.2 Naive Bayes Classifier

Naive Bayes classification is a machine learning algorithm which relies on Bayes' Theorem:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

where A and B are two different events, *P(A)* and *P(B)* are the probability of A and B occurring, respectively. *P(A—B)* is the probability of A occurring given that B has already occurred [12].

The Naive Bayes Classifier makes strong assumptions about how the data is generated, and posit a probabilistic model that embodies these assumptions; then they use a collection of labeled training examples to estimate the parameters of the generative model. Classification on new examples is performed with Bayes' rule by selecting the class that is most likely to have generated the example. It assumes that all attributes of the examples are independent of each other given the context of the class. This is the so-called "naive Bayes assumption."

2.7 Feature Engineering

In machine learning, feature engineering is the process of selecting or creating features (variables) in a data set to improve machine learning results [7]. Feature engineering is an art which results in features that better represent the underlying problem. Feature engineering when done correctly results in improved model accuracy on unseen data.

Below are some of the most common methods of feature engineering.

2.7.1 Feature Selection

Feature selection is the process of selecting important features and removing redundant or useless features which have no correlation to the dependant variable. The process of removing unnecessary variables require assessing the relevance of the variable. This can be done by creating a model to test the correlation of the variable with the dependent variable.

2.7.2 Pearson Correlation Coefficient

The Pearson correlation coefficient measures the strength of linear association between two variables. It is defined as the ratio of the covariance of two variables representing a set of numerical data, normalised to the square root of their variances i.e .

$$\rho = \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \tag{2.1}$$

2.7.3 Feature Creation

Feature creation includes modifying the variables and creating new ones by combining multiple different variables [16]. Feature creation is an art which requires the domain knowledge of the underlying problem. With tabular data, it often means a mixture of aggregating or combining features to create new features, and decomposing or splitting features to create new features.

2.7.4 Feature Extraction

Some data sets are have too many features which results in a very complex model if all the features are used as it is. Some examples include audio, images and textual data but sometime even tabular data. Feature extraction is the process of automatically reducing the number of features (dimensions) of a data set which can be moddled.

Some methods for feature engineering are Principle component analysis (PCA), line or edge detection, unsupervised clustering methods etc [1].

2.7.5 Feature Scaling

Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the Euclidean distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance [11]. The features for this thesis were also in different ranges. For example no. of time a user logged in was a small numerical range while the money spent per month had a large range. For this reason we had to normalize our features.

2.7.6 Bias Variance Tradeoff and generalization

The target of every machine learning algorithm is to generalize the problem for any unseen data to make predictions. A machine learning model which performs highly in the training data but performs poorly on test data is called an overfitted model. One way to understand the problem of overfitting is by decomposing algorithm's generalization error (errors in prediction) into bias and variance. Bias is a machine learning algorithm's tendancy to consistently learn the wrong thing. Variance is the model's tendancy of learn random things irrespective of the real value [7]. In bias-variance terms the objective of an algorithm is to find a generalized value of bias and variance so the model performs well on unseen test data.

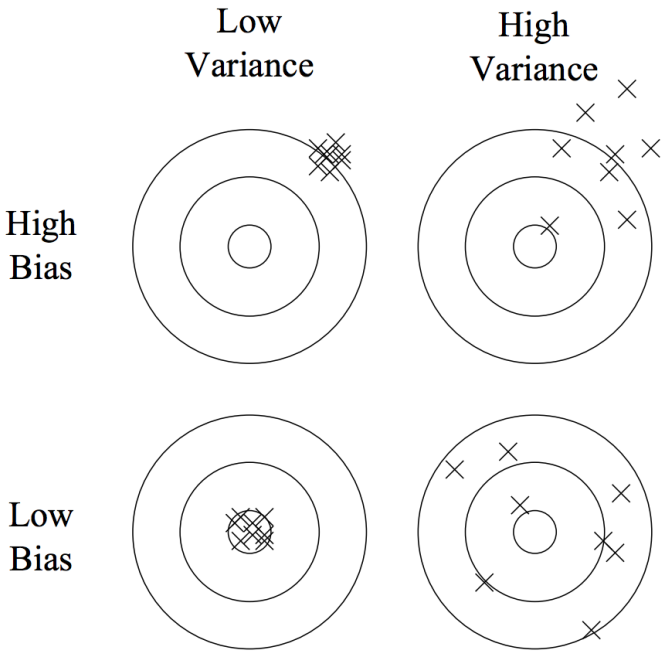

Figure 2.2: Bias and variance in dart-throwing.

Figure 2.2 illustrates an analogy of bias variance dillema with darts. Imagine that the center of the target is a model that perfectly predicts the correct values. As we move away from the bulls-eye, our predictions get worse and worse. Imagine we can repeat our entire model building process to get a number of separate hits on the target. Each hit represents an individual realization of our model, given the chance variability in the training data we gather. Sometimes we will get a good distribution of training data so we predict very well and we are close to the bulls-eye, while sometimes our training data might be full of outliers or non-standard values resulting in poorer predictions. These different realizations result in a scatter of hits on the target.

We can plot four different cases representing combinations of both high and low bias and variance.

2.7.7 Imbalanced classes in predictive analysis

A dataset with majority of data belonging to one class, modeling a predictive classifier will always be sensitive towards majority class. If this issue is not

taken care of, the classifier will be biased and will predict the majority class in most cases. There are various methods to tackle this problem, We have used oversampling for this thesis.

Chapter 3

System Overview and Methodology

This chapter is a brief overview of the overall system and how different modules are communicating, the more detailed implementations are in Chapter 4 and 5.

3.1 Data and Domain Information

3.1.1 Domain Information

The gambling site X has a definition about who is a betting addict by definition based on a user's usage history on the site. If a user satisfies any of the following criteria, they are considered a gambling addict.

- Users who spend more than 10k GBP per month on the gambling site.
- Users who spend more than 8 hours on avarage on the gambling site.
- Users who have self excluded themselves on the website.
- Users who deposit money more often are more likely to be gambling addicts.
- Users who play more than 1000 bets per month are gambling addicts.
- Users who log in more than 300 times per month are gambling addicts.

Table 3.1: Data Tracking: What was tracked from the gambling site X.

Name	Explanation
Basic User Details	Basic user details such as the currency, country, age and login ID is tracked for each user.
Time Spent Per Page	how much time user is spending on each page
Usage Summary History	Summary (aggregation) of how much money user is spending per each section of the site
Deposit/Withdrawal History	Each deposits and withdrawals the user has made during the specified time.
Gambling History	the detailed gambling history for each user for a specified time.
Responsible Gambling	They have a separate responsible gaming page on the site where the user can restrict their usage on the site in different ways (eg: reality check, self exclusion, time out etc)
Site Usage	This is a different way to track time on the site since time spent per page can be unreliable in cases when the user does not stay on the same page and closes the tab/page after staying inactive on it. Site Usage keeps on sending a flag to the server that the user has spent 15 minutes active on the site.

3.1.2 Data Information

We are collecting the following data from the website which are later used as features for the machine learning Application.

- The website has a separate sports section. It was not possible for me to track any sports section specific betting data. The data collected is from all the rest of the sections of the website.
- User's device (mobile or desktop site), IP, number of times logged in/out, timestamp etc is also tracked for each user.

3.2 Data Collection and Preprocessing

The data was collected from the website through Frosmo's tools. Frosmo use data tracking from ecommerce websites to show product recommendations

and better user experiences. We modified their data tracking module to track custom usage data from the gambling website X. Data is sent for each user to a linux server using nginx as reverse proxy and stored in log files. Nginx is responsible for storing the data in log files, sorting files using the date in the name and compressing older files. 3.1 shows how the data tracking module is working. Data is sent to the server with an API get request within the query parameters.

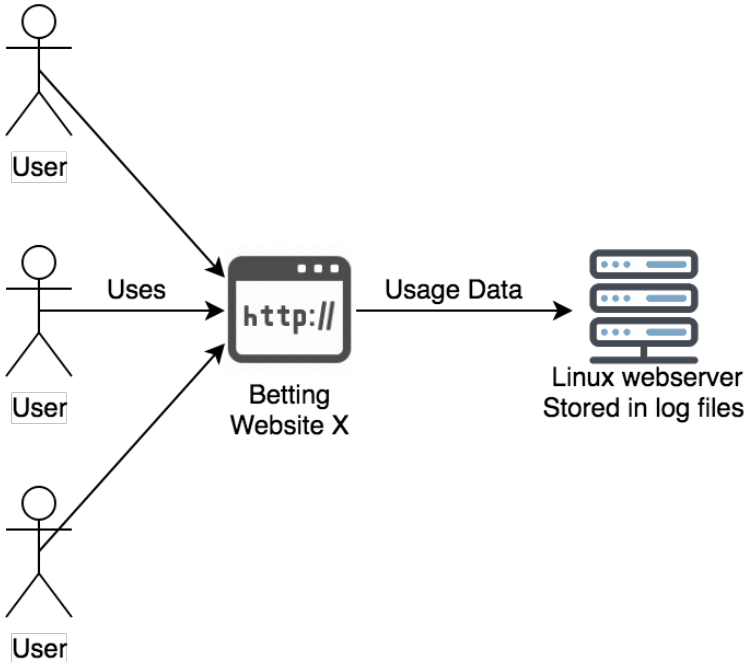

Figure 3.1: Tracking custom data architecture.

We wrote a python log analyzer which is run everyday at 5:00 AM through cron. It reads the previous day's log, cleans the data and stores it in the database. The relational database has a schema tailored for collecting data for gambling website X with composite primary keys to avoid duplication of records. There is another python based script that runs every month and stores the aggregated monthly usage data in the respective database tables. 3.2 shows the basic arcitechture of how data preprocessing is done.

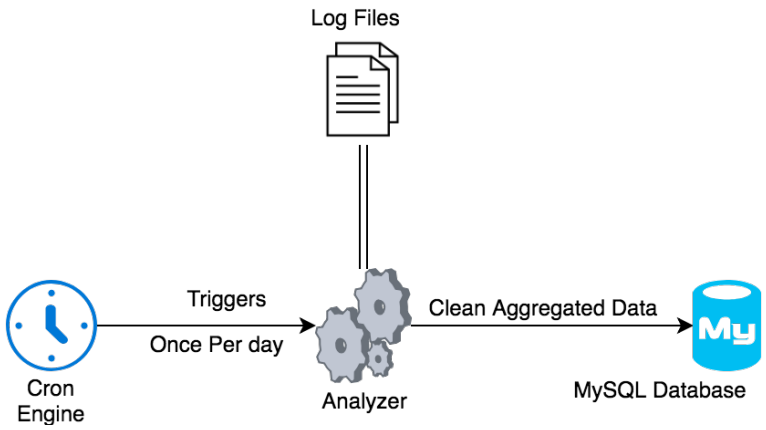

Figure 3.2: Data preprocessing and storing.

3.3 Machine Learning Modeling

The aggregated monthly usage data of each user was used to build a predictor for classifying users into potential addicts or non addicts. We used the users who are by definition gambling addicts and non addict users to create a machine learning model of betting addiction. We first scaled the features, oversampled the minority class, used grid search to find the best hyper parameters for the algorithm and then trained the model for potential betting addicts.

Once the model is trained, it can predict users based on their behavior history for potential betting addiction. We then used all the non betting addicts to predict if they are potential betting addicts based on their last 30 days usage history. This whole process is automated and repeated every day through cron scripts.

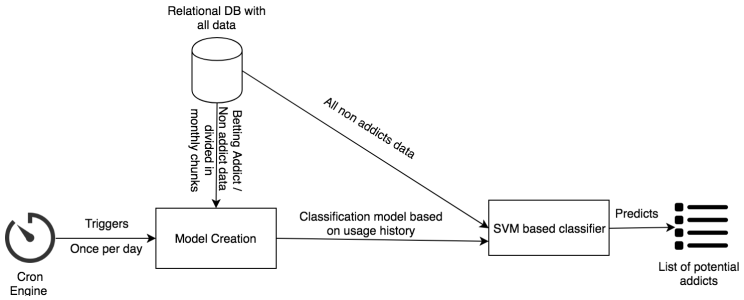

Figure 3.3: Classification of potential betting addicts

Chapter 4

Data Collection and Preprocessing

4.1 Motivation and Core Problems

For any data science application there needs to be data to use the algorithms on. The supervised learning algorithms used in this thesis generalize the user site usage data to form a machine learning model for prediction. For this reason we need to track user data somehow from the website. There are many commercial data tracking tools available like HotJar, Google Analytics among others. The problem with using commercial third party tools with a website with this much daily traffic is some solutions were costing a lot or some did not have exended tracking option available to track customized data. Due to these reasons we ended up setting up a custom tracking module using Frosmo solutions.

As described briefly in the previous chapter the data tracking module is built into the Frosmo script. The gambling site X was asked to add the script in the head section of all the web pages. The data tracking module then was able to subscribe to user actions on the page and send the data to a dedicated server.

4.2 User Data Tracking

4.2.1 Tracking Mechanism

The data tracking mechanism needed to be failsafe and scalable because the gambling site X was high traffic. The data is sent from user's browser to a dedicated server through API calls. The server where the data is sent had

nginx installed as reverse proxy which stored every API call in log files. From the user's browser instead of making AJAX call we are maintaining a queue in the user's local storage. Each record to be sent is pushed to the queue. There is a queue listener which removes one record at a time from the queue and send it to the server. By the nature of how it is implemented there are high chances of having duplicates on the server. These duplicates are later removed in the log analysis stage.

We are tracking user interaction data from different sections of the website. Following is the description of how exactly each data feature is tracked and how we intend to use it.

4.2.2 Time Spent

We are tracking the user's time spent on the site by two different methods.

1. Timespent per page.
2. User Activity time.

Time spent per page was done so that every time the user opens a page we store the timestamp in a javascript variable. When the user is leaving the page the stored timestamp is subtracted from the current timestamp to get the duration of time spent on that page. This works well if the user continues interaction on the page but for inactive pages the time tracked will be wrong if the user just keeps the page open but does not actually do anything on it. For this reason we tracked the time spent by the second method as well which caters this problem.

The second method of user activity time works differently than the first method. The difference is to cater the problems the first method has. Instead of recording exact time spent on the page it keeps on sending a flag to the server every 15 minutes if the page is active and there is some activity on the page. This works well for the game pages where the user is spending a lot of time but not on the pages where the user just bounces off after a few seconds.

A good estimate of time spent of users on the site could be average of both the methods since both methods has their problems but both combined solve each other's problems. A more detailed analysis of both the methods is presented later in this chapter.

4.2.3 Basic User Details

We need basic user details like currency, country age and loginId for various reasons.

- The methods we get the user's transactions history do not return the currency of the transactions as it can be in GBP, EUR or USD. In order to normalize the amounts in one currency unit we need to track the currency unit for each user.
- We are tracking age of the user (Date of birth) because the age could be one of the factors for user's addiction.
- We are tracking the loginId because we need to identify the users later on while sending the list of potential betting addicts back to the gambling site X.
- We are tracking the country because it could be interesting to see the geographic map of betting addicts.

Some of the basic user details data was available in a global JavaScript object. The others we had to track from the user's profile page.

4.2.4 Deposit/Withdrawal History & Gaming History

The gambling site has an account history section where all the transactions made on the site (Deposit, widthdrawal and win/loss amount of each game played) is displayed. Since the whole website was made with modern frontend tools (Angularjs), the data displayed there was fetched through ajax calls. We made the same AJAX calls to the website's server and got the data from there. In order to make sure we did not end up abusing those APIs, it was made so that those API calls were made only once per day.

Each transaction record and gaming history data record had a unique identifier as the timestamp of when the transaction was made. Even if the data is sent twice in the case of a user using multiple devices duplicates are removed in the data analysis step.

4.2.5 Responsible Gambling Section

The gambling site has a separate Responsible gambling page where the users can limit the interactions on the site. The four ways the user can limit their usage on the site are:

- Self Exclusion: Removes the user permanently from the site. A hard indicator that the user was worried about his usage on the site.
- Time out: Removes the user from the site for a specified time period. Hard indicator if it is for more than a couple of months.

- Reality Check: Warns the user about their site usage after certain number of hours of continious usage. The user can select the number of hours. It is a soft indicator because users can choose this as a precautionary measure.
- Deposit Limit: Setting a deposit limit will not let users deposit more money than they have selected. It is also a soft indicator because it can be for precautionary reasons.

This data is tracked directly from the responsible gaming page. If the user changes something on that page it is sent and updated for that user.

4.2.6 Other User Data

Other than the above mentioned data we are also tracking user's IP where the data is sent from, the device (whether handheld or desktop/laptop) and user's unique cookie ID.

4.3 Log Analysis and Aggregating Data

After the data is tracked and stored in log files, we need to process the data, analyze it, aggregate it and store it in some sort of persistent database. We wrote a python log analyzer which runs every day through cron script. It read the previous day's data and store it in the MySql database. There is another cron script that runs on first day of the month to aggregate the previous month's data in specific SQL tables. This section has detailed information about the whole process.
All the site usage data that server receives through a specific API end point is stored in the log files. The reverse proxy Nginx is configured to divide the chunk of log files based on the date it was received.

4.3.1 Daily Log Analysis

There is cron script that runs everyday at 5:00 AM. The script gets the previous day's log and analyzes it. In the analyzing process it goes through each API call record, get the user and other details and stores it in the database. There is a unique constraint on the API call identifier which ensures that no duplicates are stored in the database. After the analysis is done for that day, it stores in the database the current day's date allowing the analyzer to start from the updated log the next day.

4.3.2 Monthly Log Analysis

There is a monthly cron script that triggers the monthly analyzer on 1st day of each month. The purpose of having monthly analysis is to aggregate the data and store the monthly summary of each user in specific database tables for monthly data. Although, the data in the database is indexed with multiple keys. It is more efficient to use the aggregated value stored in a separate table than to find the aggregates separately in each query.
Both daily and monthly Analyzers stores the data in a MySQL database with a normalized schema. The most important tables with their details are given below.

4.3.3 Tables for Daily Analysis

- ml_user: User basic details (loginID, gender, language, dob, currency, country)
- ml_user_action: There are several actions users can perform on the site (eg: setting deposit limit, logging in, site usage, reality check etc). This table has action records for each user. Some actions have payload and period too, for example the amount of deposit limit or time out duration.
- ml_wd_deposit: Each record is the history of deposit and withdrawal for each user as a transaction.
- ml_user_page: The time spent on each page, pages are defined as different section of the site.
- ml_user_ip: UserID and the different ips users have accessed the site from.

4.3.4 Tables for monthly Analysis and aggregation.

As mentioned in previous section, there is a cron script that runs every month, aggregates the data for previous month and store it in the separate monthly tables, some of them are:

- ml_monthly_action: Count and aggregation of each user's actions for the previous month.
- ml_monthly_bet: Aggregated summary of each user's previous months betting. Columns include how much money was on stake, sum of win/loss, count of bets etc.

- ml_monthly_timespent: Time user spent on the site in the previous month.
- ml_monthly_wd_dep: Aggregation of widthrawal and deposit of previous month with deposit and withdrawal frequencies.

4.3.5 Database views

Since the data is stored in the database with their specific tables, we needed to create several database views to connect different tables and get specific data from there for data analysis and machine learning algorithms. Following is the list of some most important database views and their purposes.

- ml_user_monthly: It connects different monthly aggregated tables and creates a generalized view for each user for each month.
- ml_addicted_monthly: Augments the ml_user_monthly view with isAddicted column based on the addicted definition. This serves as the training data for machine learning algorithm.
- ml_addicted: This is also the last month's summary for each user. It is different from the above view because it shows the history for last 30 days rather than from monthly aggregated data. After the model is trained it classifies the user every day based on their 30 days usage from this view.

4.4 Data Analysis and Visualization

This section is dedicated to analyzing and visualizing the data to find interesting patterns out of it. Analyzing and visualizing the data will help in feature engineering part of the actual machine learning algorithm implementation.

4.4.1 Trends Validation

Attribute	Not Addicted	Addicted
Average Site Usage per day	69.75 minutes	143.7 minutes
Average # of times logged in per day	15.82	25.39
Average # of times deposit limit set in a month	0.06	2.08
Average # of times withdawal of money in a month	0.71	1.70
Average withdrawal amount from the site per month	191.0 GBP	633.6 GBP
Average User Age	42.3 years	36.8 years

Some of the trends in above section are pretty self explanatory. For example on average addicts use twice as much time on the site as compared to non addicts, they have a higher withdrawal frequency and much higher deposit frequency. They are also more likely to be concerned about their gambling problems and have used the responsible gambling section of the site (timeout, self exclusion and deposit limit). They are likely to spend much more money on the site.

4.4.2 Hidden Trends

Some of the hidden patterns realized from the visualizations are bet count, win/loss ratio, the method of time tracking and age distribution of addicted between addicted and non addicted users. The bet count is the number of times the user gambles on the site. By the definition of addicted users anyone who plays more than 1000 times per month is considered a betting addict. However, the bet count among addicts are reaching much higher with an average of 1121 and 90th percentile reaching over 2,000+ games count per month. This is an indicator that the threshold value of betting addiction bet count should be much higher. The first intuition about win/loss ratio was betting addicts who can not stay away from gambling must be loosing a lot of money, the data proves it wrong. The addicts on average wins 50GBP more than non addicts with 90th percentile winning three times as much as non addicts. This means that they are not just breaking their banks gambling on the site but do it smartly to win more money.

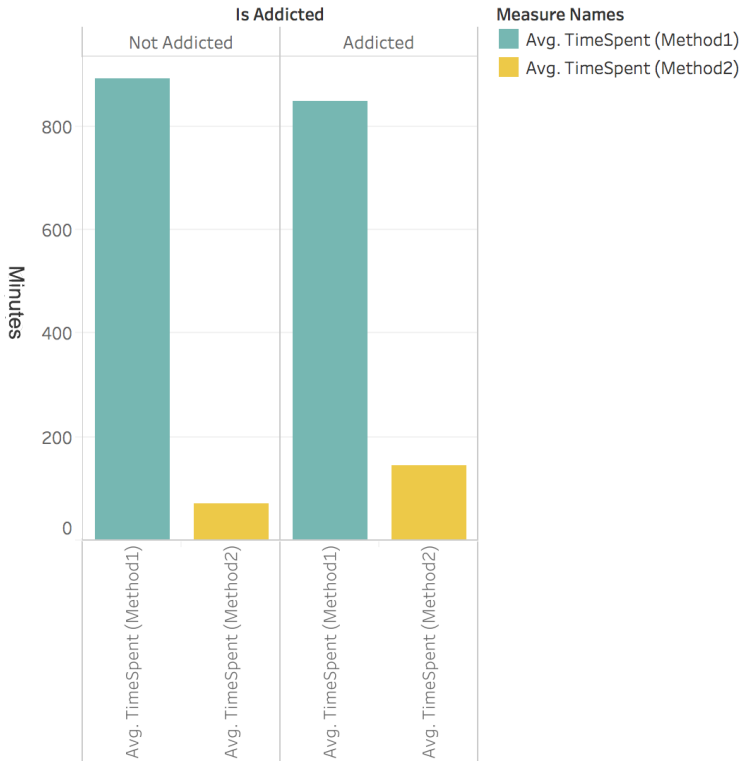

Figure 4.1: Difference of two methods for tracking time.

As discussed in the data tracking section, we have two methods of tracking user time spent on a page. One is to send time spent on each page after the user leaves a page and the other is to send a flag of timespent every 15 minutes. 4.1 shows that the two methods are not reporting relevant similar results. The first mehtod has almost the same average value for both addicts and non addicts and has a much higher average as compared to the second method. The explanation for this could be that since the first method sends time spent after each page load, the pages that are open for a long time without any activity report the time as time spent. The second method is more smartly tracked, it only reports timespent after 15 minutes of active site usage for a tab. For this reason, we will only consider the second method in our model generation.

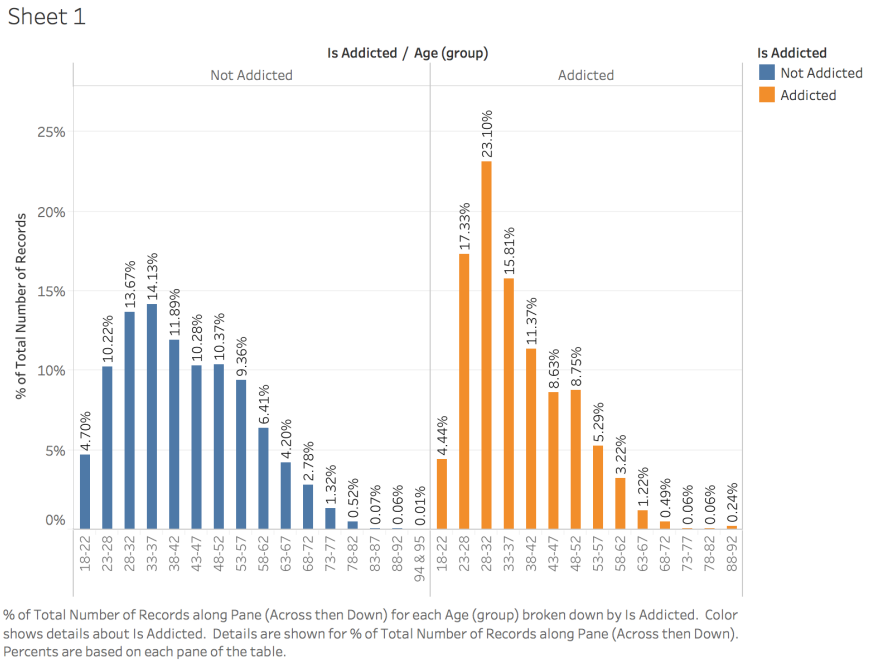

Figure 4.2: Age distribution of addicted between addicted and non addicted users

After visualizing the data we found out that users with the age between 23-32 are more likely to be addicts than other age groups.

Chapter 5

Machine Learning Modeling

5.1 Motivation and Core Problem

The machine learning problem to solve in this context is to find users whose behaviour resembles betting addicts. As already written in previous chapters that finding betting addicts is a deterministic problem. Online gambling company X has a strict definition about who to consider betting addict. The machine learning model uses both addicts and non addicts data to create a generalized model about who might be a betting addict despite them not reaching the threshold for being a gambling addict.

Setting an evaluation criteria for such a problem is difficult since after the algorithm classifies potential gambling addicts, we have no way to check if the users actually turn out to be addicts in future. We can check that if predicted users turn out to be addicts but the usage can drop down or go up based on certain real life conditions.

5.2 Dataset and Features List

The tracked, aggregated and filtered data has 29600 records of monthly user activity and an independent variable is_addicted defining if the user was addicted in that month. 5.1 shows the list of features used for the classication task, data types and the number of records.

```
In [13]: import pandas as pd
         data = pd.read_csv('addicted.csv')
         data.drop(['uid','time_spent','IP','month'], axis=1, inplace=True)
         data.info()

         <class 'pandas.core.frame.DataFrame'>
         RangeIndex: 29600 entries, 0 to 29599
         Data columns (total 14 columns):
         age              29600 non-null int64
         betCount         29600 non-null int64
         stake            29600 non-null float64
         win_loss         29600 non-null float64
         LoggedIn         29600 non-null int64
         DepositLimit     29600 non-null int64
         SiteUsage        29600 non-null int64
         TimeOut          29600 non-null int64
         SelfExclusion    29600 non-null int64
         m_dep_freq       29600 non-null float64
         m_wd             29600 non-null float64
         m_wd_freq        29600 non-null float64
         gender           29600 non-null int64
         is_addicted      29600 non-null int64
         dtypes: float64(5), int64(9)
```

Figure 5.1: Features list, data types and number of records.

The distribution of records between addictive and non addictive is not uniform. Only 18% of the users are addicted to the gambling while the rest are not addicted.

5.3 Feature Engineering

We have used various feature engineering techniques as a machine learning preprocessing step, the details of everything we did are as follows.

5.3.1 Data Normalization

The dataset features are rescaled so that they have the properties of a standard normal distribution with 0 mean (μ) and 1 standard diviation (σ). Data normalization resulted in features which had a different unit but were all in a similar scale.

5.3.2 Pearson's Correlation

Pearson's coorrelation method is used to find the correlation between the features. The yellow a box in the 5.2 shows that two features are highly correlated, while the black section shows no correlation. It can be concluded that from 5.2 that features used to solve the machine learning problem are not correlated and hence can all be used together.

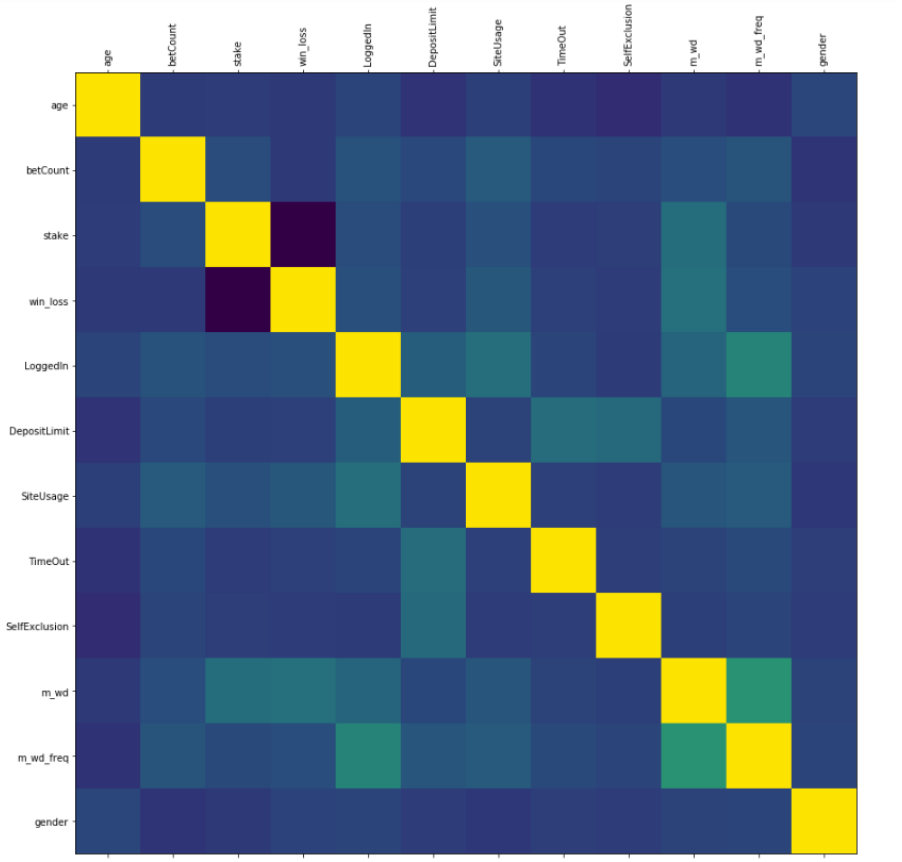

Figure 5.2: Pearson's Correlation between features.

5.3.3 Oversampling

The classification training data had an imbalance class distribution. More than 80% of the users were not addicted and the remaining were addicted. Creating a predictive model from such a data yeilds a highly biased classifier which predicts the majority class most number of times. For this reason the minority data was oversampled to match the number of majority class elements. Python's imbalanced-learn class was used to oversample the minority class using SMOTE method. It resulted in a dataset with 27955 records for each class.

5.4 Machine Learning Modeling

After the feature engineering the data was split into 75% train data and 25% test data. Two different algorithms support vector machine(SVM) and Naive Bayes Classifier were used for classification. The models were further evaluated using KFold method using 10 folds. The mean accuracy and confusion matrix were used as an evaluation criteria for model performance.

5.4.1 Hyperparameters Optimization

Hyperpermeters are different values an machine learning alogrithm can be initialized with. For Support Vector Machines GridSearchCV was used to find the different hyperparameter values like the type of kernel and value of C the penalty parameter. It was found that with over 50,000 data records, kernels other than linear were too expensive to optimize, for this reason the linear kernel was used.

5.4.2 Model Evaluation

It was found that support vector machines over performed naive bayes classifier on average. The average accuracy SVM with linear kernel acheived was 98.34% while Naive Bayes Classifier acheived 96.34% mean accuracy with 10 validations. 5.4 and 5.3 shows the model evaluation through confusion matrix. Based on these results a classifier based on support vector machines was used in the prediction of future test data.

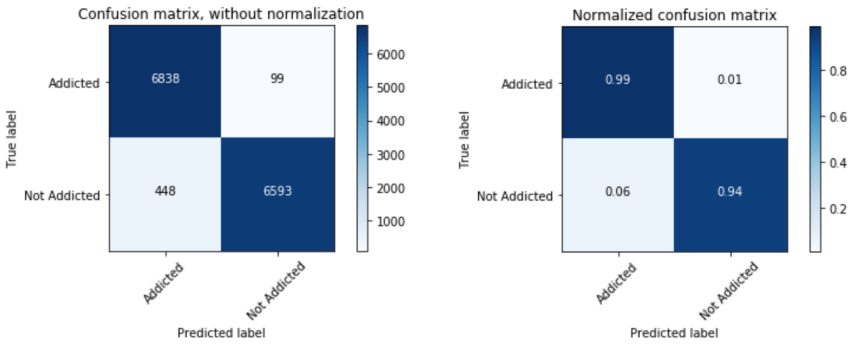

Figure 5.3: Normalized and unnormalized confusion matrix visualization for model with Naive Bayes Classifier

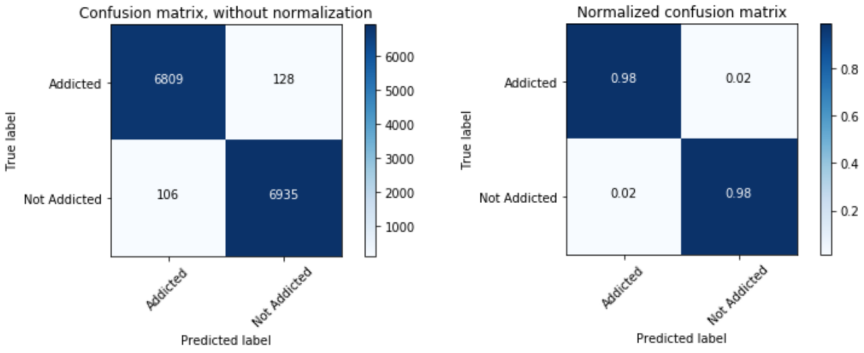

Figure 5.4: Normalized and unnormalized confusion matrix visualization for model with Support Vector Machines

5.5 Automation

The whole machine learning modeling and prediction method was needed to be performed on daily basis. The monthly record for each user was used for model creation but the prediction was done on daily basis based on last 30 day's history of each user.

A daily cron script triggered the learning and prediction process. After the predictions were made, the results were stored in a log file and a list of potentially addicted users were sent to the gambling site X.

Chapter 6

Results

The work done as part of the thesis resulted in a complete end to end system from tracking user data to predicting potential betting addicts. 6.1 shows the architecture of the complete system.

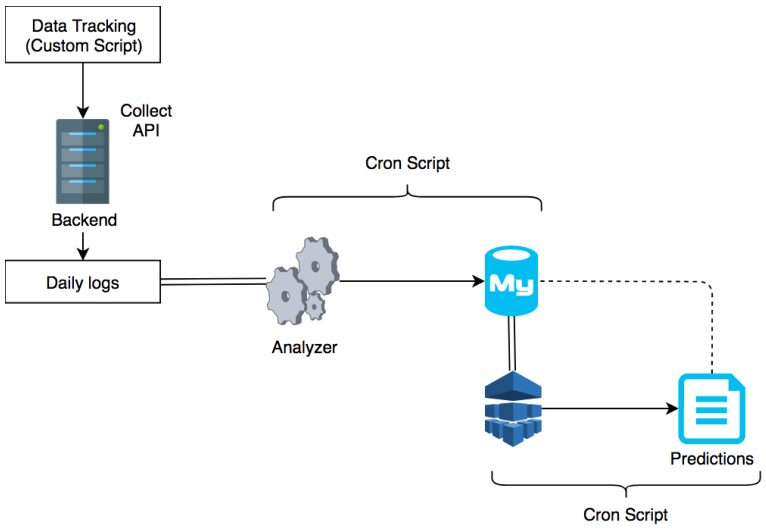

Figure 6.1: End to end architecture of the complete system

This chapter presents the key findings from the system through data analysis and machine machine learning predictions. Furthermore, We have given recommendations for how the system can be improved further for identifyign gambling addicts with more precision.

6.1 Data Analysis

Data visualization and analysis resulted in validation of some already known trends and identifying hidden trends among addicts and non addicts. These findings can be found in more details in Chapter 4.

6.1.1 Trends Validation

- Addicted users spend twice as much time on the site as compared to non addicted users.
- Addicted users on average log in 1.6 times more number of times than non addicted users.
- Most addicted users are aware of their gambling problems and they try to control their gambling habits through various responsible gambling means.
- Addicted users deposit money more frequently and larger sums of money on average as compared to non addicts.

6.1.2 Hidden Trends

- Addicted users on average wins 50GBP more than non addicts with 90th percentile winning three times as much as non addicts.
- Users with age between 23-32 are more likely to be addicts than any other age groups.
- The mean age for gambling addicts is 37 years as compared to 42 for non addicts.

6.2 Machine Learning Predictions

- The system predicted that 14.5% of all the users were potential betting addicts which weren't identified from definition of gambling addiction.
- The system predicted that most users who are predicted as gambling addicts are predicted for 1 month only, this indicates that not all potential gambling addicts end up being gambling addicts.

- The system predicted that only 4.6% of potential gambling addicts continued to be potential gambling addicts for more than 2 months in continuation.

From these results it can be deduced that the system can be used as a soft indicator for finding potential gambling addicts. However, the trend shows that out of those potential gambling addicts only 4.6% of the users continue to show addiction behaviour for more than 2 months in continuation.

Chapter 7

Discussion

7.1 Limitations of the System

- The data tracked from the gambling site did not include the sports section of the website because of technical limitations. It resulted in the limitation of finding potential betting addicts who spend a lot of time and money on sports section of the website.
- The time tracking mechanism for each user sends a flag every 15 minutes of usage from the site, the time before the next time flag is sent is not tracked resulting in an inaccurate picture of time spent tracking.

7.2 Future Recommendations

Based on the work done for the thesis, we think that the results can be improved further on the basis of choosing better tools and platforms and improving the process of feature engineering. Our recommendations for improving the performance and impact of the feature are as follows:

7.2.1 Technological Choices

- The data tracked from the gambling site X did not include usage on the sports section of the site because of technical limitations. To get a full picture of addiction behavior the data from sports section needs to be included.
- Non-relational persistent storage such as MongoDB or big data toosl such as Hadoop or Spark should be used in this or similar applications. The unstructured nature of data makes it difficult to fetch customized

views from a relational database. It is also difficult to modify the database schema once the data is stored and indexed. Using relational database was a major limitation in fethings customized data from the database.

- The current system is built and hosted on a linux virtual private server. If built on cloud like Amazon Web Services or Microsoft Azure it can use a lot more powerful off the shelf components for building the data pipeline.

7.2.2 Feature Engineering

- The current time tracking technique spans the time spent on the whole site, it can be further broken into time spent gambling and time spent on the rest of the site.
- The current bet count can be broken down into further categories of gambling to find which games are more addictive.
- The user's activity on the site can be further analyzed based on demographics and devices used. If a user uses multiple devices and gambles from multiple locations (home, office or on vacations), it can be analyzed for correlation with gambling addiction.
- The time when the user gambles on the site also can be analyzed. For example if a user spends average amount of time and money gambling throughout the week but gambles heavily on friday evenings or on the weekends, it shows an anamoly and a problem gambling behavior. This can be further analyzed for correlation with gambling addiction.

7.3 Conclusion

The work done for this thesis tries to find potential gambling addicts on a gambling site X. The gambling company X defines a user a gambling addict if he/she fulfils certain criterions based on their usage history. We used this labeled data for training the models to find potential gambling addicts based on their recent usage statistics. We tracked site usage data for all users from the site, the tracked data was cleaned, aggregated and analyzed to find patterns in the usage behaviors of users. The cleaned and aggregated data was then used for training the machine learning model, we tried two different algorithms and found Support Vector Machines to be more performant. The

whole system was then automated to perform daily learning and predictions automatically through cronscripts. The post production evaluation of the model found that only 4.6% of the users turn out to be gambling addicts for more than two months after being identified as potential betting addicts by the system. We have given several recommendations for improving the system.

Bibliography

[1] Discover feature engineering, how to engineer features and how to get good at it. https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/. Accessed: 2018-04-23.

[2] Extending the value of your data warehousing investment. https://tdwi.org/articles/2007/05/10/predictive-analytics.aspx. Accessed: 2018-04-19.

[3] Frosmo - running with e-commerce. https://frosmo.com/solution/. Accessed: 2018-04-19.

[4] Predictive analytics and machine learning solutions. https://www.sas.com/content/dam/SAS/en_us/doc/analystreport/forrester-predictive-analytics-machine-learning-108754.pdf. Accessed: 2018-04-19.

[5] Association, A. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5®)*. American Psychiatric Publishing, 2013.

[6] Atterer, R., Wnuk, M., and Schmidt, A. Knowing the user's every move: User activity tracking for website usability evaluation and implicit interaction. In *Proceedings of the 15th International Conference on World Wide Web* (New York, NY, USA, 2006), WWW '06, ACM, pp. 203–212.

[7] Domingos, P. A few useful things to know about machine learning. *Commun. ACM 55*, 10 (Oct. 2012), 78–87.

[8] Ekici, S. Support vector machines for classification and locating faults on transmission lines. *Applied Soft Computing 12*, 6 (jun 2012), 1650–1658.

[9] GOODMAN, A. Addiction: definition and implications. *Addiction 85*, 11 (nov 1990), 1403–1408.

[10] Heather, N. A conceptual framework for explaining drug addiction. *Journal of Psychopharmacology 12*, 1 (1998), 3–7. PMID: 9584962.

[11] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR abs/1502.03167* (2015).

[12] Islam, M., Akhter, F., Laz, R., and Parween, S. Islam et al. 2007 j-bio-sci, 02 2016.

[13] Kuss, D., and Griffiths, M. *Internet Gambling Addiction*, vol. 1. 01 2012.

[14] Mitchell, T. M. *Machine Learning*, 1 ed. McGraw-Hill, Inc., New York, NY, USA, 1997.

[15] Nyce, C. Predictive analytics white paper.

[16] Sap, M., Park, G., Eichstaedt, J., Kern, M., Stillwell, D., Kosinski, M., Ungar, L., and Schwartz, H. A. Developing age and gender predictive lexica over social media. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1146–1151.

[17] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. 1929–1958.

[18] Sussman, S., Lisha, N., and Griffiths, M. Prevalence of the addictions: A problem of the majority or the minority? *Evaluation & the Health Professions 34*, 1 (2011), 3–56. PMID: 20876085.

[19] Vapnik, V. An overview of statistical learning theory. *IEEE Transactions on Neural Networks 10*, 5 (1999), 988–999.

[20] West, R. Theories of addiction. *Addiction 96*, 1 (jan 2001), 3–13.