2016

Enhanced feature mining and classifier models to predict customer churn for an e-retailer

Karthik B. Subramanya
Iowa State University

Recommended Citation

Subramanya, Karthik B., "Enhanced feature mining and classifier models to predict customer churn for an e-retailer" (2016). *Graduate Theses and Dissertations*. 16023.
https://lib.dr.iastate.edu/etd/16023

Enhanced feature mining and classifier models to predict customer churn for an e-retailer

by

Karthik Subramanya

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical and Computer Engineering

Program of Study Committee:

Arun Somani, Major Professor

Srikanta Tirthapura

Glenn Luecke

Iowa State University

Ames, Iowa

2016

DEDICATION

I would like to dedicate this thesis to my parents and all my teachers who are largely responsible to have groomed us into individuals we are today. I would also like to thank my friends and family for their loving guidance and encouragement all through my life. Last but not the least, I would sincerely thank my adviser for providing an opportunity to work under him and constantly provide guidance and motivation to come up with this Work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Arun Somani for his guidance, patience and support throughout this research and the writing of this thesis. His technical prowess and words of encouragement have always inspired me all through my graduate study. He also motivated me to take up the thesis track in Graduate education. I would also like to thank my committee members for their kind guidance: Dr.Srikanta Thirthapura and Dr.Greg Luecke. I would additionally thank my employer for providing me with an opportunity to come up with this work through constant encouragement.

ABSTRACT

The evolution of large scale distributed computing, sustained progress in augmenting the technical expertise in algorithms and data sciences in the recent past have opened new avenues to address several large problems in science and commerce which were previously not feasible due to lack of computing infrastructure. Developments in algorithms and advanced statistical modeling i.e, machine learning, has provided the required intelligence methods to handle large volume of data, or BIG Data [MCB$^+$11] for applications such as basic sciences, further advanced topics like Climate patterns, Bio-informatics, GIS, Infrastructure planning, finance, E-commerce, Social networking, Policy planning, etc.

We address an important problem, Customer churn [HHR10] faced across all industries who depend on customer loyalty for growing their businesses. Customer churn is formally defined as a customer abandoning an established relation with a organization. It is also called as customer attrition, customer turnover or customer defection according to the wikipedia. Predicting customer churn is prioritized by businesses to save their businesses as the cost of retaining an existing customer is far less than acquiring a new one [FP08].

Customer churn models are applicable in many industries, like financial, telecom and automobile industries to name a few [XLNY09, AKR08, WC02]. We develop our own customer churn predictive model for E-commerce industry that leverages some of the advantages a Big Data infrastructure brings to the table. Our work is well tailored to suit the industry model. To the best of our knowledge there is no published work on customer churn prediction for an e-retailer that is similar to our model in terms of Data mining and model building.

We use a binomial classifier approach [Alp14] by first deriving a customer feature matrix using customer data. This model is often used by researchers in the field of medicine, drug discovery, disease diagnosis, sports, etc. We model our entire customer base as a feature matrix [DL97] with each customer representing a feature vector containing a combination of features

that influences his/her churn. We then apply a suitable feature selection algorithm [MBN02] to choose the best subset of features from the feature vector. Next, we apply classifier algorithms [KZP07] on the resultant data and cross-validate the results from the predictions.

CHAPTER 1. OVERVIEW

The paradigm of distributed parallel computing is increasingly enabling progress in big data technologies and enabling new avenues for large number of data-centric and compute intensive applications. Availability of supercomputers with thousands of nodes and high speed communication also allows solving complex problem involving large data and computation that was once outside the realm of possibilities.

Hadoop technology [SKRC10], provides commercially scalable big data technology that is completely open source (Apache creative common license) and is embraced by industry and academia all across the world. Several technologies that enabled solutions include high performance computation using Message passing Interface (MPI) for distributed computing [GL99], OpenMP for shared memory computing, distributed and redundant file systems, hadoop stack, NoSql [SKRC10] Databases etc. The advances in data sciences, machine learning, neural networks, deep learning, etc. have been developed to run on the hadoop stack. Some of the real world problems benefiting include but are not limited to Predictive analytics, Prescriptive analytics, Recommender systems, Financial forecasting, Sports analytics, DNA modelling, Climatic predictions and Cancer prediction.

1.1 The E-commerce Business Model

E-commerce [Gef00] (electronic commerce or EC) include the buying and selling of goods and services and the transmitting of funds or data, over an electronic network. These business transactions occur between business-to-business (B2B), business-to-consumer (B2C), consumer-to-consumer (C2C) or consumer-to-business (C2B). The terms e-commerce and e-retailer are often used interchangeably in this work. We are primarily interested in e-retail business which is a form of electronic commerce that allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app. It is projected that in the year 2017 the online e-retail industry will grow upwards of 600 billion dollars. Some of the household e-retail names are Amazon, Alibaba, Walmart, e-bay, Staples, Macy's, Apple, etc. While most of these e-retailers operate on a B2C business model, a B2B model or a combination of both is also common.

Many businesses have migrated from owning a Brick mortar shop alone to include e-retail business to cater to the needs of the customer and to keep up with the competition while others like Amazon, Alibaba follow only the e-commerce route.

Customer loyalty [SAP02] is an important driver to many E-retailers as the cost of acquiring a new customer is a significant effort in comparison to the cost of retaining one. Unlike a brick and mortar shopping experience that involves a look and feel, location advantage and human interaction component among others, the e-retail business model comes packaged in a single website from the landing page to exit. Therefore it is the most important priority of these companies to entice the customer with great line of products, pricing, attractive offers, recommendations, personalization, etc to create a desirable shopping experience.

1.2 Motivation and Previous Contributions

A review of Customer churn prediction approaches across various industry verticals and their efficiency motivates us to develop a new framework for churn in e-commerce customers. We surveyed their methodologies for data collections and algorithms, varied data sources that the researchers used for selecting features, their approach for feature selection algorithms, classification models, cross-validation techniques, etc were surveyed.

The features driving the algorithm can be used for other data science initiatives within the organization as it makes a rich set of features available for every customers that can be re-purposed to solve other problems in area of predictive & prescriptive customer analytics. This work also lays foundation for future work to drive other models like propensity to buy, customer segmentation, cross-sell, up-sell among these customers.

Our work follows closely around the model building techniques proposed in [XLNY09, YKG10, YGGH11]. Since we deal with a totally different industry vertical and a business problem to solve, the source and mixture of attributes that make up the data are different. The authors in [YKG10] predict customer churn of Google Adword customers. They first build a feature matrix for these customers and then further employ classification algorithms like Random Forests, Gradient Boost Method (GBM), etc. The authors in [XLNY09] predict customer churn in Bank's. They employ an Inverse Balance Random Forest classifier (IBRF), a technique that they use to prevent classifier algorithms from mis-classifying a minor class label on account of imbalance in the label distribution.

[YGGH11] proposes an enhanced Singular vector Machine (SVM) called the (ESVM) framework that claims to scale well over large scale data and ability to handle non-linear data effectively. [MCeC13] uses Multivariate regression Splines (MARS) as classification technique to detect customer churn. The authors in [CVdP09] and [CFS12] using several user behavioral data like email sentiment mining and longitudinal behavioral data to aid classifiers to make accurate predictions. The model proposed in [SR15] also uses significant qualitative customer behavior data to drive fraud detection in insurance claims using One class SVM (OCSVM) for classification task and K-reverse nearest neighborhood that handles class imbalances. Linear and non-linear classifiers like SVM, Logistic Regression, Artificial Neural Networks (ANN) and Tree based Ensemble classifiers and their variants are predominant choice of classifiers used by researchers to solve the customer churn problem. We observe significant gap in feature mining process in previously published work. They all fail to effectively represent the e-commerce business model. The choice of feature-set to drive churn prediction is mostly restricted to a list of conventional feature-set which has a huge share of static features and features generated through sales. Although some of the recent work [CFS12] published on customer churn for E-commerce customer focuses on behavioral features, this feature set is still narrow and restricted to only a handful of metrics like recency and frequency factors which by no means are able to capture the complete behavioral footprint for a customer during his life-cycle. The reason for this limitation in feature mining process can partly be attributed on technology limitations in data capturing and the rest on volume of the data that it entails to deal with.

The E-commerce business drives mainly on digital channels which are centered around, but not limited to online website and mobile applications. These channels act as a single window between the organization and the customer during his entire tenure. The shopping activity or the online interaction which can be labeled as a browse session generate valuable metrics and footprints for customer interaction with the online retailer. The sales generated by these online sessions, categories of products brought, etc that are more easily contained in volume, mainly qualify as explicit features for our feature matrix. The user click activity, browse path behavior and overall web interaction generates several terabytes of data every single day for an E-commerce retailer operating on a large scale. This valuable user behavior data that was previously ignored by researchers for feature mining owing to the lack of large scale data ingestion, storage and computing technology. This data can now be easily extracted through a feature mining process involving a big data pipeline. The proposed work aims to lay a firm foundation to develop a comprehensive feature mining process that starts from definition, extraction and the study of impact these features have on customer churn. We aim to capture a complete list of implicit and explicit customer footprints through feature matrix that enables better prediction of customer churn. We finally intend to come up with an end-to-end framework and make it available for the organization and the academic community for predicting customer churn for e-commerce business model.

1.3 Organization of Work

The thesis is divided into six chapters that follow a natural progression of our approach to the solution. Chapter one introduces the E-commerce business model and discusses the existing techniques and algorithms published for the customer churn. The Second chapter discusses about the customer engagement model for the e-retailer in consideration, we briefly discuss the analysis of the types of the segmentation applied for these customers, custom segmentation [Mah00] that are created on the fly. We then discuss how we mine individual features for the customer feature matrix for the sample size (segment) of customers. This is done using various customer channels within the organization that are housed across different systems.

The most important are the traditional order management system, Customer data warehouse, Clickstream logs [SLLM02], etc.

In chapter three, we discuss techniques such as feature selection [GE03] and dimensionality reduction algorithms to ensure that the feature matrix contains the best contenders to predict the output accurately and the ones that do not influence the output of the classifier are removed. This helps our model to increase speed, avoid issues like over fitting and to increase the accuracy of prediction.

In the fourth chapter, we discuss all the classification and regression models we consider for predicting customer churn for the chosen feature-set. We try a wide variety of linear and non-linear models, tree based ensemble models for deciding the best model.

In chapter five, we discuss the results of all the classification algorithms we explored in chapter four using cross validation techniques [AC$^+$10] employing ROC Curve and Confusion Matrices. We also discuss an evaluation plan for deciding the best Classifier model based on maximizing the area under the ROC curve[Bra97] where we empirically tweak the decision making probability thresholds of the algorithm to get the best possible area under the curve.

In chapter six, we present our conclusions and set the stage for future work in this area.

CHAPTER 2. OUR SOLUTION

Keeping e-commerce in mind, we develop a churn model to be able apply to any business. We use data from a particular source to experiment our model. But we choose to anonymize the name and background of the data of customers. Our data are from a popular e-retailer who has a large e-commerce presence across the north American market, whose e-retail website sells a broad range of products across multiple categories. We use the data to target the prediction of the churn rates for their B2B customers.

We model the customer churn problem as a binary classifier problem where the output of the classifier is a boolean output. A "1" indicates churn and a "0" indicates being active. This problem appears like one of the most common machine learning problem that has been solved with help of classifier algorithms. However, this has rarely been applied to the application and data we are interested in. To solve the problem, we choose a sample set of customers for our study who are similar to each other when referring to their size, spending, behavior, demographics, etc. This is to ensure that our predictive models is applied to the right set of data. The available data is divided into subsets train, test respectively. The ratio for the division is empirically decided to be 7:3.

2.1 Customer Attributes

We start with all possible customer attributes as prospective candidate for features in the feature vector. We make use of data mining tools within the Hadoop Stack to look into conventional and non-conventional channels to mine customer data, sales, behavioral data through interaction that the customer have during his interaction with the e-retailer's website. Bringing the right features inside the feature vector for training the model and ensuring that they have a definite influence on the independent variable is the goal of feature selection. Selection of some features are intuitive and mandatory as they have been used by previous works [YGGH11]. A few others which do not directly impact customer churn may have to be accounted for by the feature selection method to measure their impact on the independent variable. for example, features like spending slope of a customer is certainly a feature that has impact on customer churn and has been used before. Similarly frequency of visits of a customer to the web page, cart abandonment ratio, etc can also be considered a feature that has sizable impact on the problem, but has never been used before. Features like customer spending on a particular product category do not strike at first as a feature that can predict churn, but may indicate if a particular category of products is driving customer churn across the organization.

Our goal is to best evolve a classifier algorithm that can most optimally classify all of the existing data points to lead to an effective prediction.

2.2 Data Sources

Enterprise Data warehouse

All major e-retailers maintain large data warehouses that contain present and historic customer data, marketing data, sales and promotions data. This warehouse is a master database with many replications containing all customer data. Multiple teams across the organization would then use this data to run reports, create business views to suit their requirement. There exists complex relationship hierarchies that define different shipping and billing address and points of contacts and ordering privileges of users. This data warehouse also maintains day-to-day sales data that provides important information about the customer spending behaviors, their periodic purchase patterns that help the customer engagement managers or CRM tools like Salesforce to better interact with the customer to drive sales and win their loyalty.

Data Volume

Table 2.1 shows the size of the data we use to develop our model. As stated before, the e-retailer data we use has a huge customer base. A flavor for volume of this data can be inferred from this table.

Table 2.1 Volume of Customer data

Data	Size
Total B2B Customer base	0.5 Million
Customer Segment considered for model	86K
Total Orders / Day	60K
Total Products Sold / Day	0.35 Million
Total Number of Clicks / Day	8 Million

Big Data Lake

While a lot of structured data like the ones we discussed above is stored in Enterprise data warehouses, there are many other unconventional sources of data like user generated clickstream, social media data, chat data, product review data that is not readily stored in enterprise warehouse system as the volume of this data may be huge to fit into a relational database. Such data can be used by an open source technology for querying and computing on a non-proprietary hardware that is easily scalable.

2.3 Data Mining and Data Preparation

Hadoop Stack

Hadoop [SKRC10], is defined as a framework that allows for distributed processing of large data set across multiple clusters of machines. It encompasses a set of software library modules that provide end to end capabilities for big data processing. Hadoop and its modules are licensed under Apache commons for open source applications for both commercial and academic implementations.

Hadoop uses MapReduce [DG08] as a software framework to develop applications to process a large amount of data in-parallel on large clusters in a reliable, fault-tolerant manner. The framework involves shuffling data as a key-value pair to accomplish most of the operations in a distributed fashion.

The data mining modules from the Hadoop framework discussed below are useful for our application.

Hive

Hive [TSA^{+}10] is defined as a data warehouse infrastructure that creates a high level database layer over existing data residing on Hadoop distributed file system (HDFS) to provide data summarization and ad-hoc querying capabilities to programmers using SQL languages. These queries are in turn converted into optimized Map reduce programs that are spawned across the cluster for execution. Hive is popularly used for data aggregation, Expand Transform Load (ETL) operations, analytics, etc on the hadoop cluster.

Pig

Pig [ORS^{+}08] is defined as a high level data flow language for developing one or more rounds of map reduce jobs that are highly optimized. This tool is mainly used by inexperienced java programmers to develop analytic computation. Pig is popularly used to create custom transformation of data and implement machine learning algorithms.

2.4 Data Mining and Machine Learning Pipeline

Figure 2.1 describes a data pipeline framework which we use to build our model in the desired form using various big data ETL tools like Pig and Hive. The model building is an iterative process (not shown in the figure) that continuously monitors the output of the model and ensures changes in the data mining and data transformation process. The indicative image for the big data lake is a courtesy of www.zdatainc.com. The logos for the apache hive, apache Spark and sci-kit learn are sourced copyrighted under creative common license.

2.4.1 Machine learning libraries

Machine learning research is largely driven by a community of researchers and organizations who have embraced the open source model to drive innovation and collaboration. Thus we have a good set of libraries and packages readily available to experiment different classification models such as Python, Spark (Mllib), R, Weka, etc. We use Sci-kit learn [BLB^{+}13] that provides a number of machine learning modules in the area of regression, clustering, classification, feature selection, cross validation, etc. This library is available in Python.

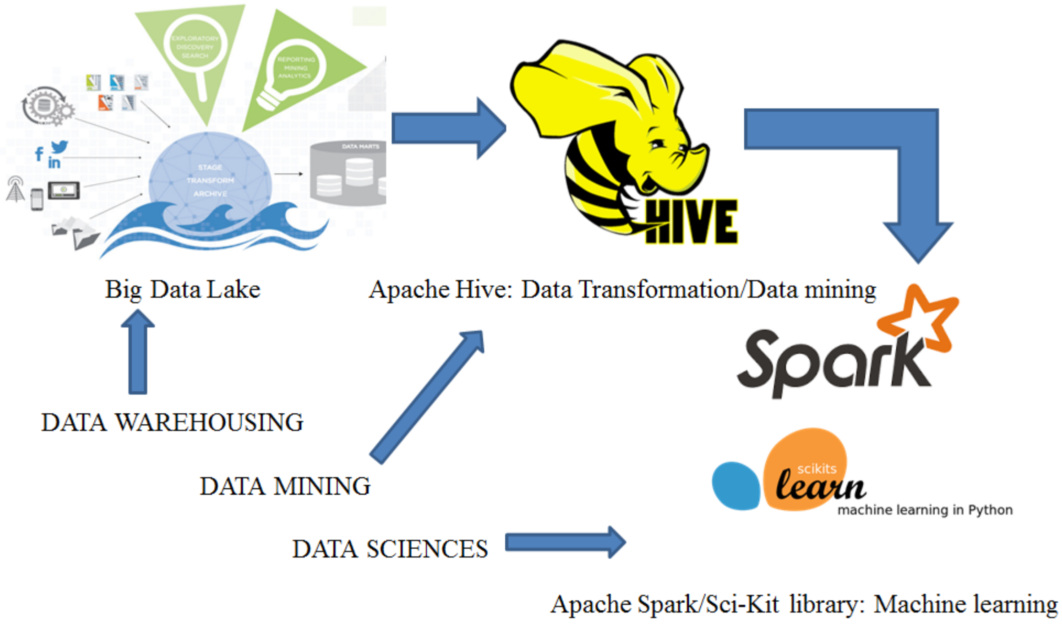

HIVE
Big Data Lake
Apache Hive: Data Transformation/Data mining
Spark
DATA WAREHOUSING
DATA MINING
scikits learn
machine learning in Python
DATA SCIENCES
Apache Spark/Sci-Kit library: Machine learning

Figure 2.1 Data pipeline framework from Apache

2.4.2 Customer churn model

The customer churn model is shown in the figure 2.2. We split the model building process into three discreet phases.

Phase 1 : Data Preparation

The purpose of data preparation step is to process the data to enable the classifier algorithm to handle all variables effectively. We can use a flat file, or a CSV file, to output the processed data for the next stage in the pipeline.

Phase 2 : Data Science model building

The task of applying additional intelligence to the data and bring about meaningful prediction models is the purpose of this process. More details of this process are discussed in chapters three and four.

Phase 3 : Cross validation, Business action and performance observation

The final step in process is to evaluate the accuracy of prediction results and make a comparison between models. Once an optimal percentage of accuracy in prediction is achieved,

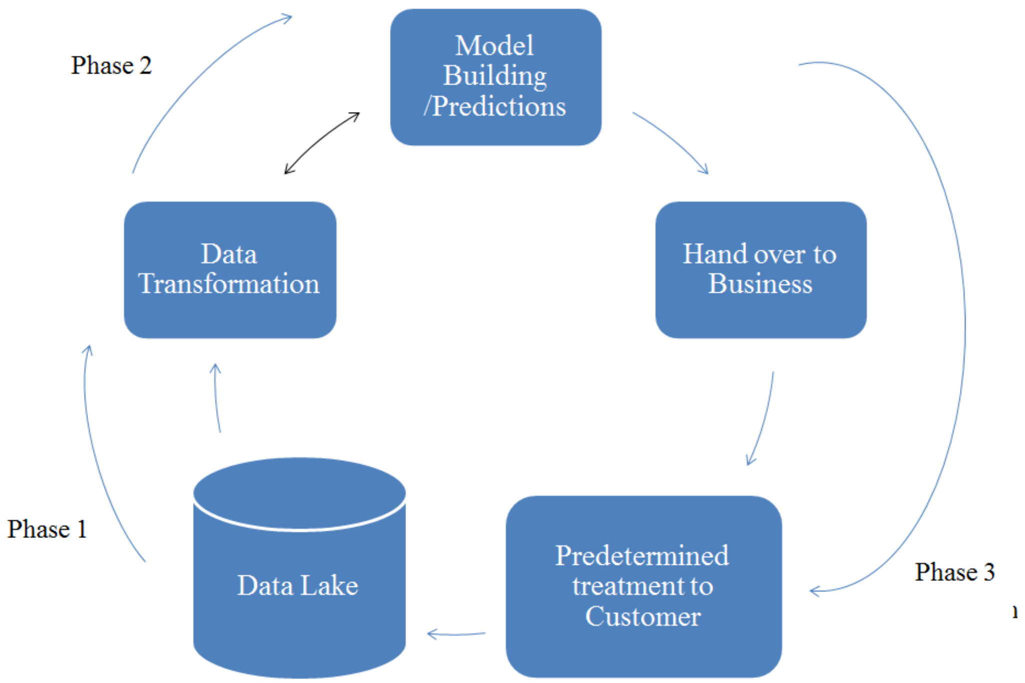


Figure 2.2 Customer churn model life cycle

the model may be used in the production systems. Details of this step are discussed in chapter five.

CHAPTER 3. FEATURE SELECTION

Feature identification and feature selection [GE03, KR92] are important steps for all supervised machine learning algorithms. Domain expertise and past experience help in identifying a set of features that play a role in any outcome prediction including customer churn. Feature Selection prunes the data set by selecting a subset of relevant features from a large pool, thus preventing problems like overfitting [Haw04], poor efficiency, etc. A small number of features would make an algorithm to do a poor job in identifying independent variables and result in high bias, while a large number of features would result in overfitting.

3.1 Previous Work

A study of other industry verticals in customer churn predictive modeling reveals how the features are identified. Earlier customer churn modeling work published revolved around the Mobile Telecom industries. Some of the factors traditionally considered by researchers are demographics, call durations of the customer, spending behaviors, types of plans enrolled, split up of long distance/short distance calling, etc. These features correspond to data that are generated at point of sales or checkout and order confirmation page with respect to an E-commerce or retail industry. These are direct metrics that establish customer behavior generated through a completed transaction. These are explicit factors. The feature mining process has undergone improvements over the years as researchers are now considering at several other behavioral attributes of the customers like the number of calls dropped, network quality experienced by the customer, etc. These are features where the metrics does not flow explicitly through a conventional data channel. Domain experience, aggressive data capturing and active customer feedback channels within the organization help to capture such data and bring them out as new factors into the feature building process.

With respect to the e-retail industry we organize our feature collection process into four broad categories.

Customer Demographics

Customer demographics refers to factors that define the type, scale and other attributes concerned with the customer alone which are independent of the e-retailer. They could depend on the size of the company, the number of employees in the company, financial worth, geographical demography, the type of the industry the customer belongs to, etc. Some of the features concerning customer demography are categorical like geographical location, industry, etc while the other features like size of the company, the number of employees, etc are Ordinal variables. If a particular industry vertical is on the verge of decline, it is not very surprising to see that customers have reduced their spending leading to churn, similar reasons may be attributed to region, etc. Thus customer demographic information plays a vital role.

Enterprise Sales Data

The sales metrics and customer buying pattern are captured from point of sales system or order management systems. Some of the important features part of enterprise sales data are total sales, recency of sales, frequency of sales, categories of products bought, year to date buy ratio, etc.

Customer Interaction Data

These metrics are captured from channels that handle and store customer interaction data, customer survey data, chat data, email marketing, marketing campaign outcomes, etc.

Customer Behavior Data

These metrics are captured from clickstream logs which captures the overall customer interaction with the organization's e-commerce platform. Some of the important metrics that are valuable features to consider are session lengths, cart activity, cart abandonment's, user navigation experience, User Product finding experience, visit to conversion ratio, response to marketing emails, etc.

A few important features that were considered for building the feature set are included below. Owing to the confidential nature of this data we use for the model, we refrain from discussing in detail the individual features that are identified to build feature vector for every customer. Much of it can be inferred from table 3.1.

Table 3.1: Feature variables for Customer feature matrix

Category	Feature	Source	Type	Feature Description
Demographic Features	Customer Size	Customer Warehouse	Static	Size of Customer
	Vertical	Customer Warehouse	Static	Type of Industry that Customer belongs
	Location	Customer Warehouse	Static	Billing address of the Customer
Customer Info	Age	Customer Warehouse	Quantitative	Age of Customer with the organization
	Customer Tier	Customer Warehouse	Static	Business Tier which the customer is enrolled in
	No. of Registered Users	Customer Warehouse	Quantitative	Total number of Registered users enrolled by customer account
Customer Sales	Annual Sales	Customer Warehouse	Quantitative	Avg. annual sales done by the Customer
	YTD Sales	Customer Warehouse	Quantitative	Year To Date Sales done by the Customer
	Spending Slope	Customer Warehouse	Quantitative	Plot of Spend over time
	Total Returns	Customer Warehouse	Quantitative	Total value of goods returned by customer
	Total Orders	Customer Warehouse	Quantitative	Total number of orders placed by Customer

Table 3.1 – *Continued*

Category	Feature	Source	Type	Feature Description
	Total Rebate	Customer Warehouse	Quantitative	Total rebates offered to the Customer
	YOY Sales	Customer Warehouse	Quantitative	Year over Year drop/rise in Sales
Product Sales	Total products	Customer Warehouse	Quantitative	Total count of unique products sold
	Cat_1 sales	Customer Warehouse	Quantitative	Percentage of wallet spent on Cat 1 products
	Cat_2 sales	Customer Warehouse	Quantitative	Percentage of wallet spent on Cat 2 products
. ..	Cat_n sales	Customer Warehouse	Quantitative	Percentage of wallet spent on Cat_n products
Frequency	Frequency orders	Customer Warehouse	Quantitative	Avg. Frequency at which orders are placed
	Frequency visits	Clickstream Logs	Quantitative	Avg. Frequency at which users visit the site
	Days since visit	Clickstream Logs	Quantitative	Num. of Elapsed days since last visit
	Avg. visits per month	Clickstream Logs	Quantitative	Avg. number of visits per month per user.
Behavioral	No. of Active Users	Clickstream Logs	Quantitative	Number of active users from the account.
	Active User Ratio	Clickstream Logs	Quantitative	Ratio of Active/Registered users
	Avg. Page Visits	Clickstream Logs	Quantitative	Avg. number of page visits in a session

Table 3.1 – *Continued*

Category	Feature	Source	Type	Feature Description
Experience	Avg. product Views	Clickstream Logs	Quantitative	Avg. number of Product Viewed in a session
	Avg. Session Length	Clickstream Logs	Quantitative	Avg. length of sessions by users
	Cart/View Ratio	Clickstream Logs	Quantitative	Ratio of Cart Addition over Product Views
	Cart/Buy Ratio	Clickstream Logs	Quantitative	Ratio of Cart Addition over Purchases
	Avg. Abandoned Cart	Clickstream Logs	Quantitative	Avg. worth of Products abandoned in Cart
	Abandoned/Buy Ratio	Clickstream Logs	Quantitative	Ratio of worth of Cart Abandoned over Purchases
	No. of Futile Sessions	Clickstream Logs	Quantitative	Number of Sessions with no orders
	Out of Stock	Clickstream Logs	Quantitative	Number of times user had a product go out of stock
	Difficulty at Checkout	Clickstream Logs	Quantitative	Number of times user had an issue at checkout
	Null results	Clickstream Logs	Quantitative	Number of times product Search yielded null results

3.2 Data Cleaning and Data Pruning

In the process of building a customer feature matrix ingesting many different customer parameters into the system, there are often certain missing values for few features. As an example, a feature like *Abandoned Cart Worth* for a traditional retail customer who might not have an online presence is null, however it may not be correct to substitute this value as 0. To deal with the missing values it is a common practice to often substitute them with a more dependable value which could be either the mean, median, most frequently occurring value, etc in the distribution. This activity, called imputing, handles the missing values. In our work we experimented with all of these techniques and choose suitably to account for a missing value.

3.3 Feature Selection and Feature Reduction

Feature selection refers to the process of selecting a subset of relevant features from a pool of features that are initially available. This process reduces the number of features as input to the model, and therefore, reduces the data acquisition and computation cost. Secondly, it yields more accurate results. As described in [KR92], Feature selection, as a preprocessing step to machine learning, has been very effective in reducing dimensionality and irrelevant data, increasing learning accuracy and improving result comprehensibility." Feature selection includes individual or subset selection. Individual feature selection ranks features separately according to a particular metric where the subset selection takes into account the interaction and correlation among features. The final goal of feature selection is to have a minimum number of features that is good enough to capture all of the trends and variations in the output. It is important to select the right feature set before implementing an effective algorithm.

The important factors to consider when removing a feature from the feature vector include the noisy nature of the feature, variance, correlation among features, F-anova scores, Regularization, etc. The target of building a feature matrix is not solely to accumulate a a number of features, but to actually gather features that have sizable impact on the outcome of the classifier.

3.3.1 Features with low threshold

The features with low variance can be removed [YL04] if a feature fails to satisfy a preset threshold, as they have no impact on the classification. This approach can be applied to both supervised and unsupervised learning. A feature is only considered effective when its Variance is non-zero and exceeds a certain threshold.

High Variance leads to low SSE (Sum of Square Errors) while high bias leads to simplicity. High variance may lead to better performance in the test data set, it leads to a complicate model with a higher model building time. High bias leads to poor performance on the test data set even though it may offer a simplistic model. Finding an optimum fit between the two characteristics is an ideal fit for the feature set.

3.3.2 Features with high correlation: Pearson correlation coefficient

Pearson Correlation [Hal99] measures the linear relationship in any two distributions assuming that they are a normal distribution. The correlation results vary from -1 to $+1$ with 0 implying that there is no correlation. We compute the Pearson correlation and remove features that have high correlation between them since one of them is redundant. This redundancy certainly impacts the accuracy of the classification algorithms.

The Pearson Correlation coefficient between two random variables X and Y is defined as

$$\rho(X,Y) = \frac{\mathbf{Cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}}. \tag{3.1}$$

Multicollinearity occurs when there is high correlation between the predictor variables leading to errors like unstable estimates of regression co-efficients. Researches use tests like Variance inflation factors (VIF) to test if multilinearity can be safely ignored. These tests are beyond the scope of this work and hence ignored. We adopt a baseline approach of looking at instances of high correlation between predictor variables and found no significant correlation between the predictor variables.

3.3.3 Imbalance in output class labels

An important observation from our data-sets is the imbalance in data. On an average about 5-10% customers churn year-on-year basis depending on the segment we are looking at. This imbalance in distribution consisting returning/non-returning customers is a good recipe for learning algorithms to classify a large number of customers under returning and still attain high overall accuracy. There are several works carried out in the past [BVdP09] that specifically focus on handling imbalance in the data leading to skew the predictions of the model. Thus we employ an in depth cross-validation technique based on confusion matrix, threshold shift and ROC curve to arrive at the best algorithm to rule out such a bias in our model. We also assign weights to the feature vectors that are inversely proportional to class frequencies in the input data as shown in equation 3.2. Here, n_samples is the total number of samples in the data-set, n_classes represents the total possible class outcomes from the output label, which in this case is two. The count_of_occurrence Y_i represents the total number of occurrences of samples belonging to a given class whose weights we are interested in calculating.

$$Weight\_class\ Y_i \propto \frac{n\_samples}{(n\_classes)(count\_of\_occurence\ Y_i)} \tag{3.2}$$

3.3.4 Uni-variate feature selection

We consider a few classic feature selection and feature reduction techniques after employing the baseline methods discussed in the preceding section. The uni-variate feature selection techniques [SIL07] select the best features based on uni-variate statistical tests. An important assumption these techniques make is that they consider all the features as independent of each other. Some of the most popular techniques of uni-variate feature selection are $Chi - Square\ tests$, $F - ANOVA$ classification tests. While Chi-Squared tests are best suited in dealing with non-negative features, categorical or sparse data, it is less suited to handle our feature matrix without several modifications. Therefore we consider $ANOVA$ based F-Classification test [SIL07] to identify important features from the feature matrix.

3.3.4.1 ANOVA F-Classification Test

We use the ANOVA F-test [SIL07] for scoring individual features to the transformed feature matrix after applying threshold variance and pearson correlation techniques. The F-ANOVA considers one feature at a time to see how well each continuous variable predicted the class label. The importance value of each variable is calculated as F score of F-statistic test of association with the predictor and class label which can also be said target variable.

The F-value is defined as follows:

$$F = \frac{MS_R}{MS_E} \tag{3.3}$$

here, MS_R is the "Mean square regression" and MS_E is the "Mean square error". While MS_R indicates the between group variability. MS_E represents the within group variability. The statistical tests concluding feature significance with the independent variability determines if the between group variability is a higher than the within group variability. The sample variance of predictor X for the target class $Y = J$ is given by the following equation:

$$S_j^2 = \sum_{i=1}^{N_j}((x_{ij} - \bar{x_j})^2)/(N_j - 1) \tag{3.4}$$

here N_j is the number of cases with $Y = j$. $\bar{x_j}$ is the sample mean of predictor X for target class $Y = j$. $\bar{x}$ referred to as a grand mean of predictor X given by the following

$$\bar{x} = \sum_{j=1}^{j}(N_j\bar{x_j})/N \tag{3.5}$$

$$F = \frac{\sum_{j=1}^{J}(N_j(\bar{x_j} - \bar{x})^2)/(J-1)}{\sum_{j=1}^{J}(s_j^2(N_j - 1))/(N-1)} \tag{3.6}$$

Once the F Value for all of the independent predictors are calculated, we employ $K\%$ percentile approach or the K best feature approach to select the features that have the most impact on the independent variable.

It is evident from Figure 3.1 that many conventional feature variable were scored by the algorithm as features that influence customer churn. Some of the these include days between purchase, spending ratio for Year-on-Year, total sales, number of online visits, number of customer cross-shopping (dotcom Visits), etc. However, it is definitely important to notice that there are several unconventional features brought out through this work like futile online sessions, Total worth of cart abandoned online, several product categories like cat_11, cat_12, cat_13 also influencing customer churn. The above features not only increases the prediction accuracy of our model, but also provides a valuable input to Business to look into reasons why certain product categories, prices or customer experience contribute to drive customer churn.

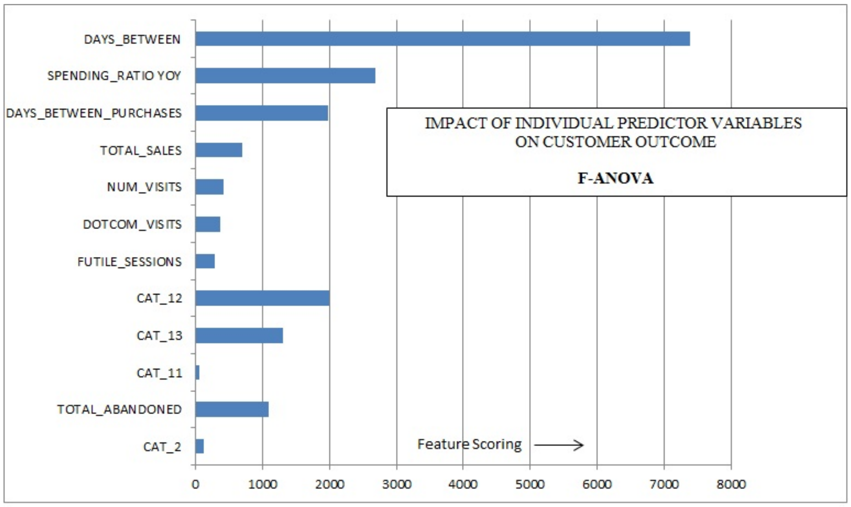


Figure 3.1 Customer feature Scoring through Univariate F-score

We rank these features by their scores and iterate the final classifier with variable number of features (Top N) to arrive at the best solution.

3.3.4.2 Principal Component Analysis

PCA [Jol02] is a multivariate feature reduction technique that reduces the dimension of the data by finding the first $'s'$ orthogonal linear combinations of the original variables with the largest variance. PCA is defined in such a way that the first principal component has the largest possible variance. Each succeeding component in turn has the next highest variance possible under the constraint that it is orthogonal to the preceding components.

Employing PCA algorithm to select the first N features orthogonal to the original feature set had poor results when this transformed feature set was applied to a classifier in predicting the customer outcome. Hence we decided not to pursue PCA further.

3.3.4.3 Regularization based Feature Selection

Adding regularization [Ng04] to learning algorithm is one of the ways to do feature selection and avoid the problem of over-fitting specially when we are handling a lot of sparse features. Since Regularization penalizes the complexity of learning model using L1, L2 norm. Having

Table 3.2 Result of L1 regularization on Feature Matrix

Coefficients	Feature Variables
0.164565497	CAT_2
0.345285729	CAT_5
0.796459688	CAT_6
-2.066845099	CAT_12
-1.316095705	CAT_13
0.098768507	CAT_16
-1.16188276	CAT_17
-0.145851299	FUTILE_SESSIONS
-0.145851299	NUM_VISITS
0.1245	TOTAL_REBATE
0.285193414	DOTCOM_VISITS
4.470811284	DAYS_BETWEEN_PURCHASES
0.93979978	RATIO_CART_ABANDONED
0.5564536	SPENDING_SLOPE

sparse solutions decreases the complexity, reduces the number of features and yields better prediction.

Previous studies [Ng04] have shown that L1-based regularization is superior than L2-based when there are many features. The complexity of L1-regularization logistic regression is logarithmic in the number of features. The sample complexity of L2-regularization logistic regression is linear in the number of features.

We discuss more details on regularization for a classification problem in chapter four when we discuss logistic regression in detail. Table 3.2 shows how the coefficients of each feature stack up against others when L1 regularization is applied. The features with lower co-efficients or close to zero co-efficients have marginal impact on the outcome of the classifier and hence ignored from this table.

CHAPTER 4. BINOMIAL CLASSIFICATION

In this chapter, we discuss the most important component of machine learning application pipeline for our problem, that is the Binomial Classification to predict if a customer would abandon or stay with the organization.

Through empirically derived assumptions, grid and randomized searches to optimize the parameters of the model, cross validation techniques and prior understanding of the nature of algorithms in handling data, we arrive at the best algorithm to experiment and finalize. We discuss this task in deeper detail in the following sections.

4.1 Previous Work

The most commonly used algorithms used by Data scientists and data analysts are listed by Rexer Analytics survey through survey polls, KDD cup submissions, etc. Figure 4.1 shows the results of these analysis.

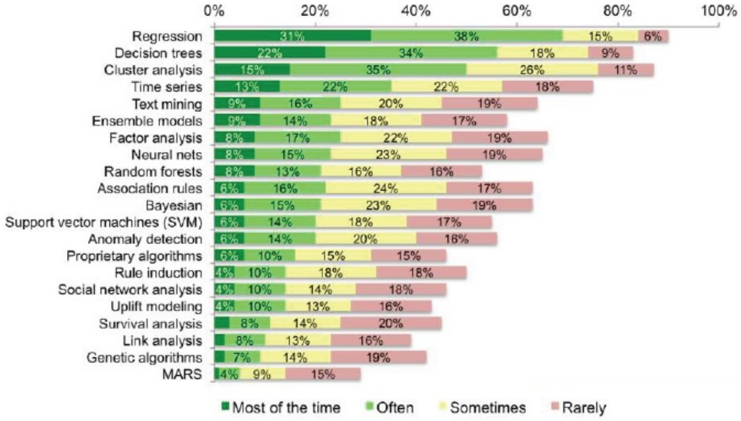


Figure 4.1 Rexer Analytics Survey on most Commonly used Algorithms

It is evident from the above figure that Logistic Regression, Decision Trees, Ensemble methods, Bayesian, SVM, etc are some of the most popular supervised algorithms in the decreasing order of their popularity used for classification problems. Upon studying customer churn studies carried out earlier, we find that these algorithms were consistently used in all these studies as well.

4.2 Naive Bayes Classifier

Naive Bayes classifiers [Ris01] belongs to the family of probabilistic classifiers based on applying Bayes theorem with assumptions of independence between the features. Naive Bayes learning generates a probabilistic model given a training set of instances. Each data point is represented as a vector of features $[x_1, x_2, x_3, ..., x_d]$. The task is to learn from the data to be able to predict the most probable class $y_i \epsilon Class_i$ of a new instance whose class is unknown. We first introduce the Bayes Theorem which describes the probability of an event, based on conditions that might be related to the event defined by Eqn. 4.1.

$$p(y_j \mid x) = \frac{p(x \mid y_j)p(y_j)}{p(x)} \tag{4.1}$$

where $p(y_j \mid x)$ is the probability of an instance x being in class y_j

$p(x \mid y_j)$ is the probability of generating instance x given class y_j

$p(y_j)$ is the probability of occurrence of class y_j

$p(x)$ is the probability of x occurring.

Equation 4.1 serializes to

$$P(y_i \mid x_1, x_2, ..., x_d) = \frac{P(y_i)P(x_1, x_2, x_3, ...x_d \mid y_i)}{P(x_1, x_2, x_3, ..., x_d)} \tag{4.2}$$

Naive Bayes employs the Bayess theorem to estimate the probabilities of the classes. Here, $P(y_i)$ is the predetermined probability of class which is estimated as its occurrence frequency in the training data, while $P(y_i \mid x_1, x_2, ..., x_d)$ is the posterior probability of class after observing the data. $P(x_1, x_2, x_3, ...x_d \mid y_i)$ denotes the conditional probability of observing an instance with the feature vector $[x_1, x_2, x_3, ..., x_d]$ among those having class y_i. Finally, $P(x_1, x_2, x3, ..., x_d)$ is the probability of observing an instance with the feature vector $[x_1, x_2, x_3, ..., x_d]$ regardless of the class.

Since the sum of the posterior probabilities, $\sum_{y_j \epsilon \ class_c} P(y_i \mid x_1, x_2, ..., x_d) = 1$, the denominator on Eqn. 4.2 right hand side is a normalizing factor and thus can be omitted.

$$P(y_i \mid x_1, x_2, ..., x_d) = P(y_i)P(x_1, x_2, x_3, ...x_d \mid y_i) \tag{4.3}$$

A data point is labeled as a particular class if it has the highest posterior probability $y(class)$ for a given class among all available classes. This is given by

$$\underset{y_i \epsilon class}{\arg\max} \, P(y_i)P(x_1, x_2, x_3, ...x_d \mid y_i) \tag{4.4}$$

In order to estimate the term $P(y_i)P(x_1, x_2, x_3, ...x_d \mid y_i)$ by counting frequencies, one needs to have a huge training set where every possible combinations $[x_1, x_2, x_3, ..., x_d]$ appear many times to obtain reliable estimates. Naive Bayes solves this problem by its Naive assumption that features that define instances are conditionally independent given the class. Therefore, the probability of observing the combination $[x_1, x_2, x_3, ..., x_d]$ is simply the product of the probabilities of observing each individual feature value $P(x_1, x_2, x_3, ...x_d \mid y_i) \prod_{i=1}^{d} P(x_i \mid y_i)$. Substituting this approximation into the main equation above to derive the Naive Bayes classification rule.

$$\underset{y_i \epsilon class}{\arg\max} \, P(y_i) \prod_{i=1}^{d} P(x_i \mid y_i) \tag{4.5}$$

As discussed above, for a nominal feature, the probability is estimated as the frequency over the training data. For continuous feature, there are two solutions. The first one is to perform discretization on those continuous features, transferring them to nominal ones. The second solution is to assume that they to follow a normal distribution.

4.3 Logistic Regression

Logistic regression [HJL04] is a representative of discriminative classifier that learns a direct map from input x to output y by modeling the posterior probability $P(y \mid x)$ directly. The parametric model proposed by logistic regression is of the form explained below. One of the simplest equations with which we can represent a logistic regression is a sigmoid function given by eq. 4.6.

$$\sigma(x) = \frac{1}{1 + e^{-z}} \tag{4.6}$$

We then define the loss function that defines the 0-1 losses for the model.

$$Loss_{0/1}(z) = \begin{cases} 1, & \text{if } z < 0 \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

let $y \epsilon \{-1, 1\}$ and $z = y.w^T x$. Here z is positive if y and $w^T x$ have same sign, else negative otherwise.

$$P(y = -1 \mid x) = \frac{1}{1 + exp(w_0 + \sum_{i=1}^{d} w_i x_i)} \tag{4.8}$$

$$P(y = 1 \mid x) = 1 - P(y = -1 \mid x) \tag{4.9}$$

The main task of logistic regression is minimizing w so that the average $0 - 1$ loss is minimized over the training points.

$$\min_w \sum_{i=1}^{n} l_{0/1}(y^{(i)}.w^T.x^i) \tag{4.10}$$

$$w = [w_0, w_1, w_2, ..., w_d] \leftarrow \arg\max_w \prod_k P(y^{(k)} \mid x^{(k)}, w) \tag{4.11}$$

Upon plotting the 0/1 loss function we transform the regression model into a logistic function whose values vary from 0 to 1 as z goes from $-\infty$ to $+\infty$.

$$l_{log}(z) = \log(1 + e^{-z}) \tag{4.12}$$

We further solve for w using gradient descent rule. Although designed for continuous features, logistic regression can still handle nominal feature and missing values effectively.

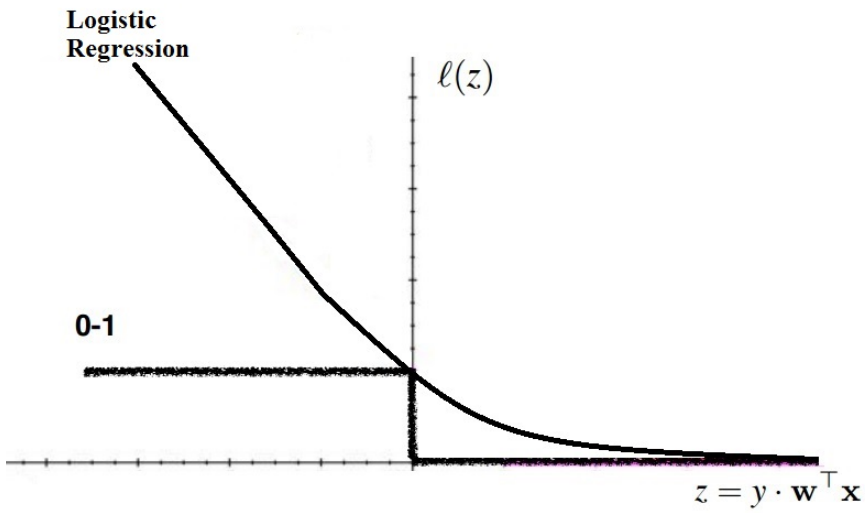


Figure 4.2 Plot of the Logistic Losses

4.3.1 Logistic regression with regularization

Adding Regularization to the learning algorithm avoids over-fitting by removing the unwanted features from the data set. The most common ways to achieve regularization is based on $L1$ and $L2$ norm resulting in sparseness thus reducing complexity.

Regularization based logistic regression learns mapping (w) that minimizes logistic loss on training data with regularization term. For Regularization in Logistic Regression, we use maximum likelihood function as given in 4.13.

$$\min_{w} \sum_{i=1}^{n} l_{log}(y^{(i)}.w^T.x^i) + \lambda \|w\|_2^2 \tag{4.13}$$

Equation 4.13 has two components to it, the training log-loss function and the model complexity. The λ from the model complexity component is the regularization parameter. This determines how much of w parameters are inflated. Using Eqn. 4.13 as the cost function, we can smoothen the output of our hypothesis function to reduce over-fitting. If w is chosen to be too large, it may smooth out the function too much and cause under-fitting. We frequently observe that $L1$ regularization in many models causes many parameters to reduce to 0, resulting in the parameter vector to be sparse.

4.4 Support Vector Machines (SVM)

Another popular classification algorithm used in supervised machine learning is the Support Vector Machines (SVM) [Joa98]. The goal of SVM is to find the optimal separating hyper plane which maximizes the margin of the training data by dividing the n-dimensional space representation of the data into two regions using a hyperplane [YM07]. The SVM methodology also has solid underpinnings in statistical learning theory. The methodology can be applied successfully to many linear and non-linear classification problems.

There are many kernel-based functions such as linear kernel function, the normalized poly kernel, polynomial kernel function, Radial Basis Function (RBF) or Gaussian Kernel and Hyperbolic Tangent (Sigmoid) Kernel sigmoid function that can be implemented in SVM [SS01]. SVM's output a class label, either positive or negative for each sample in our case of binomial classification: In order to compute metrics like ROC curve, etc. We can also find the distance between from hyper-plane that separate classes. SVM has many advantages such as obtaining the best result when deal with the binary representation, able to dealing with low number of features.

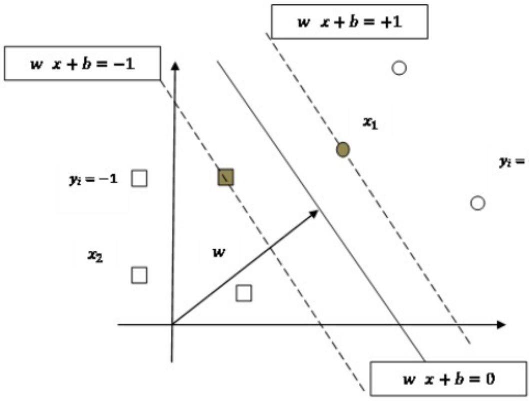


Figure 4.3 Plot of Binomial Classification using SVM

SVM classifiers [EIO14] utilize the hyper-plane to separate classes. Every hyper-plane is characterized by its direction (w) and (b) which is its exact position in space or threshold. Let us consider x_i is an input vector of dimension N. We would have a set of training data along with the labeled output $y_i \epsilon (-1, 1)$.

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), ...(x_k, y_k) \tag{4.14}$$

The decision function for the above problem is of the form $f(x, w, b) = sign((w\dot{x}_i) + b), w\epsilon R^d, b\epsilon R$ where d is the dimension of the class output. The region between the hyper-plane that separates the two classes in this case is called the margins. Width of the margin is equal to $\frac{1}{2}\|w\|$ and the underlying goal is to maximize the margin between the hyper-plane.

To satisfy this maximization, we need to minimize $f(w, b) = \frac{1}{2}\|w\|^2$. Minimizing the cost is a trade-off issue between a large margin and a small number of margin errors. The final solution to this optimization problem can be formulated as below

$$w = \sum_{(i=1)}^{N} \lambda_i \gamma_i \chi_i \tag{4.15}$$

Equation 4.15 shows the weighted average of the training features. Here λ_i is a lagrange multiplier of the optimization task and γ_i is a class label. Values of λ's are non zero for all points lying inside the margin and on the correct side of the classifier.

To prevent over-fitting by permitting some degree of miss-classifications, a cost parameter C controls the trade off between allowing training errors and rigid margins. Increasing the value of C increases the cost of miss-classifying points and may result in a model that may not generalize well. For our experiments we use a SVM classifier with linear kernel (SVM-L).

4.5 Decision Trees

Decision trees [Qui86] was first introduced in 1966 [HMS66] and currently has become one of the most widely used and researched machine learning methods especially for applications like image recognition, artificial intelligence and multi-label classification. As white boxes, decision trees generate interpretable and understandable models. Induction of decision tree involves building a tree top-down using divide and conquers strategy. The ultimate goal is recursively partition the training set, choosing one feature to split each time until all or most of instances in each partition belong to the same class.

A decision tree consists of following main elements:

1. Root
2. Branches and leaves

The branches correspond to possible outcomes of feature value and finally leaves that specify expected value of the class. Each leaf is assigned to the class that has the majority of instances inside it. To classify a new instance, one starts at the root and follow a path lead by the nodes and branches downward, end at a particular leaf and the instance is assigned a class specified by the leaf.

4.6 Random Forest Classifier

Random Forest classifier [Bre01] is a popular choice for both linear and non-linear classification problems that is relatively new. It belongs to a larger class of machine learning algorithms called ensemble methods.

4.6.1 Ensemble learning : gradient boosting

Ensemble learning, refers to a technique which involves combination of several models to solve a single predictor. It works by generating multiple classifiers/models which learn which all make independent prediction. Those predictions are then combined into a single prediction that should be as good or better than the prediction made by any one classifier. Random forest is a type of ensemble learning which uses an ensemble of decision trees.

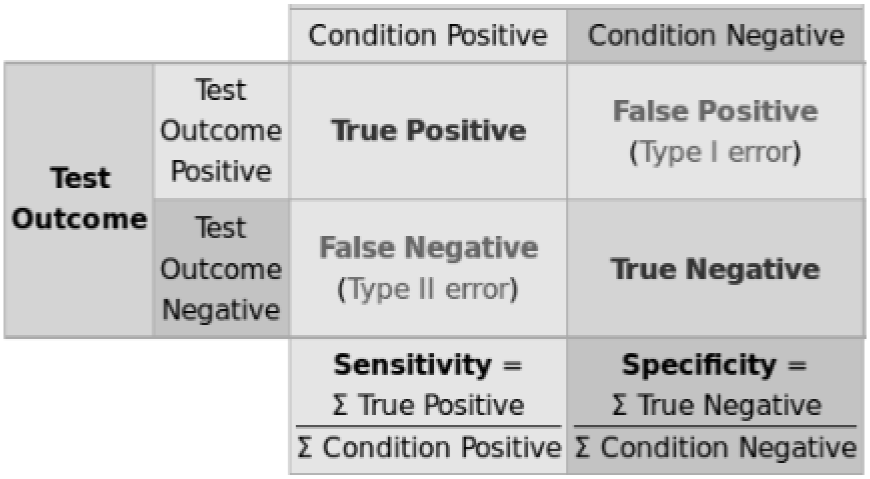
Gradient boosting [Fri01] uses a set of weak learners and delivers improved prediction accuracy. The outcome of the model at an instance $'\text{t}'$ is weighed based on the outcome of previous instant $'\text{t-1}'$. In Gradient descent shortcomings in predictions are identified by negative gradients. At each step, a new tree is fit to negative gradients of the previous tree.

CHAPTER 5. CLASSIFIER EVALUATION AND RESULTS

5.1 Classifier Performance Evaluation

Although much has been inferred about how different algorithms work to solve the classification problem, it is only after observing results, one can make accurate assumptions about the best algorithm that can be used for the given prediction problem. Some algorithms are well suited for a few types of domains while that may not hold true in all cases. It is mostly the underlying data that drives the results of different algorithms. To decide a good algorithms we essentially look at the metrics like accuracy, generality and confidence of prediction. For a classification problem, researches have been for long using confusion matrix [DG06] for studying possible outcomes when a classifier is applied on a set of class instances. Since the customer churn is a binomial classification, we are presented with four possible outcomes of prediction.

5.1.1 Confusion matrix



		Condition Positive	Condition Negative
Test Outcome	Test Outcome Positive	**True Positive**	**False Positive** (Type I error)
	Test Outcome Negative	**False Negative** (Type II error)	**True Negative**
		Sensitivity = $\frac{\Sigma \text{ True Positive}}{\Sigma \text{ Condition Positive}}$	**Specificity =** $\frac{\Sigma \text{ True Negative}}{\Sigma \text{ Condition Negative}}$

Figure 5.1 Confusion Matrix :Binomial Classification (Source : Wikipedia)

From the confusion matrix shown in Fig 5.1, there are exactly 4 possible outcomes from a binomial classifier model. A correctly classified instance is counted as a true positive (TP) or a true negative (TN) if its actual class is positive or negative respectively. A positive instance which is wrongly classified as negative is counted as a false negative (FN). A negative instance which is wrongly classified as positive is counted as a false positive (FP). The total number of positive instances in the dataset is $T = FN + TP$ and the total number of negative instances is $F = TN + FN$. Based on a confusion matrix, the most common evaluation metrics are overall accuracy, true positive rate and false positive rate.

$$Accuracy = \frac{TP + TN}{N + P}$$

The true positive rate (also known as hit rate or the Precision) is the proportion of positive instances that a classifier captures.

$$Precision = \frac{TP}{P}$$

The Recall is the ratio of number positive instances(TP) over the sum of true positives (TP) and False negatives (FN).

$$Recall = \frac{TP}{TP + FN}$$

The false positive rate (also known as false alarm rate) is the proportion of negative instances that a classifier wrongly flagged as positive.

$$FPRate = \frac{FP}{N}$$

More than the accuracy, we are interested in increasing the TP rate of our classifier. A customer being a returning Customer wrongly classified as non-returning by the classifier thus falling in False Positive quadrant has lesser impact than an abandoning customer wrongly classified as returning customer thus falling in False Negative quadrant. In this case, we ignore a potential customer who might abandon the company in the near future.

5.1.2 Reverse operating characteristics (ROC) curve

When TP rate is plotted as against FP rate as seen in fig 5.2, one obtains a receiver operating characteristics (ROC) graph [DG06]. Each classifier is represented by a point on ROC graph. A perfect classifier is represented by point $(0, 1)$ on ROC graph which classifies all positive and negative instances correctly with 100% TP rate and 0% FP rate. The diagonal line demonstrates classification that is based completely on random guesses. In that case, one can achieve the desired TP rate but unfortunately also gain equally high FP rate. The major goal of churn prediction is to detect churn. Therefore, a suitable classifier is the one having high TP rate and low FP rate given that churn is the positive class. Such classifier is located at the upper left corner of ROC graph.

A classifier provides output in probabilistic form, with exceptions for a few algorithm $\Pr(churn \mid x)$, the probability that an instance belongs to the positive class. If this probability is above the predefined threshold $\Pr(churn \mid x) > \Theta$, an instance is classified as positive, otherwise negative. A classifier using high value for Θ is considered c̈onservative·: It classifies positive instances only with strong evidence so it makes few FP mistakes but at the same time has low TP rate. A classifier using low value for threshold is considered liberal. It classifies positive instances with weak evidence so it achieve high TP rate but also makes many FP mistakes. When the performance of a classifier is plotted on ROC graph with value of varied from 0 to 1, an ROC curve will be formed. It demonstrates the trade-off between TP rate and FP rate.

Figure 5.2 shows an example of ROC curve. The red points are random guess classifiers. The pink and yellow points represent the performance of two classifiers using different values of Θ. The higher is Θ, the more conservative a classifier becomes. Conservative classifiers locate at the lower part of ROC graph. In contrast, the lower is Θ, the more liberal a classifier becomes. Liberal classifiers locate at the upper part of ROC graph. Given two ROC curves, the one that is further to the left of the random diagonal is preferred. For this reason, area under ROC curve (AUC), a quantity that measures the overall average performance of a classifier is introduced. The advantage of AUC is unlike many other evaluation metrics such as the overall

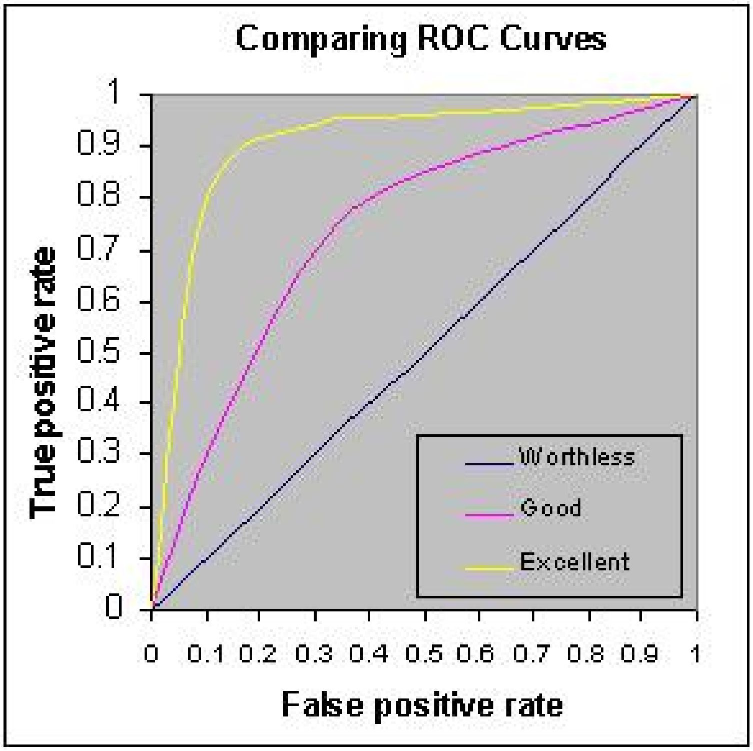


Figure 5.2 Example of an ROC Curve

accuracy, AUC is not affected by the class distribution, ratio. In the case of unbalanced class distribution such as in churn prediction data, AUC yields a fair measure for model comparison. Based on the classification result, the marketing teams focuses on customers that are classified as positive or at risk of churn. However, the business will probably be unable to react to all positive classified instances due to the lack of resources. Besides that, quality is more important than quantity. The question is not only how many percent of defaulters, in this case inactive customers a model can covers but also with what reliability. A classifier which covers 30% of defaulters with 90% reliability may be more preferable than the one which covers 50% of churners with 60% reliability. The choice is up to us to evaluate the cost of ignoring customers in churn risk versus the cost of offering unnecessary special treatment for customers that will are not likely to leave. [TXH$^+$04] suggests a formula to calculate the confidence level of a prediction.

$$Confidence\ of\ Class\ Yi = \frac{\Pr(churn \mid x) - 0.5}{0.5} \tag{5.1}$$

5.2 K-fold Strategy for Cross Validation

We used 5-fold CV by randomly splitting the training dataset (D) into five mutually exclusive subsets $(D1, D2, D3, D4, D5)$ of approximately equal size. Each classification model was trained and tested five times, where each time $(t \in 1, 2, 3, 4, 5)$, it was trained on all except one fold $(D\ Dt)$ and tested on the remaining fold (Dt). The accuracy and AUC measures were averaged over the particular measures of the five individual test folds which we shall see in the further section

5.3 Results from Classifier Models

We experiment with a mixture of both parametric and non-parametric classifiers as discussed in Chapter four. Logistic Regression with L1 regularization, SVD and Gradient Boost classifier have the highest accuracy in predicting the customer churn from the data-set. More importantly their precision scores are at the highest compared to other classifiers.

An example of a bad classifier, miss-classifying the entire distribution of abandoning customers as FN may still achieve an overall accuracy greater than 90% owing to the imbalance in the distribution of data. But such a classifier is of no use to us as we are interested in our precision rates that determines how effectively does a model predicts churn.

Let us consider how a variant of Ensemble family of classifiers like Gradient boost classifier performs on a given sample test data-set. The confusion Matrix for the prediction is as shown in table 5.3. The overall accuracy for this classifier is 90.61%. Although, this accuracy is unusually high for a learning classifier in predictive analytics, since we prune the noisy features using feature selection algorithm, apply weights for the learning algorithm and tune the hyper parameters for the model continuously, the classifier is able to attain this level of accuracy. The precision for the model is close to 75% which indicates that we are able to identify every three out of four churning customer.

We observe from Table 5.1 and 5.2 that the accuracy and other indicators for several classifiers used to predict customer churn. As is evident from the tables, Gradient boost classifier outperforms SVM and regularized logistic regression to give the highest accuracy and precision. Running this algorithm on a feature-set which has already undergone feature selection increases the predicting accuracy further as seen in 5.2.

5.4 ROC Curve

We have discussed that a larger area under the curve (AUC) is indicates a better estimator . The ”steepness” of ROC curves is also important, since it is ideal to maximize the true positive rate while minimizing the false positive rate.

The below plots for Logistic Regression and SVM classifiers shows the ROC response for different datasets, created from K-fold cross-validation techniques. Taking all of these curves, it is possible to calculate the mean area under curve, and see the variance of the curve when the training set is split into different subsets. This roughly shows how the classifier output is affected by changes in the training data, and how different the splits generated by K-fold cross-validation are from one another.

5.4.1 Regularized logistic regression: L1-norm

The graph 5.4 shows the ROC curve for L1-norm regularized logistic regression with K-fold cross validation technique. Except Fold 0 which has an area of 0.71, all of the other folds have fairly consistent AUC indicating that our dataset is fairly robust. The mean ROC as indicated by the plot is 0.77. It is apparent from the graph that ROC fold 2 strategy has the best efficiency for the model.

5.4.2 SVM classifier with F-anova feature selection

The graph 5.5 shows the ROC curve for SVM Classifier with K-fold cross validation technique. The probabilistic estimation of classes for an SVM was made available not until recently by the work proposed in [Pla99], called the Platt scaling to optimize internal variables to also produce a probabilistic score. As with the earlier case, all of the folds for different combination of Test, Training have fairly consistent AUC. The mean ROC as indicated by the plot is again 0.77. Once the model is finalized by tuning the hyper-parameters, the area under the ROC Curve can be used to tweak the threshold probability such that we can tune the classifier to return the best predictions for a given quadrant in the confusion matrix as desired by us. This tweak in threshold probability identified by TP Rate and FP Rate populates the graph for the ROC curve.

We can experiment bringing down the threshold to lower level than default value which is 0.5 which helps us decide the optimal point for the model depending on how accurately we want to identify the churning customers at the cost of mis-classifying non-churning customers. As we bring down the threshold probability, our model becomes more liberal which increases the recall value for the classifier.

5.5 Best Performing Classifiers

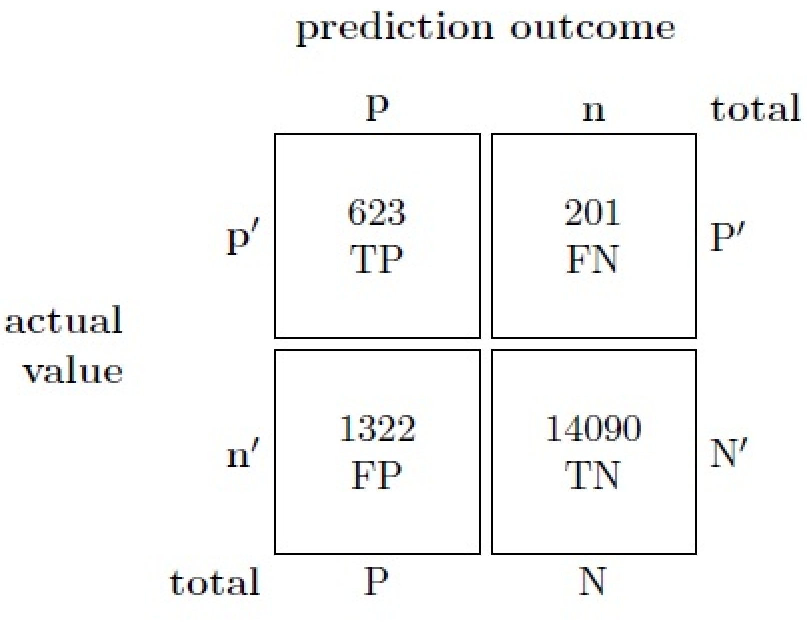
The following classifiers are the best performing classifiers in the order of their appearance at predicting the customer churn for the e-retailer in consideration.

Gradient Boost Ensemble Classifier

SVM Classifier with linear Kernel

Regularized Logistic Regression with L1 norm

Some of the other classifiers which were experimented on this data include Naive Bayes, Artificial Neural Networks, KNN classifier, Random Forests, Decision Tree, etc.



		prediction outcome		
		p	n	total
actual value	p′	623 TP	201 FN	P′
	n′	1322 FP	14090 TN	N′
	total	P	N	

Figure 5.3 Confusion Matrix for Gradient Boost Classifier

Table 5.1 Classification without Feature Selection

Classifier Algorithm	**TN**	**FN**	**FP**	**TP**	**Accuracy**	**Precision**	**Recall**
Naive Bayes	6004	484	246	86	0.892	0.259	0.1508
Support Vector Machines	5575	913	114	218	0.849	0.656	0.192
Random Forest	6428	5	326	6	0.943	0.018	0.545
Gradient Boost	5875	613	68	264	0.900	0.795	0.301
Logistic Regression	5423	921	142	190	0.835	0.572	0.208

Table 5.2 Classification with F-Anova Feature Selection

Classifier Algorithm	**TN**	**FN**	**FP**	**TP**	**Accuracy**	**Precision**	**Recall**
Naive Bayes	6372	121	286	41	0.940	0.125	0.253
Support Vector Machines	5520	968	68	259	0.850	0.79	0.211
Random Forest	6481	12	326	1	0.950	0.003	0.077
Gradient Boost	6903	590	53	274	0.917	0.838	0.317

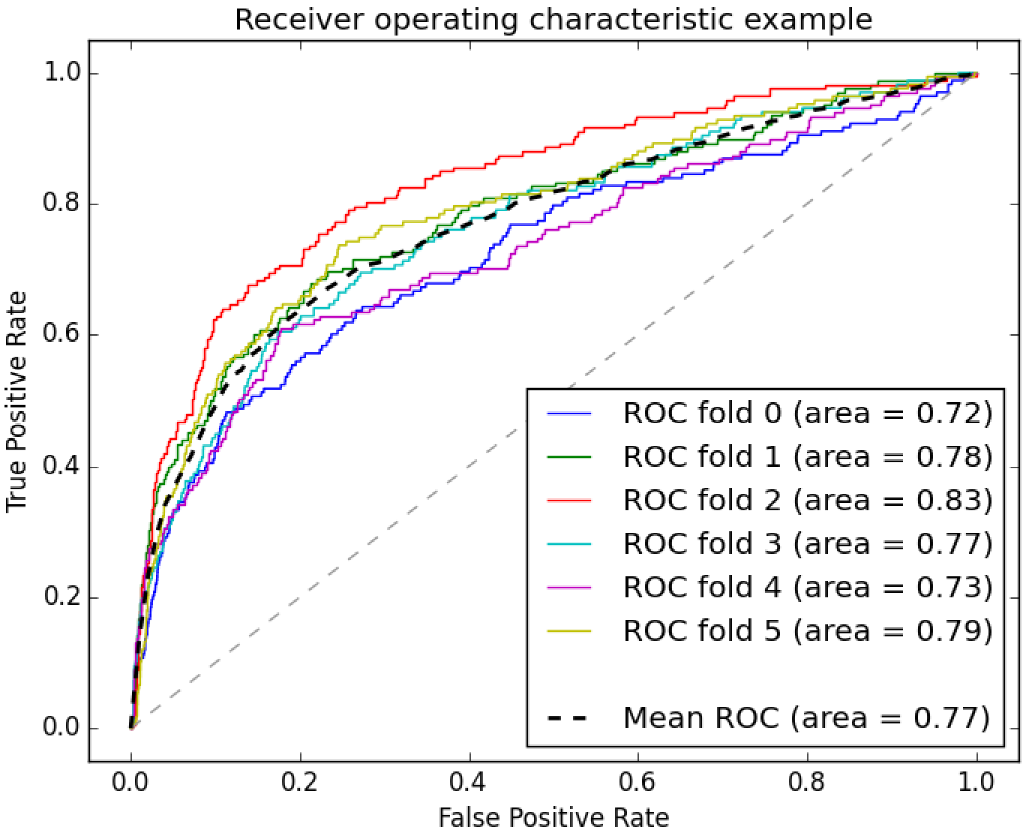


Figure 5.4 ROC Curve for Logistic Regression with L1 Regularization

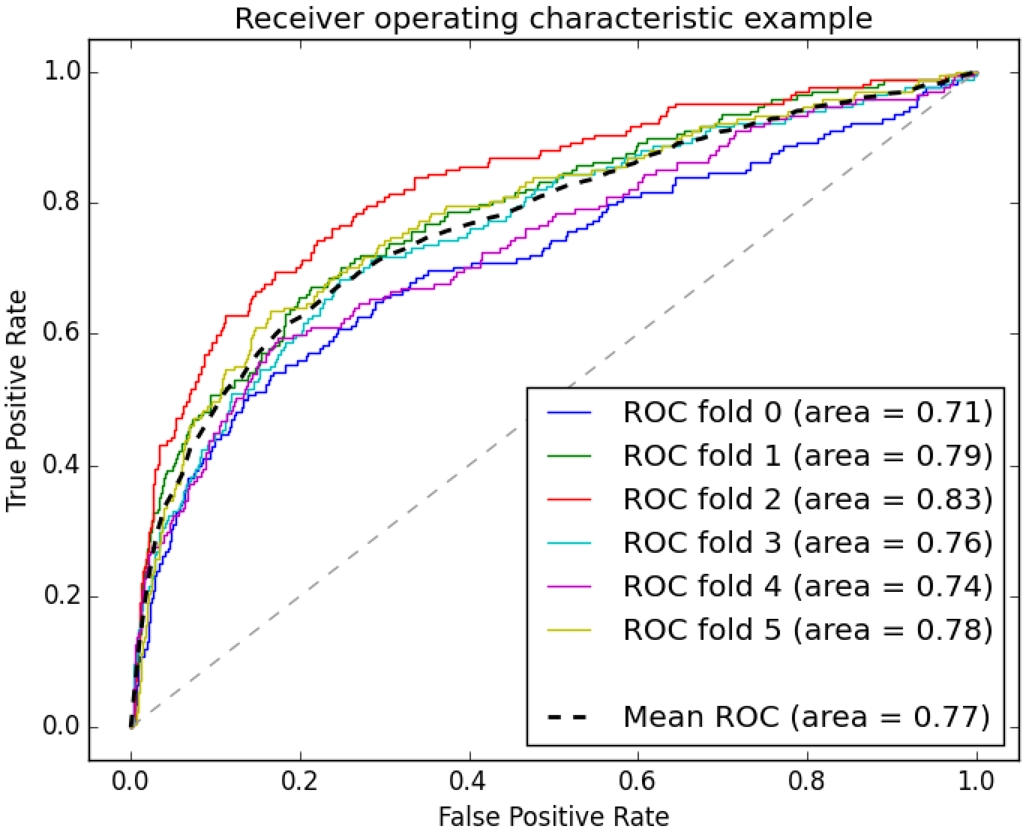


Figure 5.5 ROC Curve for SVM classifier and F-score based Feature Selection

CHAPTER 6. CONCLUSIONS

6.1 Summarization of Results

We present how a Big Data infrastructure can drive an end-to-end pipeline for predicting customer attrition in E-commerce world. The results we derive from this study and the contributions we make can be summarized below.

- We discuss how e-commerce organizations can design their data warehouse and big data infrastructure such that they can readily be used to drive data science and analytic applications with minimal effort in purposing this data.
- We discuss some of the popular tools that are available through the hadoop stack which can be used to transform raw data at huge scale, perform aggregations, filter and update continuously so that a data science pipeline can be built.
- We prove our novel proposal on how implicit features obtained through through clickstream/web logs and marketing campaign data mining, etc act as significant features in establishing customer behavior, experience and hence can be used as features to find customer churn.
- Through feature Selection methodologies, we establish how several product categories (CAT 10, CAT 11, CAT 12, etc), web channel experience (Futile Sessions,Dotcom shopping, Time spent, etc), cart activity (Carts Abandoned, etc) all play significant role in driving customer churn.
- Through cross validation techniques we establish that Gradient Boost Ensemble classifier, SVD Classifier, Logistic Regression with L1 regularization are the best models in predicting customer churn.

- Lastly but not least, the most important contribution from our work is an end to end generic Customer churn model consisting of both data and algorithmic pipeline that is applicable in a large E-commerce industry or similar industry using Big Data.

6.2 Future Work

There is a continous scope for improving the Feature Engineering process and the model building process from where we have currently stand through this work. This activity is continuous and iterative in nature. An immediate addition to improve the current results are using Grid search functionality to do hyper-parameter tuning to Gradient Boosting Classifier which happens to be our best classifier. Another important avenue worth exploring for addressing customer churn is the application of Time series analysis to this problem. As the businesses grow and get more complex there are more additional data sources, channels that continuously open up and may hold valuable information. This needs to be captured and harnessed for such or similar applications.

BIBLIOGRAPHY

[AC^{+}10] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[AKR08] Dudyala Anil Kumar and V Ravi. Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, 1(1):4–28, 2008.

[Alp14] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[BLB^{+}13] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[Bra97] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[BVdP09] Jonathan Burez and Dirk Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[CFS12] Zhen-Yu Chen, Zhi-Ping Fan, and Minghe Sun. A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of operational research*, 223(2):461–472, 2012.

[CVdP09] Kristof Coussement and Dirk Van den Poel. Improving customer attrition prediction by integrating emotions from client/company interaction emails and evaluating multiple classifiers. *Expert Systems with Applications*, 36(3):6127–6134, 2009.

[DG06] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[DL97] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[EIO14] Nadir Omer Fadl Elssied, Othman Ibrahim, and Ahmed Hamza Osman. A novel feature selection based on one-way anova f-test for e-mail spam classification. *Research Journal of Applied Sciences, Engineering and Technology*, 7(3):625–638, 2014.

[FP08] Jillian Dawes Farquhar and Tracy Panther. Acquiring and retaining customers in uk banks: An exploratory study. *Journal of Retailing and Consumer Services*, 15(1):9–21, 2008.

[Fri01] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[Gef00] David Gefen. E-commerce: the role of familiarity and trust. *Omega*, 28(6):725–737, 2000.

[GL99] William Gropp and Ewing Lusk. Reproducible measurements of mpi performance characteristics. In Jack Dongarra, Emilio Luque, and Toms Margalef, editors, *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, volume

1697 of *Lecture Notes in Computer Science*, pages 11–18. Springer Berlin Heidelberg, 1999.

[Hal99] Mark A Hall. *Correlation-based feature selection for machine learning.* PhD thesis, The University of Waikato, 1999.

[Haw04] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[HHR10] Muzammil Hanif, Sehrish Hafeez, and Adnan Riaz. Factors affecting customer satisfaction. *International Research Journal of Finance and Economics*, 60(1):44–52, 2010.

[HJL04] David W Hosmer Jr and Stanley Lemeshow. *Applied logistic regression.* John Wiley & Sons, 2004.

[HMS66] Earl B Hunt, Janet Marin, and Philip J Stone. Experiments in induction. 1966.

[Joa98] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features.* Springer, 1998.

[Jol02] Ian Jolliffe. *Principal component analysis.* Wiley Online Library, 2002.

[KR92] Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning*, pages 249–256, 1992.

[KZP07] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[Mah00] Balasubramaniam Mahadevan. Business models for internet-based e-commerce: An anatomy. *California management review*, 42(4):55–69, 2000.

[MBN02] Luis Carlos Molina, Lluís Belanche, and Àngela Nebot. Feature selection algorithms: a survey and experimental evaluation. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*, pages 306–313. IEEE, 2002.

[MCB$^+$11] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela H Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.

[MCeC13] Vera L Miguéis, Ana Camanho, and João Falcão e Cunha. Customer attrition in retailing: an application of multivariate adaptive regression splines. *Expert Systems with Applications*, 40(16):6225–6232, 2013.

[Ng04] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM, 2004.

[ORS$^+$08] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1099–1110. ACM, 2008.

[Pla99] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[Ris01] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM New York, 2001.

[SAP02] Srini S Srinivasan, Rolph Anderson, and Kishore Ponnavolu. Customer loyalty in e-commerce: an exploration of its antecedents and consequences. *Journal of retailing*, 78(1):41–50, 2002.

[SIL07] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[SKRC10] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE, 2010.

[SLLM02] Mark Sweiger, Jimmy Langston, Howard Lombard, and Mark R Madsen. *Clickstream data warehousing*. John Wiley & Sons, Inc., 2002.

[SR15] G Ganesh Sundarkumar and Vadlamani Ravi. A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, 37:368–377, 2015.

[SS01] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[TSA$^+$10] Ashish Thusoo, Zheng Shao, Suresh Anthony, Dhruba Borthakur, Namit Jain, Joydeep Sen Sarma, Raghotham Murthy, and Hao Liu. Data warehousing and analytics infrastructure at facebook. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1013–1020. ACM, 2010.

[TXH$^+$04] Weida Tong, Qian Xie, Huixiao Hong, Leming Shi, Hong Fang, and Roger Perkins. Assessment of prediction confidence and domain extrapolation of two structure-activity relationship models for predicting estrogen receptor binding activity. *Environmental health perspectives*, pages 1249–1254, 2004.

[WC02] Chih-Ping Wei and I-Tang Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert systems with applications*, 23(2):103–112, 2002.

[XLNY09] Yaya Xie, Xiu Li, EWT Ngai, and Weiyun Ying. Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36(3):5445–5449, 2009.

[YGGH11] Xiaobing Yu, Shunsheng Guo, Jun Guo, and Xiaorong Huang. An extended support vector machine forecasting framework for customer churn in e-commerce. *Expert Systems with Applications*, 38(3):1425–1430, 2011.

[YKG10] Sangho Yoon, Jim Koehler, and Adam Ghobarah. Prediction of advertiser churn for google adwords. 2010.

[YL04] Lei Yu and Huan Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[YM07] Seongwook Youn and Dennis McLeod. A comparative study for email classification. In *Advances and innovations in systems, computing sciences and software engineering*, pages 387–391. Springer, 2007.