Customer Personality Analysis

A report submitted in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

By

Nethala Nikhil

Regd. No.: 20B91A12D7

Under Supervision of Mr. Gundala Nagaraju

Henotic Technology Pvt Ltd, Hyderabad

(Duration: 7th July, 2022 to 6th September, 2022)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SAGI RAMA KRISHNAM RAJUENGINEERING COLLEGE

(An Autonomous Institution)

Approved by AICTE, NEW DELHI and Affiliated to JNTUK, Kakinada

CHINNA AMIRAM, BHIMAVARAM,

ANDHRA PRADESH

Table of Contents

Abstract

Delivering superior value to customers is an ongoing concern of management in many business markets of today. Knowing where value resides from the standpoint of the customer has become critical for suppliers.In this i provide a greatest applications of machine learning is to classify individuals based on their personality traits.Each person on this planet is unique and carries a unique personality. The availability of a high-dimensional and large amount of data has paved the way for increasing marketing campaigns' effectiveness by targeting specific people. Such personality-based communications are highly effective in increasing the popularity and attractiveness of products and services.It increased usage, customer satisfaction, and broader acceptance among users.

1.0 Introduction

Customer Personality Analysis is a detailed analysis of a company customers. It helps a business to better understand its customers and makes it easier for them to modify products according to the specific needs, behaviors, and concerns of different types of customers.

Customer personality analysis helps a business to modify its product based on its target customers from different types of customer segments. For example, instead of spending money to market a new product to every customer in the company's database, a company can analyze which customer segment is most likely to buy the product and then market the product only on that particular segment

1.1. What are the different types of Machine Learning?

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1.Supervised Machine Learning

2.Unsupervised Machine Learning

3.Semi-Supervised Machine Learning

4.Reinforcement Learning

1. Supervised Machine Learning

As its name suggests, Supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labelled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.Let's understand supervised learning with an example. Suppose we have an input dataset of cats and dog images. So, first, we will provide the training to the machine to understand the images, such as the **shape & size of the tail of cat and dog, Shape of eyes, colour, height (dogs are taller, cats are smaller), etc.** After completion of training, we input the picture of a cat and ask the machine to identify the object and predict the output. Now, the machine is well trained, so it will check all the features of the object, such as height,

shape, colour, eyes, ears, tail, etc., and find that it's a cat. So, it will put it in the Catcategory. This is the process of how the machine identifies the objects in Supervised Learning. **The main goal of the supervised learning technique is to map the input variable(x)with the output variable(y).** Some real-world applications of supervised learning are **RiskAssessment, Fraud Detection, Spam filtering,** etc.

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given

below:

- **Classification**
- **Regression**

a) Classification

Classification algorithms are used to solve the classification problems in which the outputvariable is categorical, such as "Yes" or No, Male or Female, Red or Blue, etc. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc.

b) Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

2. Unsupervised Machine Learning

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.In unsupervised learning, the models are trained with the data that is neither classified nor labelled, and the model acts on that data without any supervision.

Categories of Unsupervised Machine Learning

Unsupervised Learning can be further classified into two types, which are given below:

- **Clustering**
- **Association**

1) Clustering

The clustering technique is used when we want to find the inherent groups from the data. It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. An example of the clustering algorithm is grouping the customers by their purchasing behaviour.

2) Association

Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit. This algorithm is mainly applied in Market Basket analysis, Web usage mining, continuous production, etc.

3. Semi-Supervised Learning

Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate groundbetween Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabeled datasets during the training period.Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for corporate purposes, they may have few labels. It is completely different from supervised and unsupervised learning as they are based on the presence & absence of labels.To overcome the drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced

4. Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance. Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

In reinforcement learning, there is no labelled data like supervised learning, and agentslearn from their experiences only.The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards.Due to its way of working, reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, multi-agent systems.A reinforcement learning problem can be formalized using Markov Decision Process(MDP). In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state

1.2. Benefits of Using Machine Learning in Customer personality analysis

Customer Personality Analysis is a detailed analysis of a company's ideal customers. It helps a business to better understand its customers and makes it easier for them to modify products according to the specific needs, behaviours, and concerns of different types of customers. In this article, I'm going to introduce you to a data science project on customer personality analysis with Python.

Data Science and Artificial Intelligence are revolutionizing the world through technical transformations. We can observe many machine learning applications in day-to-day lives, but one of the greatest applications of machine learning is to classify individuals based on their personality traits. Each person on this planet is unique and carries a unique personality. The availability of a high-dimensional and large amount of data has paved the way for increasing marketing campaigns' effectiveness by targeting specific people. Such personality-based communications are highly effective in increasing the popularity and attractiveness of products and services. It

increased usage, customer satisfaction, and broader acceptance among users. Some common examples are Personalization of online advertisement campaigns leads to more revenue and click-through rates.

Personality traits are closely associated with an individual's behaviour and preferences. Hence the fusion of a personality-based approach has primarily increased the Recommender System's attractiveness.

Personality-based adaptions are also used to provide personalized visualizations and could even suggest better music recommendations.

Data Science and Artificial Intelligence are revolutionizing the world through technical transformations. We can observe many machine learning applications in day-to-day lives, but one of the greatest applications of machine learning is to classify individuals based on their personality traits. Each person on this planet is unique and carries a unique personality. The availability of a high-dimensional and large amount of data has paved the way for increasing marketing campaigns' effectiveness by targeting specific people. Such personality-based communications are highly effective in increasing the popularity and attractiveness of products and services. It increased usage, customer satisfaction, and broader acceptance among users. Some common examples are Personalization of online advertisement campaigns leads to more revenue and click-through rates.

1.3. About Customer analysis

The purpose of customer segmentation is to divide customers into many different ways. Customers can be grouped by their demographic, behavior, lifestyle, psychographic, value, etc. Segmentation is mostly used for marketing, but there are other reasons to segment your customer base. Using customer segmentation in marketing means that you can target the right people with the right messaging about your products. This will increase the success of your marketing campaigns

1.3.1 AI / ML Role in Customer Personality Analysis

Several researchers are working on embedding personality and enhancing social interactions between AI systems and their environments (Rodić et al., 2015). Voice personality could influence the 6 acceptance and continued use of social AI systems, especially for elderly people (Rodić et al., 2016; Shareef et al., 2021). Furthermore, people generally prefer female extraverted voices in social AI systems, and it is important to consider individual preferences during design (Loideain & Adams, 2020). Such studies have created the need for research that can help assess, understand, and apply individual differences in adaptation to AI technologies that manifest social agency capabilities (Chang et al., 2018; Matthews et al., 2020). On the other hand, peoples' personality traits have been used to train AI algorithms that help explain human behaviors like gambling (Cerasa et al., 2018), cyberbullying (Sánchez-Medina et al., 2020), and desirability (Fatahi & Moradi, 2016). They have also been used to train AI algorithms for candidate recruitment (Lee & Ahn, 2020) and predict peoples' reactions to tweets (Gallo et al., 2020) based on personality analyses. Furthermore, some studies suggest that the way people interact with AI is different from the way they do with other humans. Although people tend to be more open, agreeable, extraverted, conscientious, and self-disclosing with humans than with AI (Mou & Xu, 2017), extraverts are more likely to delegate decision making to AI than introverts, and conscientious people tend to prioritize performance over convenience (Goldbach et al., 2019). Personality traits influence consumers' preferences and online shopping behaviors and experiences (Anaza, 2014; Bosnjak et al., 2007; Marbach et al., 2016; Wu & Ke, 2015). For

example, the Big Five traits (neuroticism, conscientiousness, extraversion, openness, and agreeableness) influence impulsive and compulsive online shopping behaviors (Gohary & Hanzaee, 2014; Olsen et al., 2016; Turkyilmaz et al., 2015). They also influence self-reported happiness and non-grocery shopping (Goldsmith, 2016). Furthermore, extraversion and conscientiousness have been shown to influence consumers' willingness to pay (Ufer et al., 2019). Meanwhile, aggressiveness and altruism have significant impacts on consumers' complaining attitudes and behaviors (Souiden et al., 2019). This review reveals the conspicuous absence of personality studies in the context of voice shopping despite the importance of personalization and perceptions of ease of use for the customer experience of voice shoppers. Therefore, this study seeks to fill this research gap by investigating the relationship between personality traits and perceptions of customer experience performance in the context of voice shopping.

2.0 Literature review

Consumers increasingly expect to use digital platforms to obtain instant, frictionless, and memorable experiences during online shopping (Behrenbeck et al., 2015; Williams et al., 2020). Consequently, firms are constantly developing strategies to satisfy their customers' experiential needs through the latest technologies adopted by consumers (Fanderl et al., 2019; Lim et al., 2020). One strategy that many firms are currently using to stand out from the competition is by providing voice shopping services (Arnett et al., 2018; Fiona, 2017; Kinsella & Mutchier, 2019). The term voice shopping today mostly describes the use of artificial intelligence (AI)-based voice assistants like Amazon's Alexa and Google's Google Assistant to shop online. In the US alone, one in five consumers has performed voice shopping through this shopping channel, which is already worth over 1.8 billion USD (Kinsella & Mutchler, 2018a). This has led to calls for studies that explain how to improve individual customer experiences when using voice assistants through personalization (Davenport et al., 2020; de Barcelos Silva et al., 2020; Duan et al., 2019; Dwivedi et al., 2020). Prior research has established the importance of personalization in customer experience, especially when using AI-enabled technologies (Ameen et al.,2021; Tyrväinen et al., 2020; von Briel, 2018). It is also known that consumer personality is a key determinant of personalization in e-commerce (Kazeminia et al., 2019; Kim et al., 2015; Moon, 2002). Yet, no study investigates how/if personality affects customer experience during voice shopping in particular

2.1. Customer experience performance

Customer experience is holistically conceptualized as a multidimensional construct that characterizes a customer's cognitive, emotional, behavioral, sensorial, and social responses to service delivery processes (Hsu & Tsou, 2011; Lemon & Verhoef, 2016; Shi et al., 2020; Verhoef et al., 2009). It encompasses the total experience of the customer throughout the customer journey (Laming & Mason, 2014; Verhoef et al., 2009). Although measuring customer experience is critical for decision making, scholars and practitioners started measuring the overall customer experience only recently. Consequently, there is, to date, no well-established customer experience measurement scale or approach (Lemon & Verhoef, 2016; Morgeson et al., 2015). Customer satisfaction and Net Promoter Score (NPS) are currently the most popular approaches used to measure customer experience (Klie, 2013; Level 3 Communications, 2010; Santander UK, 2014). However, customer satisfaction only captures the customer's emotional state resulting from the customer's interaction with a platform or business (Verhoef, 2003). Thus, customer feedback metrics that focus on a specific dimension of customer experience are not strong predictors of customer experience performance, thus calling for the development of stronger measurement scales (Lemke et al., 2011; Lemon & Verhoef, 2016). 4 While some researchers have attempted to conceptualize customer experience and to evaluate its impact on shopping intentions (Hsu & Tsou, 2011; Shi et al., 2020), others have investigated tools that can help firms comprehensively measure their overall customer experience performance (Kuppelwieser & Klaus, 2020; Scheidt & Chung, 2019; Sperkova, 2019). Some are also investigating the antecedents of customer experience (Foroudi et al., 2018; Hsu & Tsou, 2011; McLean & Wilson, 2016) and how to best manage the customer experience in this era of big data (Grewal et al., 2009; Holmlund et al., 2020; Witell et al., 2020). Others have explored the mediating role of customer experience in relation to utilitarian/hedonic attributes of a product and brand equity, social interaction, convenience, and customer satisfaction (Sheng & Teo, 2012; Srivastava & Kaul, 2014). In online contexts, customer experience is centered around information technology (IT) access and design, customer support, customer service, and fulfillment in relation to product quality, price, description, and delivery time (Stanworth et al., 2015). Online customer experience is influenced by a web page's verbal and visual design elements (Bleier et al., 2019). Depending on the product type and brand trustworthiness, this experience could influence purchase decisions. Perceived utilitarian and hedonic benefits have been found to influence customer satisfaction with online social network services (Hsu et al., 2014). While web design quality enjoyment and web service quality influence customer satisfaction, these relationships are moderated by websites' interactivity (Ku & Chen, 2015). In the context of mobile commerce, customer experience is an important factor for the improvement of customer conversion and repurchase intention (Chopdar & Balakrishnan, 2020; Kaatz et al., 2019; Wagner et al., 2020). Perceived enjoyment and ubiquity directly affect customer satisfaction, and perceived enjoyment is influenced by two-way communication, responsiveness, and synchronicity of the mobile commerce platform

(Chopdar & Balakrishnan, 2020; Yang & Lee, 2017). Utilitarian factors of technology, ease of use, convenience, and customization influence enjoyment, while the perceived amount of time spent on a shopping activity using mobile applications influences the customer's shopping experience (McLean, Al-Nabhani, & Wilson, 2018). Also, perceived visual complexities negatively affect satisfaction with customer experience, and this relationship is mediated by perceived psychological cost (time, effort, and visual crowdedness) (Sohn et al., 2017). Furthermore, customers with good customer experience in terms of interactional justice tend to complain less than others when they face an issue with a vendor (Wu, 2013).

2.2. Internship Project - Data Link

Project dataset link is https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis

3.0 AI / ML Modelling and Results

3.1. Your Problem of Statement

Predictive models are most effective when they are constructed using a company's own historical claims data since this allows the model to recognize the specific nature of a company's exposure as well as its claims practices. The construction of the model also involves input from the company throughout the process, as well as consideration of industry leading claims practices and benchmarks.

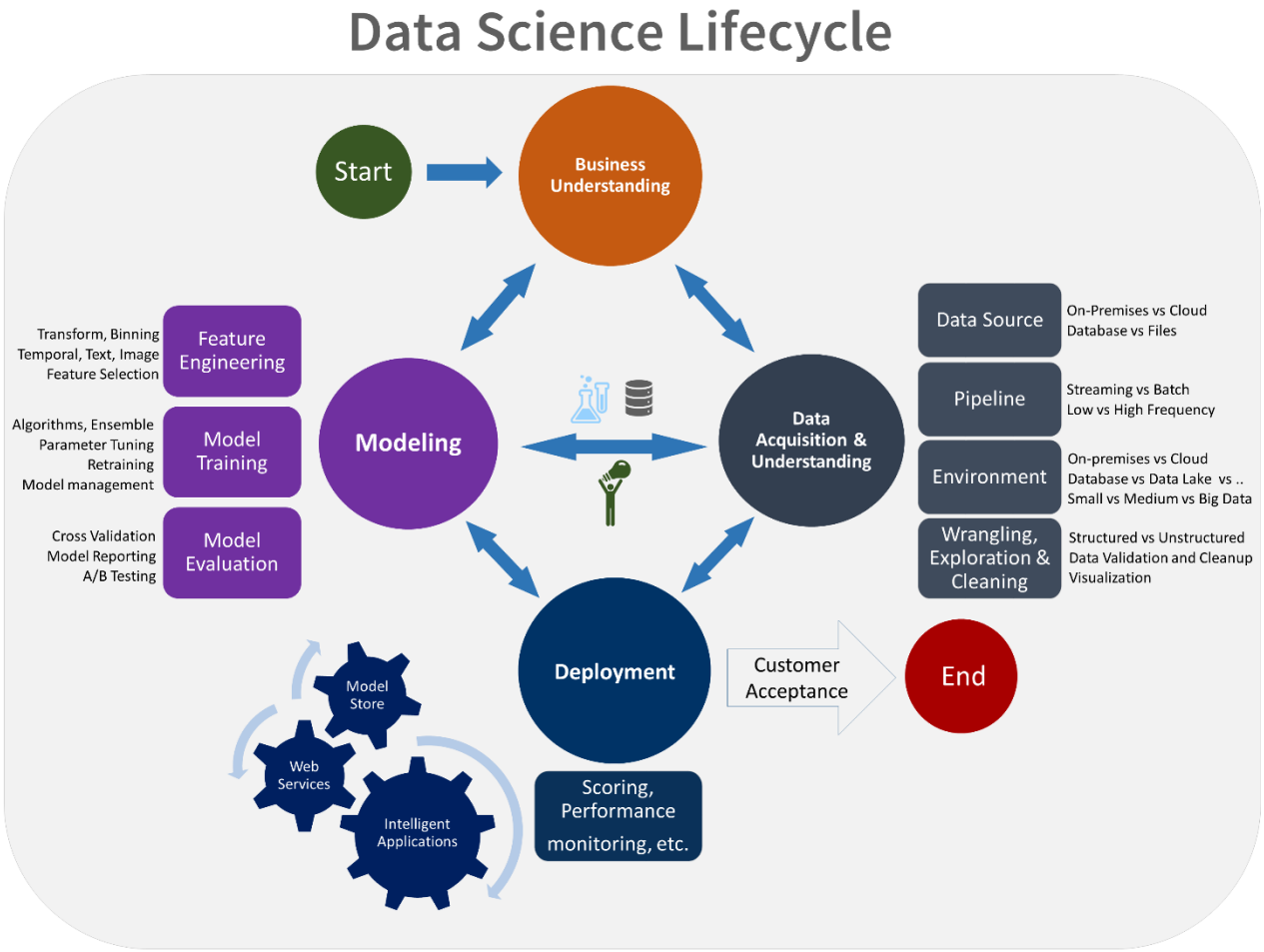
Predictive modelling can be used to quantify the impact to the claims department resulting from the failure to meet or exceed claim service leading practices. It can also be used to identify the root cause of claim leakage. Proper use of predictive modelling will allow for potential savings across two dimensions:

Early identification of claims with the potential for high leakage, thereby allowing for the proactive management of the claim

Recognition of practices that are unnecessarily increasing claims settlement payments

3.2. Data Science Project Life Cycle

Data Science is a multidisciplinary field of study that combines programming skills, domain expertise and knowledge of statistics and mathematics to extract useful insights and knowledge from data



3.2.1 Data Exploratory Analysis

Exploratory data analysis has been done on the data to look for relationship and correlation between different variables and to understand how they impact or target variable.

The exploratory analysis is done for Auto Quote / Policy Conversionwith different parameters and all the charts are presented in **Appendices 6.2 - List of charts (6.2.1 to 6.2.9)**

3.2.2 Data Pre-processing

We removed variables which does not affect our target variable(Claimed_Target)as they mayadd noise and also increase our computation time,we checked the data for anomalous data points and outliers.We did principal component analysis on the data set to filter out unnecessary variables and to select only the important variables which have greater correlation with our target variable.

3.2.2.1. Check the Duplicate and low variation data

Duplicate observations occur when two or more rows have the same values or nearly the same values.

A duplicate value is one in which all values in at least one row are identical to all of the values in another row. A comparison of duplicate values depends on the what appears in the cell—not the underlying value stored in the cell.

To find duplicates on a specific column, we can simply call duplicated() method on the column. The result is a boolean Series with the value True denoting duplicate. In other words, the value True means the entry is identical to a previous one.

3.2.2.2. Identify and address the missing variables

The cause of the presence of missing values in the dataset can be **loss of information, disagreement in uploading the data**, and many more. Missing values need to be imputed to proceed to the next step of the model development pipeline

The missing data model was similar to other features in the data set, but beyond that, the missing data values are not random. If the data is missing in the variable considered then it is said to be missing not at random(MNAR).

We handle missing data in a dataset by

Imputing the Missing Value

Replacing With Arbitrary Value. ...

Replacing With Mode. ...

Replacing With Median. ...

Replacing with previous value – Forward fill. ...

Replacing with next value – Backward fill. ...

Interpolation. ...

Impute the Most Frequent Value

3.2.2.3. Handling of Outliers

Outliers are **extreme values that deviate from other observations on data** , they may indicate a variability in a measurement, experimental errors or a novelty. In other words, an outlier is an observation that diverges from an overall pattern on a sample.

An *outlier* is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. Before abnormal observations can be singled out, it is necessary to characterize normal observations

deal with outliers or missing values in a dataset

Filling in zero : The easiest way to treat null values is to **fill the missing values as zero or replace the outliers with a zero**. It would not be the best method. Filling in with a number : One can fill all the null values with a single number by using . fillna() function.

3.2.2.4. Categorical data and Encoding Techniques

Categorical Data is **the data that generally takes a limited number of possible values**. Also, the data in the category need not be numerical, it can be textual in nature. All machine learning models are some kind of mathematical model that need numbers to work with.

Categorical data is simply **information aggregated into groups rather than being in numeric formats**, such as Gender, Sex or Education Level. They are present in almost all real-life datasets, yet the current algorithms still struggle to deal with them. Take, for instance, XGBoost or most SKlearn models

3.2.2.5. Feature Scaling

if the data in any conditions has data points far from each other, scaling is a technique to make them closer to each other or in simpler words, we can say that the scaling is used for making data points generalized so that the distance between them will be lower.

We used minmax scaling technique here

Min-Max scaling is a normalization technique that enables us to scale data in a dataset to a specific range using each feature's minimum and maximum value.

A Min-Max scaling is typically done via the following equation: $Xsc=X-XminXmax-Xmin$. One family of algorithms that is scale-invariant encompasses tree-based learning algorithms

3.2.3 Selection of Dependent and Independent variables

The dependent or target variable here is Claimed Target which tells us a particular policy holder has filed a claim or not the target variable is selected based on our business problem and what we are trying to predict.

The independent variables are selected after doing exploratory data analysis and we used Boruta to select which variables are most affecting our target variable.

3.2.4 Data Sampling Methods

The data we have is highly unbalanced data so we used some sampling methods which are used to balance the target variable so we our model will be developed with good accuracy and precision. We used three Sampling methods

3.2.4.1. Stratified sampling

Stratified sampling randomly selects data points from majority class so they will be equal to the data points in the minority class. So, after the sampling both the class will have same no of observations.

It can be performed using strata function from the library sampling.

3.2.4.2. Simple random sampling

Simple random sampling is a sampling technique where a set percentage of the data is selected randomly. It is generally done to reduce bias in the dataset which can occur if data is selected manually without randomizing the dataset.

We used this method to split the dataset into train dataset which contains 70% of the total data and test dataset with the remaining 30% of the data.

3.2.5 Models Used for Development

We built our predictive models by using the following ten algorithms

3.2.5.1. Model 01-LOGISTIC REGRESSION

Logistic uses logit link function to convert the likelihood values to probabilities so we can get a good estimate on the probability of a particular observation to be positive class or negative class. The also gives us p-value of the variables which tells us about significance of each independent variable.

3.2.5.2. Model 02-DECISION TREE

• Decision Tree is a supervised learning Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome

• In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. • The decisions or the test are performed on the basis of features of the given dataset.

3.2.5.3. Model 03-RANDOM FOREST

Random forest is an algorithm that consists of many decision trees. It was first developed by Leo Breiman and Adele Cutler. The idea behind it is to build several trees, to have the instance classified by each tree, and to give a "vote" at each class. The model uses a "bagging" approach and the random selection of features to build a collection of decision trees with controlled variance. The instance's class is to the class with the highest number of

votes, the class that occurs the most within the leaf in which the instance is placed. The error of the forest depends on: Trees correlation: the higher the correlation, the higher the forest error rate.

The strength of each tree in the forest. A strong tree is a tree with low error. By using trees that classify the instances with low error the error rate of the forest decreases.

3.2.5.4. Model 04-EXTRA TREES CLASSIFER

Extremely Randomized Trees Classifier(Extra Trees Classifier) is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, Each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees

3.2.5.5. Model 05-KNN CLASSIFIER

The k-nearest neighbours algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. KNN works by finding the distances between a query and all the examples in the data, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression). KNN is a supervised learning algorithm used for classification. KNN is a classification algorithm which falls under the greedy techniques however k-means is a clustering algorithm (unsupervised machine learning technique).

3.2.6 Model 06-GAUSSIAN NAIVE BAYES

Gaussian Naive Bayes is a probabilistic classification algorithm based on applying Bayes' theorem with strong independence assumptions. Gaussian Naive Bayes supports continuous valued features and models each as conforming to a Gaussian (normal) distribution. An approach to create a simple model is to assume that the data is described by a Gaussian distribution with no co-variance (independent dimensions) between dimensions.

3.2.7 Model 07-XGB CLASSIFER

XGBoost, which stands for Extreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems. It uses more accurate approximations to find the best tree model. Boosting: N new training data sets are formed by random sampling with replacement from the original dataset, during which some observations may be repeated in each new training data set.

3.2.8 Model 08-SVM

A support vector machine (SVM) is a type of deep learning algorithm that performs supervised learning for classification or regression of data groups. In AI and machine learning, supervised learning systems provide both input and desired output data, which are labeled for classification. Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection. The advantages of support vector machines are: Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.

3.2.9 Model 09-LIGHT GBM

Light GBM, short for Light Gradient Boosting Machine, is a free and open source distributed gradient boosting framework for machine learning originally developed by Microsoft. It is based on decision tree algorithms and used for ranking, classification and other machine learning tasks. Light GBM is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks

3.3. AI / ML ModelsAnalysis and Final Results

We used our train dataset to build the above models and used our test data to check the accuracy and performance of our models.

We used confusion matrix to check accuracy, Precision, Recall and F1 score of our models and compare and select the best model for given auto dataset of size ~ 272252 policies.

3.3.1 Different Model codes

The Python code for models with simple random sampling technique as follows:

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import GradientBoostingClassifier
import lightgbm as lgb
# Create objects of classification algorithm with default hyper-parameters
ModelLR = LogisticRegression()
ModelDC = DecisionTreeClassifier()
ModelRF = RandomForestClassifier()
ModelET = ExtraTreesClassifier()
ModelKNN = KNeighborsClassifier(n_neighbors=5)
ModelSVM = SVC(probability=True)
modelBAG = BaggingClassifier(base_estimator=None, n_estimators=100, max_samples=1.0, max_features=1.0,
                    bootstrap=True, bootstrap_features=False, oob_score=False, warm_start=False,
                    n_jobs=None, random_state=None, verbose=0)
ModelGB = GradientBoostingClassifier(loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0,
                        criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1,
```

```
                    min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0,
                     init=None, random_state=None,
                    max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False,
                    validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)
ModelLGB = lgb.LGBMClassifier()
ModelGNB = GaussianNB()
# Evalution matrix for all the algorithms
MM = [ModelLR, ModelDC, ModelRF, ModelET, ModelKNN, ModelSVM, modelBAG, ModelGB, ModelLGB,
ModelGNB]
for models in MM:
    # Fit the model
    models.fit(x_train, y_train)
    # Prediction
    y_pred = models.predict(x_test)
    y_pred_prob = models.predict_proba(x_test)
    # Print the model name
    print('Model Name: ', models)
    # confusion matrix in sklearn
    from sklearn.metrics import confusion_matrix
    from sklearn.metrics import classification_report
    # actual values
    actual = y_test
    # predicted values
    predicted = y_pred
    # confusion matrix
    matrix = confusion_matrix(actual,predicted, labels=[1,0],sample_weight=None, normalize=None)
    print('Confusion matrix : \n', matrix)
    # outcome values order in sklearn
    tp, fn, fp, tn = confusion_matrix(actual,predicted,labels=[1,0]).reshape(-1)
    print('Outcome values : \n', tp, fn, fp, tn)
    # classification report for precision, recall f1-score and accuracy
    C_Report = classification_report(actual,predicted,labels=[1,0])
    print('Classification report : \n', C_Report)
    # calculating the metrics
    sensitivity = round(tp/(tp+fn), 3);
    specificity = round(tn/(tn+fp), 3);
```

```
accuracy = round((tp+tn)/(tp+fp+tn+fn), 3);
balanced_accuracy = round((sensitivity+specificity)/2, 3);
precision = round(tp/(tp+fp), 3);
f1Score = round((2*tp/(2*tp + fp + fn)), 3);
# Matthews Correlation Coefficient (MCC). Range of values of MCC lie between -1 to +1.
# A model with a score of +1 is a perfect model and -1 is a poor model
from math import sqrt
mx = (tp+fp) * (tp+fn) * (tn+fp) * (tn+fn)
MCC = round(((tp * tn) - (fp * fn)) / sqrt(mx), 3)
print('Accuracy :', round(accuracy*100, 2),'%')
print('Precision :', round(precision*100, 2),'%')
print('Recall :', round(sensitivity*100,2), '%')
print('F1 Score :', f1Score)
print('Specificity or True Negative Rate :', round(specificity*100,2), '%'  )
print('Balanced Accuracy :', round(balanced_accuracy*100, 2),'%')
print('MCC :', MCC)
# Area under ROC curve
from sklearn.metrics import roc_curve, roc_auc_score
print('roc_auc_score:', round(roc_auc_score(actual, predicted), 3))
# ROC Curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
logit_roc_auc = roc_auc_score(actual, predicted)
fpr, tpr, thresholds = roc_curve(actual, models.predict_proba(x_test)[:,1])
plt.figure()
# plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot(fpr, tpr, label= 'Classification Model' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
```

```
plt.show()
print('-----------------------------------------------------------------------------------------------------')
#-----------------------------------------------------------------------------------------------------
new_row = {'Model Name' : models,
        'True_Positive' : tp,
        'False_Negative' : fn,
        'False_Positive' : fp,
        'True_Negative' : tn,
        'Accuracy' : accuracy,
        'Precision' : precision,
        'Recall' : sensitivity,
        'F1 Score' : f1Score,
        'Specificity' : specificity,
        'MCC':MCC,
        'ROC_AUC_Score':roc_auc_score(actual, predicted),
        'Balanced Accuracy':balanced_accuracy}
BResults = BResults.append(new_row, ignore_index=True)
#-----------------------------------------------------------------------------------------------------
```

4.0 Conclusions and Future work

The model results in the following order by considering the model accuracy, F1 score and RoC AUC score.

1) **DecisionTreeClassifier(max_features='sqrt,r')**with Stratified and Random Sampling
2) **Logistic Regression** with Simple Random Sampling
3) **DecisionTreeClassifier** with Simple Random Sampling

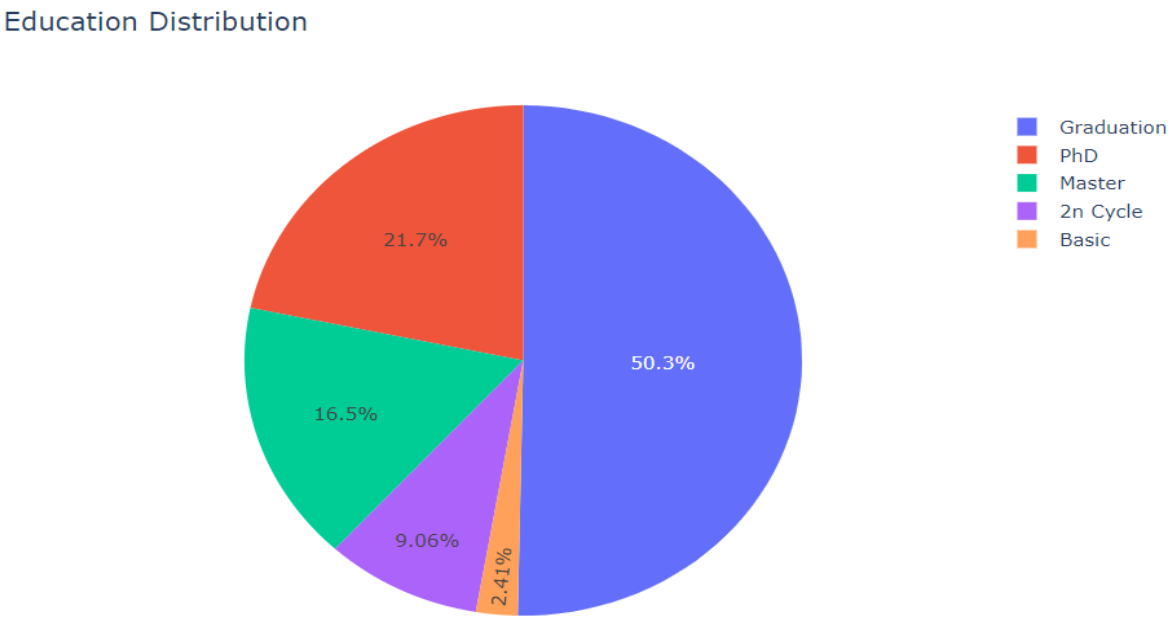
	Model Name	K Value	True_Positive	False_Negative	False_Positive	True_Negative	Accuracy	Precision	Recall	F1 Score	Sp
)	LogisticRegression()	NaN	17	79	13	556	0.862	0.567	0.177	0.270	
	DecisionTreeClassifier()	NaN	43	53	64	505	0.824	0.402	0.448	0.424	
:	(DecisionTreeClassifier(max_features='sqrt', r...	NaN	27	69	16	553	0.872	0.628	0.281	0.388	
;	(ExtraTreeClassifier(random_state=1638273757),...	NaN	30	66	20	549	0.871	0.600	0.312	0.411	
;	KNeighborsClassifier()	NaN	13	83	23	546	0.841	0.361	0.135	0.197	
;	SVC(probability=True)	NaN	0	96	0	569	0.856	NaN	0.000	0.000	
;	(DecisionTreeClassifier(random_state=207891646...	NaN	37	59	30	539	0.866	0.552	0.385	0.454	
'	([DecisionTreeRegressor(criterion='friedman_ms...	NaN	35	61	23	546	0.874	0.603	0.365	0.455	
;	LGBMClassifier()	NaN	36	60	28	541	0.868	0.562	0.375	0.450	

We recommend model – **DecisiontreeClassifier(max** with Stratified and Random Sampling technique as a best fit for the give n BI claims dataset. We considered Random Forest because it uses bootstrap aggregation which can reduce bias and variance in the data and can leads to good predictions with claims dataset.

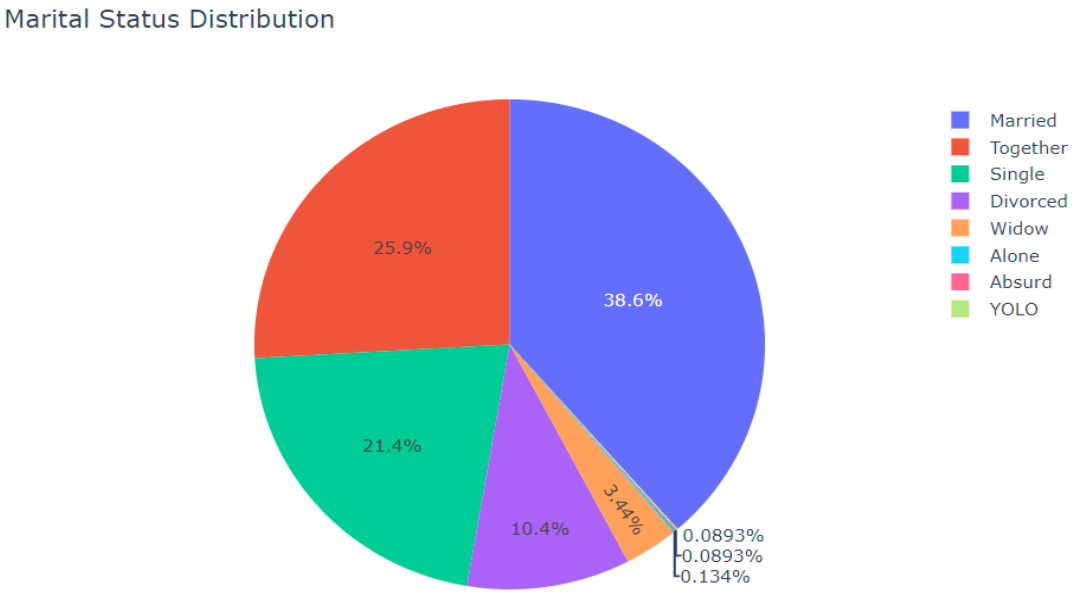
5.0 Appendices

5.1. List of Charts

5.1.1 Chart 01: Education Distribution



5.1.2 Chart 02: Martial status Distribution



5.1.3 Chart 03 : income distribution