A PROJECT ON HEALTH INSURANCE LEAD PREDICTION USING PYTHON

Project submitted and prepared by Bishal Biswas
under guidance of Prof. Sambita Chakraborty

Project done as a part of assignments for
Post Graduate Diploma Program
From
Bombay Stock Exchange (BSE)
In collaboration with
Maulana Abul Kalam Azad University of Technology
(MAKAUT)

Certification of Approval

This document is hereby approved as credible study of the science subject carried out and represented in a manner to satisfy to the warrants of its acceptance as a prerequisite to the degree for which it has been submitted.

Moreover, it is understood that by this approval the undersigned does not necessarily endorse or approve any statements made, the opinion expressed or conclusion drawn therein but approved only for the sole purpose for which it has been indeed submitted.

Signatures of the Examiners with date.

×_____

×_____

×_____

Dated:

Countersigned by:

×_____

Prof. Sambita Chakraborty

Acknowledgement

Our project and everything started during the ongoing reign of SARS – CoVID19, virtually crippling the society and world as a whole sending everything into a lockdown. Although at better stage but with a second wave looming on the pre-existing distress in-spite of new vaccines combating the situation. Overcoming all this chaos the course of Post Graduate Diploma in Data Science by BSE in collaboration with MAKAUT was made possible thanks to the diplomacy and steps taken by both institutes to combat the situation and make this course and project a possibility.

I want to take this opportunity of the project to thank the people at BSE and MAKAUT who provided us this opportunity to have an exposure to real life scenarios and the status of the present market. I also want to thank Prof. Sambita Chakraborty for guiding with every step from imparting knowledge about the subject to the intricacies of the Data Preparation and Analysis including Cleaning, Visualization, clearing doubts and issues faced in addition to solving problems encountered.

I am also grateful to my batchmates and peers where our collective knowledgebase and doubt clearing helped a lot in completing this project. Lastly, I want to thank my family for the mental support they provided me that played a big part in completing this project.

×_____

Bishal Biswas.
BSE GENERATED ID: PGDDSPJULY2020/1
MAKAUT ENROLMENT:20BIL001P12029005
MAKAUT APPLICATION ID: 91268
b.biswas_94587@ieee.org

Contents

* * *

Objective and Purpose

Data is truly considered a resource in today's world. Data is everywhere and part of our daily lives in more ways than most of us realize in our daily lives. The amount of digital data that exists—that we create—is growing exponentially. As per the World Economic Forum, by 2025 we will be generating about 463 exabytes of data globally per day. Hence, there is a need for professionals who understand the basics of data science, big data, and data analytics. These three terms are often heard frequently in the industry, and while their meanings share some similarities, they also mean different things.

Data science is the combination of statistics, mathematics, programming, problem-solving, capturing data in ingenious ways, the ability to look at things differently, and the activity of cleansing, preparing, and aligning data. This umbrella term includes various techniques that are used when extracting insights and information from data.

Now, Big data refers to significant volumes of data that cannot be processed effectively with the traditional applications that are currently used. The processing of big data begins with raw data that isn't aggregated and is most often impossible to store in the memory of a single computer. A buzzword that is used to describe immense volumes of data, both unstructured and structured, big data can inundate a business on a day-to-day basis. Big data is used to analyze insights, which can lead to better decisions and strategic business moves.

Gartner provides the following definition of big data: "Big data is high-volume, and high-velocity or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation."

Data analytics involves applying an algorithmic or mechanical process to derive insights and running through several data sets to look for meaningful correlations. It is used in several industries, which enables organizations and data analytics companies to make more informed decisions, as well as verify and disprove existing theories or models. The focus of data analytics lies in inference, which is the process of deriving conclusions that are solely based on what the researcher already knows.

Industries like IT, Retail, Manufacturing, Automobile, Financial Institute, E Commerce etc. are focusing in depth towards Big Data Concept because they have found out its importance, they know Data is Asset and its value will grow day by day and it can lead the Global business. Some benefits of it are:

- Data driven decision making with more accuracy.
- Customer active engagement.
- Operation optimization.
- Data driven Promotions.
- Preventing frauds & threats.

- Exploring new sources of revenue.
- Being ahead of your competitors.

Thus, the growth of big data analytics will also probably be good for data scientists, especially those who have strong backgrounds in big data. Based on the growth of the big data analytics market in the past few years, along with the rising number of job openings, it's likely that demand for these skills will continue to increase in the near future.

Introduction

Since the invention of computers, people have used the term data to refer to computer information, and this information was either transmitted or stored. But that is not the only data definition; there exist other types of data as well. So, what is the data? Data can be texts or numbers written on papers, or it can be bytes and bits inside the memory of electronic devices, or it could be facts that are stored inside a person's mind.

Now, if we talk about data mainly in the field of science, then the answer to "what is data" will be that data is different types of information that usually is formatted in a particular manner. All the software is divided into two major categories, and those are programs and data. Programs are the collection made of instructions that are used to manipulate data. So, now after thoroughly understanding what is data and data science, let us learn some fantastic facts.

Growth in the field of technology, specifically in smartphones has led to text, video, and audio is included under data plus the web and log activity records as well. Most of this data is unstructured.

The term Big Data is used in the data definition to describe the data that is in the petabyte range or higher. Big Data is also described as 5Vs: variety, volume, value, veracity, and velocity. Nowadays, web-based eCommerce has spread vastly, business models based on Big Data have evolved, and they treat data as an asset itself. And there are many benefits of Big Data as well, such as reduced costs, enhanced efficiency, enhanced sales, etc.

The meaning of data expands beyond the processing of data in computing applications. When it comes to what data science is, a body made of facts is called data science. Accordingly, finance, demographics, health, and marketing also have different meanings of data, which ultimately make up different answers for what is data.

When we talk about data, we usually think of some large datasets with huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos etc. Machines don't understand free text, image or video data as it is, they understand 1s and 0s. So, it probably won't be good enough if we put on a

slideshow of all our images and expect our machine learning model to get trained just by that, hence a need for processing the data is required before being fed to the system for any analysis or estimation making learning or prediction too farfetched an idea to see realization.

Clean data is crucial for insightful data analysis. Data cleansing, data cleaning or data scrubbing is the first step in the overall data preparation process. It is the process of analyzing, identifying and correcting messy, raw data. Data cleaning involves filling in missing values, identifying and fixing errors and determining if all the information is in the right rows and columns. When analyzing organizational data to make strategic decisions you must start with a thorough data cleansing process. Cleaning data is crucial to data analysis. Data cleaning lays the groundwork for efficient, accurate and effective data analysis. Without cleaning data beforehand, the analysis process won't be clear or as accurate because the information in the dataset will be unorganized and scattered. Good analysis rests on clean data–it's as simple as that.

Analysis is the process of breaking a complex topic or substance into smaller parts in order to gain a better understanding of it. The technique has been applied in the study of mathematics and logic since before Aristotle, though analysis as a formal concept is a relatively recent development. Implementing these ideas of analysis using statistical processes to determine the future or optimize situations using available data at the disposal or data obtained from a specific source is the very idea on which the Data Analysis and subsequently Big Data Analysis is based on.

Beyond the analysis, even if we stop at this point, we are just holding a set of numbers, some arranged others not but all equivalently shared between fields of significance, which if not properly visualized wouldn't make any sense to a person working beyond the realm or field from where the data is based. Hence, the need for visualization stands. Visualization is the process of putting together visual mental imagery of what you are wanting to manifest. Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Data Gathering

Data gathering or Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.

Data collection can be carried by from various ways like:

1. Surveys. Surveys are one way in which you can directly ask customers for information.
2. Online Tracking.
3. Transactional Data Tracking.

4. Online Marketing Analytics.
5. Social Media Monitoring.
6. Collecting Subscription and Registration Data.
7. In-Store Traffic Monitoring.

Primary Data Collection

The term "primary data" refers to data you collect yourself, rather than data you gather after another party initially recorded it. Primary data is information obtained directly from the source. You will be the first party to use this exact set of data.

When it comes to data businesses collect about their customers, primary data is also typically first-party data. First-party data is the information you gather directly from your audience. It could include data you gathered from online properties, data in your customer relationship management system or non-online data you collect from your customers through surveys and various other sources.

First-party data differs from second-party and third-party data. Second-party data is the first-party data of another company. You can purchase second-party data directly from the organization that collected it or buy it in a private marketplace. Third-party data is information a company has pulled together from numerous sources. You can buy and sell this kind of data on a data exchange, and it typically contains a large number of data points. Because first-party data comes directly from your audience, you can have high confidence in its accuracy, as well as its relevance to your business.

Second-party data has many of the same positive attributes as first-party data. It comes directly from the source, so you can be confident in its accuracy, but it also gives you insights you couldn't get with your first-party data. Third-party data offers much more scale than any other type of data, which is its primary benefit.

Different types of data can be useful in different scenarios. It can also be helpful to use different types of data together. First-party data will typically be the foundation of your dataset. If your first-party data is limited, though, you may want to supplement it with second-party or third-party data. Adding these other types of data can increase the scale of your audience or help you reach new audiences.

Quantitative vs. Qualitative Data

Quantitative data comes in the form of numbers, quantities and values. It describes things in concrete and easily measurable terms. Examples include the number of customers who bought a given product, the rating a customer gave a product out of five stars and the amount of time a visitor spent on your website.

Because quantitative data is numeric and measurable, it lends itself well to analytics. When you analyze quantitative data, you may uncover insights that can

help you better understand your audience. Because this kind of data deals with numbers, it is very objective and has a reputation for reliability.

Qualitative data is descriptive, rather than numeric. It is less concrete and less easily measurable than quantitative data. This data may contain descriptive phrases and opinions. Examples include an online review a customer writes about a product, an answer to an open-ended survey question about what type of videos a customer likes to watch online and the conversation a customer had with a customer service representative.

Qualitative data helps explains the "why" behind the information quantitative data reveals. For this reason, it is useful for supplementing quantitative data, which will form the foundation of your data strategy. Because quantitative data is so foundational, this article will focus on collection methods for quantitative primary data.

Data Preparation

Data preparation is the process of cleaning and transforming raw data prior to processing and analysis. It is an important step prior to processing and often involves reformatting data, making corrections to data and the combining of data sets to enrich data. Good data preparation allows for efficient analysis, limits errors and inaccuracies that can occur to data during processing, and makes all processed data more accessible to users. It's also gotten easier with new tools that enable any user to cleanse and qualify data on their own.

Data preparation is often a lengthy undertaking for data professionals or business users, but it is essential as a prerequisite to put data in context in order to turn it into insights and eliminate bias resulting from poor data quality.

Benefits of Data Preparation

76% of data scientists say that data preparation is the worst part of their job, but the efficient, accurate business decisions can only be made with clean data. Data preparation helps:

- Fix errors quickly — Data preparation helps catch errors before processing. After data has been removed from its original source, these errors become more difficult to understand and correct.
- Produce top-quality data — Cleaning and reformatting datasets ensures that all data used in analysis will be high quality.
- Make better business decisions — Higher quality data that can be processed and analyzed more quickly and efficiently leads to more timely, efficient and high-quality business decisions.

Integrating the Cloud technology to the data preparation

As data and data processes move to the cloud, data preparation moves with it for even greater benefits, such as:

- Superior scalability — Cloud data preparation can grow at the pace of the business. Enterprise don't have to worry about the underlying infrastructure or try to anticipate their evolutions.
- Future proof — Cloud data preparation upgrades automatically so that new capabilities or problem fixes can be turned on as soon as they are released. This allows organizations to stay ahead of the innovation curve without delays and added costs.
- Accelerated data usage and collaboration — Doing data prep in the cloud means it is always on, doesn't require any technical installation, and lets teams collaborate on the work for faster results.

Considering beyond the above points, a good, cloud-native data preparation tool will offer other benefits (like an intuitive and simple to use GUI) for easier and more efficient preparation.

Data Preparation Steps

The specifics of the data preparation process vary by industry, organization and need, but the framework remains largely the same.

1. Gather data: The data preparation process begins with finding the right data. This can come from an existing data catalog or can be added ad-hoc.
2. Discover and assess data: After collecting the data, it is important to discover each dataset. This step is about getting to know the data and understanding what has to be done before the data becomes useful in a particular context.
3. Cleanse and validate data: Cleaning up the data is traditionally the most time-consuming part of the data preparation process, but it's crucial for removing faulty data and filling in gaps. Important tasks here include:
    a. Removing extraneous data and outliers.
    b. Filling in missing values.
    c. Conforming data to a standardized pattern.
    d. Masking private or sensitive data entries.

    Once data has been cleansed, it must be validated by testing for errors in the data preparation process up to this point. Often times, an error in the system will become apparent during this step and will need to be resolved before moving forward.
4. Transform and enrich data: Transforming data is the process of updating the format or value entries in order to reach a well-defined outcome, or to make the data more easily understood by a wider audience. Enriching data refers to adding and connecting data with other related information to provide deeper insights.

5. Store data: Once prepared, the data can be stored or channeled into a third-party application—such as a business intelligence tool—clearing the way for processing and analysis to take place.

Data Cleaning

Data cleaning is the process of preparing data for analysis by removing or modifying data that is incorrect, incomplete, irrelevant, duplicated, or improperly formatted. This data is usually not necessary or helpful when it comes to analyzing data because it may hinder the process or provide inaccurate results. There are several methods for cleaning data depending on how it is stored along with the answers being sought.

Data cleaning is not simply about erasing information to make space for new data, but rather finding a way to maximize a data set's accuracy without necessarily deleting information. For one, data cleaning includes more actions than removing data, such as fixing spelling and syntax errors, standardizing data sets, and correcting mistakes such as empty fields, missing codes, and identifying duplicate data points. Data cleaning is considered a foundational element of the data science basics, as it plays an important role in the analytical process and uncovering reliable answers. Most importantly, the goal of data cleaning is to create data sets that are standardized and uniform to allow business intelligence and data analytics tools to easily access and find the right data for each query.

Regardless of the type of analysis or data visualizations you need, data cleaning is a vital step to ensure that the answers you generate are accurate. When collecting data from several streams and with manual input from users, information can carry mistakes, be incorrectly inputted, or have gaps. Data cleaning helps ensure that information always matches the correct fields while making it easier for business intelligence tools to interact with data sets to find information more efficiently. One of the most common data cleaning examples is its application in data warehouses.

A successful data warehouse stores a variety of data from disparate sources and optimizes it for analysis before any modeling is done. To do so, warehouse applications must parse through millions of incoming data points to make sure they're accurate before they can be slotted into the right database, table, or other structure. Organizations that collect data directly from consumers filling in surveys, questionnaires, and forms also use data cleaning extensively. In their cases, this includes checking that data was entered in the correct field, that it doesn't feature invalid characters, and that there are no gaps in the information provided

Data Analysis

Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

A simple example of Data analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analyzing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called Data Analysis.

Types of Data Analysis: Techniques and Methods

There are several types of Data Analysis techniques that exist based on business and technology. However, the major Data Analysis methods are:

- Text Analysis
- Statistical Analysis
- Diagnostic Analysis
- Predictive Analysis
- Prescriptive Analysis

Text Analysis

Text Analysis is also referred to as Data Mining. It is one of the methods of data analysis to discover a pattern in large data sets using databases or data mining tools. It used to transform raw data into business information. Business Intelligence tools are present in the market which is used to take strategic business decisions. Overall, it offers a way to extract and examine data and deriving patterns and finally interpretation of the data.

Statistical Analysis

Statistical Analysis shows "What happen?" by using past data in the form of dashboards. Statistical Analysis includes collection, Analysis, interpretation, presentation, and modeling of data. It analyses a set of data or a sample of data. There are two categories of this type of Analysis - Descriptive Analysis and Inferential Analysis.

- Descriptive Analysis: analyses complete data or a sample of summarized numerical data. It shows mean and deviation for continuous data whereas percentage and frequency for categorical data.
- Inferential Analysis: analyses sample from complete data. In this type of Analysis, you can find different conclusions from the same data by selecting different samples.

Diagnostic Analysis

Diagnostic Analysis shows "Why did it happen?" by finding the cause from the insight found in Statistical Analysis. This Analysis is useful to identify behavior patterns of data. If a new problem arrives in your business process, then you can look into this Analysis to find similar patterns of that problem. And it may have chances to use similar prescriptions for the new problems.

Predictive Analysis

Predictive Analysis shows "what is likely to happen" by using previous data. The simplest data analysis example is like if last year I bought two dresses based on my savings and if this year my salary is increasing double then I can buy four dresses. But of course, it's not easy like this because you have to think about other circumstances like chances of prices of clothes is increased this year or maybe instead of dresses you want to buy a new bike, or you need to buy a house!

So here, this Analysis makes predictions about future outcomes based on current or past data. Forecasting is just an estimate. Its accuracy is based on how much detailed information you have and how much you dig in it.

Prescriptive Analysis

Prescriptive Analysis combines the insight from all previous Analysis to determine which action to take in a current problem or decision. Most data-driven companies are utilizing Prescriptive Analysis because predictive and descriptive Analysis are not enough to improve data performance. Based on current situations and problems, they analyze the data and make decisions.

Exploratory Data Analysis ~ EDA

Exploratory Data Analysis (EDA) is an approach/philosophy for data analysis that employs a variety of techniques (mostly graphical) to

1. maximize insight into a data set;
2. uncover underlying structure;
3. extract important variables;
4. detect outliers and anomalies;
5. test underlying assumptions;
6. develop parsimonious models; and
7. determine optimal factor settings.

The EDA approach is precisely that--an approach--not a set of techniques, but an attitude/philosophy about how a data analysis should be carried out.

EDA is not identical to statistical graphics although the two terms are used almost interchangeably. Statistical graphics is a collection of techniques--all

graphically based and all focusing on one data characterization aspect. EDA encompasses a larger venue; EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model. EDA is not a mere collection of techniques; EDA is a philosophy as to how we dissect a data set; what we look for; how we look; and how we interpret. It is true that EDA heavily uses the collection of techniques that we call "statistical graphics", but it is not identical to statistical graphics per se.

History of EDA

EDA holds its roots from the seminal work in EDA that is Exploratory Data Analysis, Tukey, (1977). Over the years it has benefitted from other noteworthy publications such as Data Analysis and Regression, Mosteller and Tukey (1977), Interactive Data Analysis, Hoaglin (1977), The ABC's of EDA, Velleman and Hoaglin (1981) and has gained a large following as "the" way to analyze a data set.

Techniques of EDA

Most EDA techniques are graphical in nature with a few quantitative techniques. The reason for the heavy reliance on graphics is that by its very nature the main role of EDA is to open-mindedly explore, and graphics gives the analysts unparalleled power to do so, enticing the data to reveal its structural secrets, and being always ready to gain some new, often unsuspected, insight into the data. In combination with the natural pattern-recognition capabilities that we all possess, graphics provides, of course, unparalleled power to carry this out.

The particular graphical techniques employed in EDA are often quite simple, consisting of various techniques of:

1. Plotting the raw data (such as data traces, histograms, bi-histograms, probability plots, lag plots, block plots, and Youden plots.
2. Plotting simple statistics such as mean plots, standard deviation plots, box plots, and main effects plots of the raw data.
3. Positioning such plots so as to maximize our natural pattern-recognition abilities, such as using multiple plots per page.

Exploratory Data Analysis vs Classical Data Analysis (EDA vs CDA)

EDA is a data analysis approach. Besides EDA other data analysis approaches also exist and now question arises how does EDA differ from these other approaches. Three popular data analysis approaches are:

1. Classical
2. Exploratory (EDA)
3. Bayesian

These three approaches are similar in that they all start with a general science/engineering problem and all yield science/engineering conclusions. The difference is the sequence and focus of the intermediate steps.

- For classical analysis, the sequence is
  Problem → Data → Model → Analysis → Conclusions
- For EDA, the sequence is
  Problem → Data → Analysis → Model → Conclusions
- For Bayesian, the sequence is
  Problem → Data → Model → Prior Distribution → Analysis → Conclusions

Thus, for classical analysis, the data collection is followed by the imposition of a model (normality, linearity, etc.) and the analysis, estimation, and testing that follows are focused on the parameters of that model. For EDA, the data collection is not followed by a model imposition; rather it is followed immediately by analysis with a goal of inferring what model would be appropriate. Finally, for a Bayesian analysis, the analyst attempts to incorporate scientific/engineering knowledge/expertise into the analysis by imposing a data-independent distribution on the parameters of the selected model; the analysis thus consists of formally combining both the prior distribution on the parameters and the collected data to jointly make inferences and/or test assumptions about the model parameters.

In the real world, data analysts freely mix elements of all of the above three approaches (and other approaches). The above distinctions were made to emphasize the major differences among the three approaches.

Focusing on EDA versus classical, these two approaches differ as follows:

1. Models
2. Focus
3. Techniques
4. Rigor
5. Data Treatment
6. Assumptions

Data Interpretation

Data interpretation is the process of reviewing data through some predefined processes which will help assign some meaning to the data and arrive at a relevant conclusion. It involves taking the result of data analysis, making inferences on the relations studied, and using them to conclude.

Therefore, before one can talk about interpreting data, they need to be analyzed first. From the previous two sections we know that data analysis is the process of ordering, categorizing, manipulating, and summarizing data to obtain

answers to research questions. It is usually the first step taken towards data interpretation. It is evident that the interpretation of data is very important, and as such needs to be done properly. Therefore, researchers have identified some data interpretation methods to aid this process.

Methods involved in Data Interpretation

Data interpretation methods are how analysts help people make sense of numerical data that has been collected, analyzed and presented. Data, when collected in raw form, may be difficult for the layman to understand, which is why analysts need to break down the information gathered so that others can make sense of it. For example, when founders are pitching to potential investors, they must interpret data (e.g., market size, growth rate, etc.) for better understanding.

There are 2 main methods in which this can be done, namely; quantitative methods and qualitative methods.

Qualitative Data Interpretation Method

The qualitative data interpretation method is used to analyze qualitative data, which is also known as categorical data. This method uses texts, rather than numbers or patterns to describe data. Qualitative data is usually gathered using a wide variety of person-to-person techniques, which may be difficult to analyze compared to the quantitative research method.

Unlike the quantitative data which can be analyzed directly after it has been collected and sorted, qualitative data needs to first be coded into numbers before it can be analyzed. This is because texts are usually cumbersome, and will take more time and result in a lot of errors if analyzed in its original state. Coding done by the analyst should also be documented so that it can be reused by others and also analyzed. There are 2 main types of qualitative data, namely; nominal and ordinal data. These 2 data types are both interpreted using the same method, but ordinal data interpretation is quite easier than that of nominal data.

In most cases, ordinal data is usually labelled with numbers during the process of data collection, and coding may not be required. This is different from nominal data that still needs to be coded for proper interpretation.

Quantitative Data Interpretation Method

The quantitative data interpretation method is used to analyze quantitative data, which is also known as numerical data. This data type contains numbers and is therefore analyzed with the use of numbers and not texts. Quantitative data are of 2 main types, namely; discrete and continuous data. Continuous data is further divided into interval data and ratio data, with all the data types being numeric.

Due to its natural existence as a number, analysts do not need to employ the coding technique on quantitative data before it is analyzed. The process of analyzing quantitative data involves statistical modelling techniques such as

standard deviation, mean and median. Some of the statistical methods used in analyzing quantitative data are highlighted below:

1. Mean: The mean is a numerical average for a set of data and is calculated by dividing the sum of the values by the number of values in a dataset. It is used to get an estimate of a large population from the dataset obtained from a sample of the population.
2. Standard deviation: This technique is used to measure how well the responses align with or deviates from the mean. It describes the degree of consistency within the responses; together with the mean, it provides insight into data sets.
3. Frequency distribution: This technique is used to assess the demography of the respondents or the number of times a particular response appears in research. It is extremely keen on determining the degree of intersection between data points.

Some other interpretation processes of quantitative data not used as popularly as the previous three and uses quite hold a small niche includes:

- Regression analysis
- Cohort analysis
- Predictive and prescriptive analysis

Important points while collecting data for accurate data interpretation

- Identify the Required Data Type

  Researchers need to identify the type of data required for particular research. Is it nominal, ordinal, interval, or ratio data? The key to collecting the required data to conduct research is to properly understand the research question. If the researcher can understand the research question, then he can identify the kind of data that is required to carry out the research.

- Avoid Biases

  There are different kinds of biases a researcher might encounter when collecting data for analysis. Although biases sometimes come from the researcher, most of the biases encountered during the data collection process is caused by the respondent. There are 2 main biases, that can be caused by the President, namely; response bias and non-response bias. Researchers may not be able to eliminate these biases, but there are ways in which they can be avoided and reduced to a minimum.

  Response biases are biases that are caused by respondents intentionally giving wrong answers to responses, while non-response bias occurs when the respondents don't give answers to questions at all. Biases are capable of affecting the process of data interpretation.

- Use Close Ended Surveys

  Although open-ended surveys are capable of giving detailed information about the questions and allow respondents to fully express themselves, it is not the best kind of survey for data interpretation. It requires a lot of coding before the data can be analyzed. Close-ended surveys, on the other hand, restrict the respondents' answer to some predefined options, while simultaneously eliminating irrelevant data. This way, researchers can easily analyze and interpret data.

Data Visualization

Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large data sets. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics.

Data visualization is one of the steps of the data science process, which states that after data has been collected, processed and modeled, it must be visualized for conclusions to be made. Data visualization is also an element of the broader data presentation architecture (DPA) discipline, which aims to identify, locate, manipulate, format and deliver data in the most efficient way possible.

Data visualization is important for almost every career. It can be used by teachers to display student test results, by computer scientists exploring advancements in artificial intelligence (AI) or by executives looking to share information with stakeholders. It also plays an important role in big data projects. As businesses accumulated massive collections of data during the early years of the big data trend, they needed a way to quickly and easily get an overview of their data. Hence, visualization tools were a natural fit in these cases. Visualization is central to advanced analytics for similar reasons. When a data scientist is writing advanced predictive analytics or machine learning (ML) algorithms, it becomes important to visualize the outputs to monitor results and ensure that models are performing as intended. This is because visualizations of complex algorithms are generally easier to interpret than numerical outputs.

Data visualization and big data

The increased popularity of big data and data analysis projects have made visualization more important than ever. Companies are increasingly using machine learning to gather massive amounts of data that can be difficult and slow to sort through, comprehend and explain. Visualization offers a means to speed this up and present information to business owners and stakeholders in ways they can understand. Big data visualization often goes beyond the typical techniques used in

normal visualization, such as pie charts, histograms and corporate graphs. It instead uses more complex representations, such as heat maps and fever charts. Big data visualization requires powerful computer systems to collect raw data, process it and turn it into graphical representations that humans can use to quickly draw insights.

While big data visualization can be beneficial, it can pose several disadvantages to organizations. They are as follows:

- To get the most out of big data visualization tools, a visualization specialist must be hired. This specialist must be able to identify the best data sets and visualization styles to guarantee organizations are optimizing the use of their data.
- Big data visualization projects often require involvement from IT, as well as management, since the visualization of big data requires powerful computer hardware, efficient storage systems and even a move to the cloud.
- The insights provided by big data visualization will only be as accurate as the information being visualized. Therefore, it is essential to have people and processes in place to govern and control the quality of corporate data, metadata and data sources.

Techniques involved in visualization

In the early days of visualization, the most common visualization technique was using a Microsoft Excel™ spreadsheet to transform the information into a table, bar graph or pie chart. While these visualization methods are still commonly used, more intricate techniques are now available, including the following:

- infographics
- bubble clouds
- bullet graphs
- heat maps
- fever charts
- time series charts

Some other popular techniques are as follows.

- Line charts. This is one of the most basic and common techniques used. Line charts display how variables can change over time.
- Area charts. This visualization method is a variation of a line chart; it displays multiple values in a time series -- or a sequence of data collected at consecutive, equally spaced points in time.
- Scatter plots. This technique displays the relationship between two variables. A scatter plot takes the form of an x- and y-axis with dots to represent data points.

- Treemaps. This method shows hierarchical data in a nested format. The size of the rectangles used for each category is proportional to its percentage of the whole. Treemaps are best used when multiple categories are present, and the goal is to compare different parts of a whole.
- Population pyramids. This technique uses a stacked bar graph to display the complex social narrative of a population. It is best used when trying to display the distribution of a population.

Importance of Visualization

Data visualization provides a quick and effective way to communicate information in a universal manner using visual information. The practice can also help businesses identify which factors affect customer behavior; pinpoint areas that need to be improved or need more attention; make data more memorable for stakeholders; understand when and where to place specific products; and predict sales volumes.

Other benefits of data visualization include the following:

- the ability to absorb information quickly, improve insights and make faster decisions;
- an increased understanding of the next steps that must be taken to improve the organization;
- an improved ability to maintain the audience's interest with information they can understand;
- an easy distribution of information that increases the opportunity to share insights with everyone involved;
- eliminate the need for data scientists since data is more accessible and understandable; and
- an increased ability to act on findings quickly and, therefore, achieve success with greater speed and less mistakes

A little intro into the problem at hand

A semi hypothetical company 'FinMan' with various similar establishments popping from a simple name search in the google database, considering all these the most match to the problem initiated to us corroborates with either these three possible companies

1. Paarth FinMan, located at Kolkata, India.
2. Finman AG, located at Zollikon, ZÜRICH, Switzerland.

Since this name very closely matches along with the problem statement that was given in a job search hackathon by Analytics Vidya in the field of Data Science and also shared in the Kaggle community regarding the same field.

Given the task we are supposed to:

1. Study the data provided.
2. Find inconsistencies in the data.
3. Eventually clean the data by recommended methods.
4. Set up the required datatypes for each fields of data.
5. Fix the data to make it code readable.
6. Categorize or randomize data by requirement.
7. Regress logistically primarily to get an interpretation of customer behavior.
8. If Logistical processes fails it is opinionated to shift to other algorithms with better performance.
9. Assess the metrics of the interpretation and check the efficacy of the method.
10. Make informed guess about potential buyers of health insurances from a fixed pool of buyers based on the knowledge gained of the customer interaction from the interpretation stated in the last point.
11. Provide a visual compilation of the buyer pool back to the customer('FinMan') with all leads and solutions.

Logistic Regression

Logistic regression analysis studies the association between a categorical dependent variable and a set of independent (explanatory) variables. The name logistic regression is used when the dependent variable has only two values, such as 0 and 1 or Yes and No.

Logistic regression competes with discriminant analysis as a method for analyzing categorical-response variables. Many statisticians feel that logistic regression is more versatile and better suited for modelling most situations than is discriminant analysis. This is because logistic regression does not assume that the independent variables are normally distributed, as discriminant analysis does.

Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes.

Comparison to linear regression

Linear Regression and logistic regression can predict different things:

- Linear regression predictions are continuous (numbers in a range).
- Logistic regression predictions are discrete (only specific values or categories are allowed). We can also view probability scores underlying the model's classifications.

Types of logistic regression

- Binary (Pass/Fail)
- Multi (Cats, Dogs, Sheep)
- Ordinal (Low, Medium, High)

The Logit and Logistic Transformations

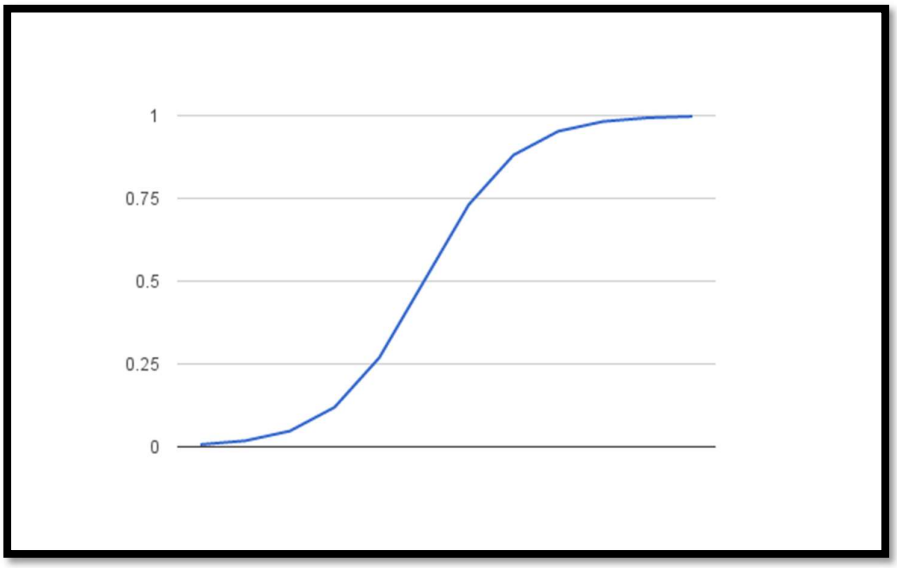
In multiple regression, a mathematical model of a set of explanatory variables is used to predict the mean of a continuous dependent variable. In logistic regression, a mathematical model of a set of explanatory variables is used to predict a logit transformation of the dependent variable. Suppose the numerical values of 0 and 1 are assigned to the two outcomes of a binary variable. Often, the 0 represents a negative response and the 1 represents a positive response. The mean of this variable will be the proportion of positive responses. If p is the proportion of observations with an outcome of 1, then 1-p is the probability of an outcome of 0. The ratio *p/(1-p)* is called the odds and the logit is the logarithm of the odds, or just log odds. Mathematically, the logit transformation is written

$$l = \text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

The logistic transformation is the inverse of the logit transformation. It is written

$$p = \text{logistic}(l) = \frac{e^l}{1+e^l}$$

This, gives us the equation for the sigmoid, which in perfect scenario gives us a curve of the likes described below.



The Log Odds Ratio Transformation

The difference between two log odds can be used to compare two proportions, such as that of males versus females. Mathematically, this difference is written

$$\begin{aligned} l_1 - l_2 &= \text{logit}(p_1) - \text{logit}(p_2) \\ &= \ln\left(\frac{p_1}{1-p_1}\right) - \ln\left(\frac{p_2}{1-p_2}\right) \\ &= \ln\left(\frac{\left(\frac{p_1}{1-p_1}\right)}{\left(\frac{p_2}{1-p_2}\right)}\right) \\ &= \ln\left(\frac{p_1(1-p_2)}{p_2(1-p_1)}\right) \\ &= \ln(OR_{1,2}) \end{aligned}$$

This difference is often referred to as the log odds ratio. The odds ratio is often used to compare proportions across groups. Note that the logistic transformation is closely related to the odds ratio. The reverse relationship is

$$OR_{1,2} = \mathrm{e}^{(l_1 - l_2)}$$

The Logistic Regression and Logit Models

In logistic regression, a categorical dependent variable Y having G (usually G = 2) unique values is regressed on a set of *p* independent variables $X_1, X_2, \ldots, X_p$. Since the names of these partitions are arbitrary, we often refer to them by consecutive numbers. That is, in the discussion below, Y will take on the values 1, 2, … *G*. Let,

$$\mathrm{X} = (X_1, X_2, \cdots, X_p)$$

$$B_g = \begin{pmatrix} \beta_{g1} \\ \vdots \\ \beta_{gp} \end{pmatrix}$$

The logistic regression model is given by the *G* equations

$$\begin{aligned} \ln\left(\frac{p_g}{p_1}\right) &= \ln\left(\frac{P_g}{P_1}\right) + \beta_{g1}X_1 + \beta_{g2}X_2 + \cdots + \beta_{gp}X_p \\ &= \ln\left(\frac{P_g}{P_1}\right) + \mathrm{XB}_g \end{aligned}$$

Here, p_g is the probability that an individual with values $X_1, X_2, \ldots, X_p$ is in outcome g. That is,

$$p_g = \Pr(Y = g \mid \mathbf{X})$$

Usually $X_1 \equiv 1$ (that is, an intercept is included), but this is not necessary. The quantities $P_1, P_2, \ldots, P_G$, represent the prior probabilities of outcome membership. If these prior probabilities are assumed equal, then the term $ln\ (P_g/P_1)$ becomes zero and drops out. If the priors are not assumed equal, they change the values of the intercepts in the logistic regression equation.

Outcome one is called the *reference value*. The regression coefficients β_{11}, β_{12}, …, β_{1p} for the reference value are set to zero. The choice of the reference value is arbitrary. Usually, it is the most frequent value or a control outcome to which the other outcomes are to be compared. This leaves G-1 logistic regression equations in the logistic model.

The β's are population regression coefficients that are to be estimated from the data. Their estimates are represented by b's. The β's represents unknown parameters to be estimated, while the b's are their estimates.

These equations are linear in the logits of p. However, in terms of the probabilities, they are nonlinear. The corresponding nonlinear equations are

$$\mathrm{p_g} = \mathrm{Prob}(Y = g \mid \mathbf{X}) = \frac{e^{\mathrm{XB}_g}}{1 + e^{\mathrm{XB}_2} + e^{\mathrm{XB}_3} + \cdots + e^{\mathrm{XB}_G}}$$

since $e^{XB_1} = 1$ because all of its regression coefficients are zero.

A note on the names of the models. Often, all of these models are referred to as logistic regression models. However, when the independent variables are coded as ANOVA type models, they are sometimes called *logit models*.

Another note about the interpretation of e^{XB} may be useful. Using the fact that $e^{a+b} = (e^a)(e^b)$, e^{XB} may be re-expressed as follows

$$\begin{aligned} e^{\mathrm{XB}} &= e^{\beta_1 \mathrm{X}_1 + \beta_2 \mathrm{X}_2 + \cdots + \beta_p \mathrm{X}_p} \\ &= e^{\beta_1 \mathrm{X}_1} e^{\beta_2 \mathrm{X}_2} \cdots e^{\beta_p \mathrm{X}_p} \end{aligned}$$

This shows that the final value is the product of its individual terms.

Solving the Likelihood Equations

To improve notation, let

$$\begin{aligned}\pi_{gj} &= \text{Prob}(Y = g \mid \mathrm{X}_j) \\ &= \frac{e^{\mathrm{X}_j\mathrm{B}_g}}{e^{\mathrm{X}_j\mathrm{B}_1} + e^{\mathrm{X}_j\mathrm{B}_2} + \cdots + e^{\mathrm{X}_j\mathrm{B}_G}} \\ &= \frac{e^{\mathrm{X}_j\mathrm{B}_g}}{\sum_{s=1}^{G} e^{\mathrm{X}_j\mathrm{B}_s}}\end{aligned}$$

The likelihood for a sample of N observations is then given by

$$l = \prod_{j=1}^{N}\prod_{g=1}^{G} \pi_{gj}^{\,y_{gj}}$$

where y_{gj} is one if the j^{th} observation is in outcome g and zero otherwise.

Using the fact that $\sum_{g=1}^{G} y_{gj} = 1$, the log likelihood, L, is given by

$$\begin{aligned}L = \ln(l) &= \sum_{j=1}^{N}\sum_{g=1}^{G} \mathrm{y}_{gj} \ln(\pi_{gj}) \\ &= \sum_{j=1}^{N}\sum_{g=1}^{G} \mathrm{y}_{gj} \ln\left(\frac{\mathrm{e}^{\mathrm{X}_j\mathrm{B}_g}}{\sum_{s=1}^{G} \mathrm{e}^{\mathrm{X}_j\mathrm{B}_s}}\right) \\ &= \sum_{\mathrm{j}=1}^{N}\left[\sum_{\mathrm{g}=1}^{G} y_{gj}\mathrm{X}_j\mathrm{B}_g - \ln\left(\sum_{g=1}^{G} \mathrm{e}^{\mathrm{X}_j\mathrm{B}_g}\right)\right]\end{aligned}$$

Maximum likelihood estimates of the β's are those values that maximize this log likelihood equation. This is accomplished by calculating the partial derivatives and setting them to zero. The resulting likelihood equations are

$$\frac{\partial L}{\partial \beta_{ik}} = \sum_{j=1}^{N} x_{kj}\left(y_{ig} - \pi_{ig}\right)$$

for g = 1, 2, ..., G and k = 1, 2, ..., p. Actually, since all coefficients are zero for g= 1, the effective range of g is from 2 to G.

Because of the nonlinear nature of the parameters, there is no closed-form solution to these equations and they must be solved iteratively. The Newton-Raphson method as described in Albert and Harris (1987) is used to solve these equations. This method makes use of the information matrix, I(β), which is formed from the matrix of second partial derivatives. The elements of the information matrix are given by

$$\frac{\partial^2 L}{\partial \beta_{ik} \partial \beta_{ik'}} = -\sum_{j=1}^{N} x_{kj} x_{k'j} \pi_{ig} \left(1 - \pi_{ig}\right)$$

$$\frac{\partial^2 L}{\partial \beta_{ik} \partial \beta_{i'k'}} = \sum_{j=1}^{N} x_{kj} x_{k'j} \pi_{ig} \pi_{i'g}$$

The information matrix is used because the asymptotic covariance matrix of the maximum likelihood estimates is equal to the inverse of the information matrix. That is,

$$V\left(\hat{\beta}\right) = I(\beta)^{-1}$$

This covariance matrix is used in the calculation of confidence intervals for the regression coefficients, odds ratios, and predicted probabilities.

Interpretation of Regression Coefficients

The interpretation of the estimated regression coefficients is not as easy as in multiple regression. In logistic regression, not only is the relationship between X and Y nonlinear, but also, if the dependent variable has more than two unique values, there are several regression equations.

Consider the usual case of a binary dependent variable, Y, and a single independent variable, X. Assume that Y is coded so it takes on the values 0 and 1. In this case, the logistic regression equation is

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X$$

Now consider impact of a unit increase in X. The logistic regression equation becomes

$$\begin{aligned} \ln\left(\frac{p'}{1-p'}\right) &= \beta_0 + \beta_1 (X+1) \\ &= \beta_0 + \beta_1 X + \beta_1 \end{aligned}$$

We can isolate the slope by taking the difference between these two equations. We have

$$\begin{aligned}\beta_1 &= \beta_0 + \beta_1(X+1) - (\beta_0 + \beta_1 X) \\ &= \ln\left(\frac{p'}{1-p'}\right) - \ln\left(\frac{p}{1-p}\right) \\ &= \ln\left(\frac{\frac{p'}{1-p'}}{\frac{p}{1-p}}\right) \\ &= \ln\left(\frac{odds'}{odds}\right)\end{aligned}$$

That is, β_1 is the log of the ratio of the odds at *X+1* and *X*. Removing the logarithm by exponentiating both sides gives

$$e^{\beta_1} = \frac{odds'}{odds}$$

The regression coefficient β_1 is interpreted as the log of the odds ratio comparing the odds after a one unit increase in *X* to the original odds. Note that, unlike multiple regression, the interpretation of β_1 depends on the particular value of *X* since the probability values, the *p*'s, will vary for different *X*.

Binary X

When X can take on only two values, say 0 and 1, the above interpretation becomes even simpler. Since there are only two possible values of *X*, there is a unique interpretation for β_1 given by the log of the odds ratio. In mathematical terms, the meaning of β_1 is then

$$\beta_1 = \ln\left(\frac{odds(X=1)}{odds(X=0)}\right)$$

Multiple Independent Variables

When there are multiple independent variables, the interpretation of each regression coefficient becomes more difficult, especially if interaction terms are included in the model. In general, however, the regression coefficient is interpreted the same as above, except that the caveat 'holding all other independent variables constant' must be added. The question becomes, can the value of this independent variable be increased by one without changing any of the other variables. If it can, then the interpretation is as before. If not, then some type of conditional statement must be added that accounts for the values of the other variables.

Multinomial Dependent Variable

When the dependent variable has more than two values, there will be more than one regression equation. In fact, the number of regression equations is equal to one less than the number of outcomes. This makes interpretation more difficult

because there are several regression coefficients associated with each independent variable. In this case, care must be taken to understand what each regression equation is predicting.

Decision Trees

Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree. Learned trees can also be re-represented as sets of if-then rules to improve human readability. These learning methods are among the most popular of inductive inference algorithms and have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants.

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. Each node in the tree specifies a test of some attribute of the instance, and each branch descending from that node corresponds to one of the possible values for this attribute. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute in the given example. This process is then repeated for the subtree rooted at the new node.

Appropriate Problems for Decision Tree Learning

Although a variety of decision tree learning methods have been developed with somewhat differing capabilities and requirements, decision tree learning is generally best suited to problems with the following characteristics:

- Instances are represented by attribute-value pairs:

  Instances are described by a fixed set of attributes and their values. The easiest situation for decision tree learning is when each attribute takes on a small number of disjoint possible values. However, extensions to the basic algorithm allow handling real-valued attributes as well.

- The target function has discrete output values:

  Decision tree methods easily extend to learning functions with more than two possible output values. A more substantial extension allows learning target functions with real-valued outputs, though the application of decision trees in this setting is less common.

- Disjunctive descriptions may be required.

  As noted above, decision trees naturally represent disjunctive expressions

- The training data may contain errors.

  Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.

- The training data may contain missing attribute values.

  Decision tree methods can be used even when some training examples have unknown values.

Many practical problems have been found to fit these characteristics. Decision tree learning has therefore been applied to problems such as learning to classify medical patients by their disease, equipment malfunctions by their cause, and loan applicants by their likelihood of defaulting on payments. Such problems, in which the task is to classify examples into one of a discrete set of possible categories, are often referred to as classification problems.

The Basic Decision Tree Learning Algorithm

Most algorithms that have been developed for learning decision trees are variations on a core algorithm that employs a top-down, greedy search through the space of possible decision trees. This approach is exemplified by the ID3 algorithm (Quinlan 1986) and its successor C4.5 (Quinlan 1993).

The ID3, learns decision trees by constructing them top- down, beginning with the question "which attribute should be tested at the root of the tree?", to answer this question, each instance attribute is evaluated using a statistical test to determine how well it alone classifies the training examples. The best attribute is selected and used as the test at the root node of the tree. A descendant of the root node is then created for each possible value of this attribute, and the training examples are sorted to the appropriate descendant node (i.e., down the branch corresponding to the example's value for this attribute). The entire process is then repeated using the training examples associated with each descendant node to select the best attribute to test at that point in the tree. This forms a greedy search for an acceptable decision tree, in which the algorithm never backtracks to reconsider earlier choices. A simplified version of the algorithm, specialized to learning Boolean-valued functions (i.e., concept learning), is described in the following tabulation.

ID3(Examples, Target_attribute, Attributes)

Examples are the training examples. Target_attribute is the attribute whose value is to be predicted by the tree. Attributes is a list of other attributes that may be tested by the learned decision tree. Returns a decision tree that correctly classifies the given Examples.

- Create a Root node for the tree
- If all Examples are positive, Return the single-node tree Root, with label = +
- If all Examples are negative, Return the single-node tree Root, with label = -
- If Attributes is empty, Return the single-node tree Root, with label = most common value of Target_attribute in Example
- Otherwise Begin
    - A ← the attribute from Attributes that best* classifies Examples
    - The decision attribute for Root ← A
    - For each possible value, v_i, of A,
        - Add a new tree branch below Root, corresponding to the test A = v_i
        - Let Examples, v_i, be the subset of Examples that have value v_i for A
        - If Examples, v_i, is empty
            - Then below this new branch add a leaf node with label = most common value of Target_attribute in Examples
            - Else below this new branch add the subtree ID3(Examples, v_i, Target_attribute, Attributes – {A})
- End
- Return Root

* The best attribute is the one with highest information gain

Summary of the ID3 algorithm specialized to learning Boolean-valued functions. ID3 is a greedy algorithm that grows the tree top-down, at each node selecting the attribute that best classifies the local training examples. This process continues until the tree perfectly classifies the training examples, or until all attributes have been used.

Which Attribute Is the Best Classifier?

The central choice in the ID3 algorithm is selecting which attribute to test at each node in the tree. We would like to select the attribute that is most useful for classifying examples. What is a good quantitative measure of the worth of an attribute? We will define a statistical property, called information gain, that measures how well a given attribute separates the training examples according to their target classification. ID3 uses this information gain measure to select among the candidate attributes at each step while growing the tree.

Entropy Measures Homogeneity

In order to define information gain precisely, we begin by defining a measure commonly used in information theory, called entropy, that characterizes the impurity of an arbitrary collection of examples. Given a collection S, containing positive and negative examples of some target concept, the entropy of S relative to this Boolean classification is

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

where $p_{\oplus}$, is the proportion of positive examples in S and $p_{\ominus}$, is the proportion of negative examples in S. In all calculations involving entropy we define 0 log 0 to be 0.

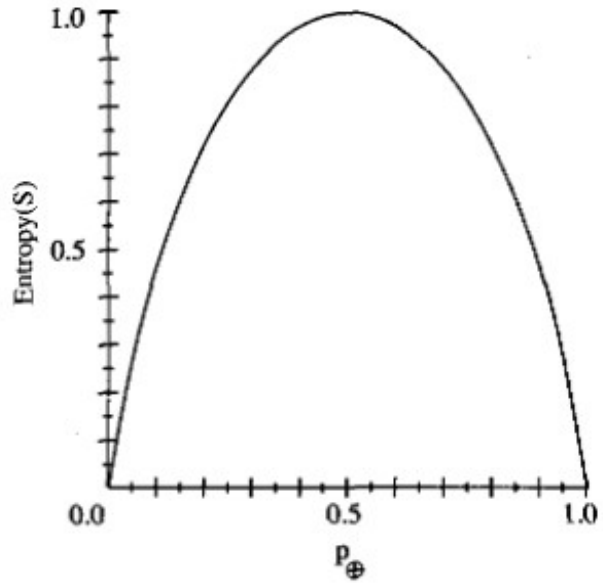
To illustrate, suppose S is a collection of 14 examples of some Boolean concept, including 9 positive and 5 negative examples (we adopt the notation [9+, 5-1 to summarize such a sample of data). Then the entropy of S relative to this Boolean classification is

$$Entropy([9+, 5-]) = -(9/14) \log_2(9/14) - (5/14) \log_2(5/14)$$
$$= 0.940$$

Notice that the entropy is 0 if all members of S belong to the same class. For example, if all members are positive ($p_{\oplus}$ = 1), then $p_{\ominus}$, is 0, and,

Entropy(S) = $-1 \times \log_2(1) - 0 \times \log_2 0 = -1 \times 0 - 0 \times \log_2 0 = 0$.

Note the entropy is 1 when the collection contains an equal number of positive and negative examples. If the collection contains unequal numbers of positive and negative examples, the entropy is between 0 and 1. Following figure shows the form of the entropy function relative to a Boolean classification, as $p_{\oplus}$, varies between 0 and 1.



The entropy function relative to a Boolean classification, as the proportion, $p_\oplus$, of positive examples varies between 0 and 1.

One interpretation of entropy from information theory is that it specifies the minimum number of bits of information needed to encode the classification of an arbitrary member of S (i.e., a member of S drawn at random with uniform probability). For example, if $p_\oplus$ is 1, the receiver knows the drawn example will be positive, so no message need be sent, and the entropy is zero. On the other hand, if $p_\oplus$ is 0.5, one bit is required to indicate whether the drawn example is positive or negative. If $p_\oplus$ is 0.8, then a collection of messages can be encoded using on average less than 1 bit per message by assigning shorter codes to collections of positive examples and longer codes to less likely negative examples. Thus far we have discussed entropy in the special case where the target classification is Boolean. More generally, if the target attribute can take on c different values, then the entropy of S relative to this c-wise classification is defined as

$$Entropy(S) \equiv \sum_{i=1}^{c} -p_i \log_2 p_i$$

Where p_i is the proportion of S belonging to class i. Note the logarithm is still base 2 because entropy is a measure of the expected encoding length measured in bits. Note also that if the target attribute can take on c possible values, the entropy can be as large as $\log_2 c$

Information Gain Measures the Expected Reduction in Entropy

Given entropy as a measure of the impurity in a collection of training examples, we can now define a measure of the effectiveness of an attribute in classifying the training data. The measure we will use, called information gain, is

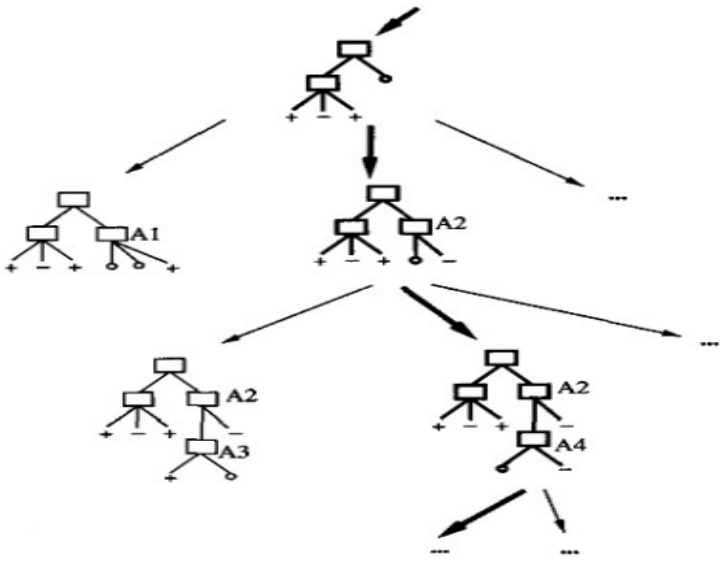
simply the expected reduction in entropy caused by partitioning the examples according to this attribute. More precisely, the information gain, Gain (S, A) of an attribute A, relative to a collection of examples S, is defined as

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where Values(A) is the set of all possible values for attribute A, and S, is the subset of S for which attribute A has value *v* (i.e., $S_v = \{s \in S | A(s) = v\}$). Note the first term in Equation is just the entropy of the original collection S, and the second term is the expected value of the entropy after S is partitioned using attribute A. The expected entropy described by this second term is simply the sum of the entropies of each subset S_v, weighted by the fraction of examples $\frac{|S_v|}{|S|}$ that belong to S_v. Gain (S, A) is therefore the expected reduction in entropy caused by knowing the value of attribute A. Put another way, Gain (S, A) is the information provided about the target &action value, given the value of some other attribute A. The value of Gain (S, A) is the number of bits saved when encoding the target value of an arbitrary member of S, by knowing the value of attribute A. Information gain is precisely the measure used by ID3 to select the best attribute at each step in growing the tree.

Hypothesis Space Search in Decision Tree Learning

As with other inductive learning methods, ID3 can be characterized as searching a space of hypotheses for one that fits the training examples. The hypothesis space searched by ID3 is the set of possible decision trees. ID3 performs a simple-to- complex, hill-climbing search through this hypothesis space, beginning with the empty tree, then considering progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data. The evaluation function that guides this hill-climbing search is the information gain measure. This search is depicted in the following figure.



Hypothesis space search by ID3. ID3 searches through the space of possible decision trees from simplest to increasingly complex, guided by the information gain heuristic.

By viewing ID3 in terms of its search space and search strategy, we can get some insight into its capabilities and limitations

- ID3's hypothesis space of all decision trees is a complete space of finite discrete-valued functions, relative to the available attributes. Because every finite discrete-valued function can be represented by some decision tree, ID3 avoids one of the major risks of methods that search incomplete hypothesis spaces (such as methods that consider only conjunctive hypotheses): that the hypothesis space might not contain the target function.
- ID3 maintains only a single current hypothesis as it searches through the space of decision trees. This contrast, for example, with the earlier version space Candidate-Elimination method, which maintains the set of all hypotheses consistent with the available training examples. By determining only a single hypothesis, ID3 loses the capabilities that follow from explicitly representing all consistent hypotheses. For example, it does not have the ability to determine how many alternative decision trees are consistent with the available training data, or to pose new instance queries that optimally resolve among these competing hypotheses.
- ID3 in its pure form performs no backtracking in its search. Once it selects an attribute to test at a particular level in the tree, it never backtracks to reconsider this choice. Therefore, it is susceptible to the usual risks of hill-climbing search without backtracking: converging to locally optimal solutions that are not globally optimal. In the case of ID3, a locally optimal solution corresponds to the decision tree it selects along the single search path it explores. However, this locally optimal solution may be less desirable than trees that would have been encountered along a different branch of the search. Below we discuss an extension that adds a form of backtracking (post-pruning the decision tree).

- ID3 uses all training examples at each step in the search to make statistically based decisions regarding how to refine its current hypothesis. This contrasts with methods that make decisions incrementally, based on individual training examples (e.g., FIND-S or CANDIDATE-ELIMINATION). One advantage of using statistical properties of all the examples (e.g., information gain) is that the resulting search is much less sensitive to errors in individual training examples. ID3 can be easily extended to handle noisy training data by modifying its termination criterion to accept hypotheses that imperfectly fit the training data.

Inductive Bias in Decision Tree Learning

Given a collection of training examples, there are typically many decision trees consistent with these examples. Describing the inductive bias of ID3 therefore consists of describing the basis by which it chooses one of these consistent hypotheses over the others. Which of these decision trees does ID3 choose? It chooses the first acceptable tree it encounters in its simple-to-complex, hill-climbing search through the space of possible trees. Roughly speaking, then, the ID3 search strategy

(a) selects in favor of shorter trees over longer ones, and
(b) selects trees that place the attributes with highest information gain closest to the root.

Because of the subtle interaction between the attribute selection heuristic used by ID3 and the particular training examples it encounters, it is difficult to characterize precisely the inductive bias exhibited by ID3. However, we can approximately characterize its bias as a preference for short decision trees over complex trees.

Approximate inductive bias of ID3: Shorter trees are preferred over larger trees.

In fact, one could imagine an algorithm similar to ID3 that exhibits precisely this inductive bias. Consider an algorithm that begins with the empty tree and searches breadth first through progressively more complex trees, first considering all trees of depth 1, then all trees of depth 2, etc. Once it finds a decision tree consistent with the training data, it returns the smallest consistent tree at that search depth (e.g., the tree with the fewest nodes). Let us call this breadth-first search algorithm BFS-ID3. BFS-ID3 finds a shortest decision tree and thus exhibits precisely the bias "shorter trees are preferred over longer trees." ID3 can be viewed as an efficient approximation to BFS-ID3, using a greedy heuristic search to attempt to find the shortest tree without conducting the entire breadth-first search through the hypothesis space.

Because ID3 uses the information gain heuristic and a hill climbing strategy, it exhibits a more complex bias than BFS-ID3. In particular, it does not always find the shortest consistent tree, and it is biased to favor trees that place attributes with high information gain closest to the root.

A closer approximation to the inductive bias of ID3: Shorter trees are preferred over longer trees. Trees that place high information gain attributes close to the root are preferred over those that do not.

Restriction Biases and Preference Biases

There is an interesting difference between the types of inductive bias exhibited by ID3 and by the CANDIDATE-ELIMINATION algorithm.

Consider the difference between the hypothesis space search in these two approaches:

- ID3 searches a complete hypothesis space (i.e., one capable of expressing any finite discrete-valued function). It searches incompletely through this space, from simple to complex hypotheses, until its termination condition is met (e.g., until it finds a hypothesis consistent with the data). Its inductive bias is solely a consequence of the ordering of hypotheses by its search strategy. Its hypothesis space introduces no additional bias.
- The version space CANDIDATE-ELIMINATION algorithm searches an incomplete hypothesis space (i.e., one that can express only a subset of the potentially teachable concepts). It searches this space completely, finding every hypothesis consistent with the training data. Its inductive bias is solely a consequence of the expressive power of its hypothesis representation. Its search strategy introduces no additional bias.

In brief, the inductive bias of ID3 follows from its search strategy, whereas the inductive bias of the CANDIDATE-ELIMINATION algorithm follows from the definition of its search space.

The inductive bias of ID3 is thus a preference for certain hypotheses over others (e.g., for shorter hypotheses), with no hard restriction on the hypotheses that can be eventually enumerated. This form of bias is typically called a preference bias (or, alternatively, a search bias). In contrast, the bias of the CANDIDATE-ELIMINATION algorithm is in the form of a categorical restriction on the set of hypotheses considered. This form of bias is typically called a restriction bias (or, alternatively, a language bias).

Why Prefer Short Hypotheses?

Philosophers and others have debated the question regarding ID3's inductive bias favoring shorter decision trees a sound basis for generalizing beyond the training data for centuries, and the debate remains unresolved to this day.

William of Occam was one of the first to discuss the question, around the year 1320, so this bias often goes by the name of Occam's razor.

Occam's razor: Prefer the simplest hypothesis that fits the data.

Of course, giving an inductive bias a name does not justify it. Why should one prefer simpler hypotheses? Notice that scientists sometimes appear to follow this inductive bias. Physicists, for example, prefer simple explanations for the motions of the planets, over more complex explanations. Why? One argument is that because there are fewer short hypotheses than long ones (based on straightforward combinatorial arguments), it is less likely that one will find a short hypothesis that coincidentally fits the training data. In contrast there are often many very complex hypotheses that fit the current training data but fail to generalize correctly to subsequent data. Consider decision tree hypotheses, for example. There are many more 500-node decisions trees than 5-node decision trees. Given a small set of 20 training examples, we might expect to be able to find many 500-node decision trees consistent with these, whereas we would be more surprised if a 5-node decision tree could perfectly fit this data. We might therefore believe the 5-node tree is less likely to be a statistical coincidence and prefer this hypothesis over the 500-node hypothesis.

Upon closer examination, it turns out there is a major difficulty with the above argument. By the same reasoning we could have argued that one should prefer decision trees containing exactly 17 leaf nodes with 11 nonleafy nodes, that use the decision attribute A1 at the root, and test attributes A2 through All, in numerical order. There are relatively few such trees, and we might argue (by the same reasoning as above) that our a priori chance of finding one consistent with an arbitrary set of data is therefore small. The difficulty here is that there are very many small sets of hypotheses that one can define-most of them rather arcane. Why should we believe that the small set of hypotheses consisting of decision trees with short descriptions should be any more relevant than the multitude of other small sets of hypotheses that we might define?

A second problem with the above argument for Occam's razor is that the size of a hypothesis is determined by the particular representation used internally by the learner. Two learners using different internal representations could therefore arrive at different hypotheses, both justifying their contradictory conclusions by Occam's razor!

This last argument shows that Occam's razor will produce two different hypotheses from the same training examples when it is applied by two learners that perceive these examples in terms of different internal representations. On this basis we might be tempted to reject Occam's razor altogether. However, consider the following scenario that examines the question of which internal representations might arise from a process of evolution and natural selection. Imagine a population of artificial learning agents created by a simulated evolutionary process involving reproduction, mutation, and natural selection of these agents. Let us assume that

this evolutionary process can alter the perceptual systems of these agents from generation to generation, thereby changing the internal attributes by which they perceive their world. For the sake of argument, let us also assume that the learning agents employ a fixed learning algorithm (say ID3) that cannot be altered by evolution. It is reasonable to assume that over time evolution will produce internal representation that make these agents increasingly successful within their environment. Assuming that the success of an agent depends highly on its ability to generalize accurately, we would therefore expect evolution to develop internal representations that work well with whatever learning algorithm and inductive bias is present. If the species of agents employs a learning algorithm whose inductive bias is Occam's razor, then we expect evolution to produce internal representations for which Occam's razor is a successful strategy. The essence of the argument here is that evolution will create internal representations that make the learning algorithm's inductive bias a self-fulfilling prophecy, simply because it can alter the representation easier than it can alter the learning algorithm.

Issues in Decision Tree Learning

Practical issues in learning decision trees include determining how deeply to grow the decision tree, handling continuous attributes, choosing an appropriate attribute selection measure, handling training data with missing attribute values, handling attributes with differing costs, and improving computational efficiency. Below we discuss each of these issues and extensions to the basic ID3 algorithm that address them. ID3 has itself been extended to address most of these issues, with the resulting system renamed C4.5 (Quinlan 1993)

Avoiding Overfitting the Data

The algorithm described in Table in page 30 grows each branch of the tree just deeply enough to perfectly classify the training examples. While this is sometimes a reasonable strategy, in fact it can lead to difficulties when there is noise in the data, or when the number of training examples is too small to produce a representative sample of the true target function. In either of these cases, this simple algorithm can produce trees that overfit the training examples.

We will say that a hypothesis overfits the training examples if some other hypothesis that fits the training examples less well actually performs better over the entire distribution of instances (i.e., including instances beyond the training set).

Definition: Given a hypothesis space H, a hypothesis $h \in H$ is said to overlit the training data if there exists some alternative hypothesis $h' \in H$, such that h has smaller error than h' over the training examples, but h' has a smaller error than h over the entire distribution of instances.

Overfitting is a significant practical difficulty for decision tree learning and many other learning methods. For example, in one experimental study of ID3 involving five different learning tasks with noisy, nondeterministic data (Mingers1989b), overfitting was found to decrease the accuracy of learned decision trees by 10-25% on most problems.

There are several approaches to avoiding overfitting in decision tree learning. These can be grouped into two classes:

- approaches that stop growing the tree earlier, before it reaches the point where it perfectly classifies the training data,
- approaches that allow the tree to overfit the data, and then post-prune the tree.

Although the first of these approaches might seem more direct, the second approach of post-pruning overfit trees has been found to be more successful in practice. This is due to the difficulty in the first approach of estimating precisely when to stop growing the tree.

Regardless of whether the correct tree size is found by stopping early or by post-pruning, a key question is what criterion is to be used to determine the correct final tree size. Approaches include:

- Use a separate set of examples, distinct from the training examples, to evaluate the utility of post-pruning nodes from the tree.
- Use all the available data for training, but apply a statistical test to estimate whether expanding (or pruning) a particular node is likely to produce an improvement beyond the training set. For example, Quinlan (1986) uses a chi-square test to estimate whether further expanding a node is likely to improve performance over the entire instance distribution, or only on the current sample of training data.
- Use an explicit measure of the complexity for encoding the training examples and the decision tree, halting growth of the tree when this encoding size is minimized. This approach, based on a heuristic called the Minimum Description Length principle, as well as in Quinlan and Rivest (1989) and Mehta et al. (199.5).

The first of the above approaches is the most common and is often referred to as a training and validation set approach. We discuss the two main variants of this approach below. In this approach, the available data are separated into two sets of examples: a training set, which is used to form the learned hypothesis, and a separate validation set, which is used to evaluate the accuracy of this hypothesis over subsequent data and, in particular, to evaluate the impact of pruning this hypothesis. The motivation is this: Even though the learner may be misled by random errors and coincidental regularities within the training set, the validation set is unlikely to exhibit the same random fluctuations. Therefore, the validation set can be expected to provide a safety check against overfitting the spurious

characteristics of the training set. Of course, it is important that the validation set be large enough to itself provide a statistically significant sample of the instances. One common heuristic is to withhold one-third of the available examples for the validation set, using the other two-thirds for training.

Reduced Error Pruning

How exactly might we use a validation set to prevent overfitting? One approach, called reduced-error pruning (Quinlan 1987), is to consider each of the decision nodes in the tree to be candidates for pruning. Pruning a decision node consists of removing the subtree rooted at that node, making it a leaf node, and assigning it the most common classification of the training examples affiliated with that node. Nodes are removed only if the resulting pruned tree performs no worse than-the original over the validation set. This has the effect that any leaf node added due to coincidental regularities in the training set is likely to be pruned because these same coincidences are unlikely to occur in the validation set. Nodes are pruned iteratively, always choosing the node whose removal most increases the decision tree accuracy over the validation set. Pruning of nodes continues until further pruning is harmful (i.e., decreases accuracy of the tree over the validation set).

Using a separate set of data to guide pruning is an effective approach provided a large amount of data is available. The major drawback of this approach is that when data is limited, withholding part of it for the validation set reduces even further the number of examples available for training. The following section presents an alternative approach to pruning that has been found useful in many practical situations where data is limited. Many additional techniques have been proposed as well, involving partitioning the available data several different times in multiple ways, then averaging the results. Empirical evaluations of alternative tree pruning methods are reported by Mingers (1989b) and by Malerba et al. (1995).

Rule Post-Pruning

In practice, one quite successful method for finding high accuracy hypotheses is a technique we shall call rule post-pruning. A variant of this pruning method is used by C4.5 (Quinlan 1993), which is an outgrowth of the original ID3 algorithm. Rule post-pruning involves the following steps:

1. Infer the decision tree from the training set, growing the tree until the training data is fit as well as possible and allowing overfitting to occur.
2. Convert the learned tree into an equivalent set of rules by creating one rule for each path from the root node to a leaf node.

3. Prune (generalize) each rule by removing any preconditions that result in improving its estimated accuracy.
4. Sort the pruned rules by their estimated accuracy, and consider them in this sequence when classifying subsequent instances.

As noted above, one method to estimate rule accuracy is to use a validation set of examples disjoint from the training set. Another method, used by C4.5, is to evaluate performance based on the training set itself, using a pessimistic estimate to make up for the fact that the training data gives an estimate biased in favor of the rules. More precisely, C4.5 calculates its pessimistic estimate by calculating the rule accuracy over the training examples to which it applies, then calculating the standard deviation in this estimated accuracy assuming a binomial distribution. For a given confidence level, the lower-bound estimate is then taken as the measure of rule performance (e.g., for a 95% confidence interval, rule accuracy is pessimistically estimated by the observed accuracy over the training set, minus 1.96 times the estimated standard deviation). The net effect is that for large data sets, the pessimistic estimate is very close to the observed accuracy (e.g., the standard deviation is very small), whereas it grows further from the observed accuracy as the size of the data set decreases. Although this heuristic method is not statistically valid, it has nevertheless been found useful in practice.

Why convert the decision tree to rules before pruning? There are three main advantages.

- Converting to rules allows distinguishing among the different contexts in which a decision node is used. Because each distinct path through the decision tree node produces a distinct rule, the pruning decision regarding that attribute test can be made differently for each path. In contrast, if the tree itself were pruned, the only two choices would be to remove the decision node completely, or to retain it in its original form.
- Converting to rules removes the distinction between attribute tests that occur near the root of the tree and those that occur near the leaves. Thus, we avoid messy bookkeeping issues such as how to reorganize the tree if the root node is pruned while retaining part of the subtree below this test.
- Converting to rules improves readability. Rules are often easier for people to understand

Incorporating Continuous-Valued Attributes

Our initial definition of ID3 is restricted to attributes that take on a discrete set of values. First, the target attributes whose value is predicted by the learned tree must be discrete valued. Second, the attributes tested in the decision nodes of the tree must also be discrete valued. This second restriction can easily be re- moved so that continuous-valued decision attributes can be incorporated into the learned tree. This can be accomplished by dynamically defining new discrete- valued attributes that partition the continuous attribute value into a discrete set of

intervals. In particular, for an attribute A that is continuous-valued, the algorithm can dynamically create a new Boolean attribute A_c, that is true if A < c and false otherwise. The only question is how to select the best value for the threshold c.

Alternative Measures for Selecting Attributes

There is a natural bias in the information gain measure that favors attributes with many values over those with few values. As an extreme example, consider the attribute Date, which has a very large number of possible values (e.g., March 4, 1979). If we were to add this attribute to any data, it would have the highest information gain of any of the attributes. This is because Date alone perfectly predicts the target attribute over the training data. Thus, it would be selected as the decision attribute for the root node of the tree and lead to a (quite broad) tree of depth one, which perfectly classifies the training data. Of course, this decision tree would fare poorly on subsequent examples, because it is not a useful predictor despite the fact that it perfectly separates the training data.

What is wrong with the attribute Date? Simply put, it has so many possible values that it is bound to separate the training examples into very small subsets. Because of this, it will have a very high information gain relative to the training examples, despite being a very poor predictor of the target function over unseen instances.

One way to avoid this difficulty is to select decision attributes based on some measure other than information gain. One alternative measure that has been used successfully is the gain ratio (Quinlan 1986). The gain ratio measure penalizes attributes such as Date by incorporating a term, called split information, that is sensitive to how broadly and uniformly the attribute splits the data:

$$SplitInformation(S, A) \equiv -\sum_{i=1}^{c} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

where S_1 through S_c, are the c subsets of examples resulting from partitioning S by the c-valued attribute A. Note that SplitInfomation is actually the entropy of S with respect to the values of attribute A. This is in contrast to our previous uses of entropy, in which we considered only the entropy of S with respect to the target attribute whose value is to be predicted by the learned tree.

The Gain Ratio measure is defined in terms of the earlier Gain measure, as well as this SplitInfomation, as follows

$$GainRatio(S, A) \equiv \frac{Gain(S, A)}{SplitInformation(S, A)}$$

Notice that the SplitInfomation term discourages the selection of attributes with many uniformly distributed values. For example, consider a collection of n

examples that are completely separated by attribute A (e.g., Date). In this case, the SplitInfomation value will be $\log_2 n$. In contrast, a boolean attribute B that splits the same n examples exactly in half will have SplitInfomation of 1. If attributes A and B produce the same information gain, then clearly B will score higher according to the Gain Ratio measure.

One practical issue that arises in using GainRatio in place of Gain to select attributes is that the denominator can be zero or very small when $|S_i| \approx |S|$ for one of the Si. This either makes the GainRatio undefined or very large for attributes that happen to have the same value for nearly all members of S. To avoid selecting attributes purely on this basis, we can adopt some heuristic such as first calculating the Gain of each attribute, then applying the GainRatio test only considering those attributes with above average Gain (Quinlan 1986).

An alternative to the GainRatio, designed to directly address the above difficulty, is a distance-based measure introduced by Lopez de Mantaras (1991). This measure is based on defining a distance metric between partitions of the data. Each attribute is evaluated based on the distance between the data partition it creates and the perfect partition (i.e., the partition that perfectly classifies the training data). The attribute whose partition is closest to the perfect partition is chosen. Lopez de Mantaras (1991) defines this distance measure, proves that it is not biased toward attributes with large numbers of values, and reports experimental studies indicating that the predictive accuracy of the induced trees is not significantly different from that obtained with the Gain and Gain Ratio measures. However, this distance measure avoids the practical difficulties associated with the GainRatio measure, and in his experiments, it produces significantly smaller trees in the case of data sets whose attributes have very different numbers of values.

A variety of other selection measures have been proposed as well (e.g., see Breiman et al. 1984; Mingers 1989a; Kearns and Mansour 1996; Dietterich et al. 1996). Mingers (1989a) provides an experimental analysis of the relative effectiveness of several selection measures over a variety of problems. He reports significant differences in the sizes of the unpruned trees produced by the different selection measures. However, in his experimental domains the choice of attribute selection measure appears to have a smaller impact on final accuracy than does the extent and method of post-pruning.

Handling Training Examples with Missing Attribute Values

In certain cases, the available data may be missing values for some attributes. For example, in a medical domain in which we wish to predict patient outcome based on various laboratory tests, it may be that the lab test Blood-Test-Result is available only for a subset of the patients. In such cases, it is common to estimate the missing attribute value based on other examples for which this attribute has a known value. Consider the situation in which Gain (S, A) is to be calculated at

node n in the decision tree to evaluate whether the attribute A is the best attribute to test at this decision node. Suppose that $\langle x, c(x) \rangle$ is one of the training examples in S and that the value A(x) is unknown. One strategy for dealing with the missing attribute value is to assign it the value that is most common among training examples at node n. Alternatively, we might assign it the most common value among examples at node n that have the classification c(x). The elaborated training example using this estimated value for A(x) can then be used directly by the existing decision tree learning algorithm. This strategy is examined by Mingers(1989a).

A second, more complex procedure is to assign a probability to each of the possible values of A rather than simply assigning the most common value to A(x). These probabilities can be estimated again based on the observed frequencies of the various values for A among the examples at node n. For example, given a Boolean attribute A, if node n contains six known examples with A = 1 and four with A = 0, then we would say the probability that A(x) = 1 is 0.6, and the probability that A(x) = 0 is 0.4. A fractional 0.6 of instance x is now distributed down the branch for A = 1, and a fractional 0.4 of x down the other tree branch. These fractional examples are used for the purpose of computing information Gain and can be further subdivided at subsequent branches of the tree if a second missing attribute value must be tested. This same fractioning of examples can also be applied after learning, to classify new instances whose attribute values are unknown. In this case, the classification of the new instance is simply the most probable classification, computed by summing the weights of the instance fragments classified in different ways at the leaf nodes of the tree. This method for handling missing attribute values is used in C4.5 (Quinlan 1993).

Handling Attributes with Differing Costs

In some learning tasks the instance attributes may have associated costs. For example, in learning to classify medical diseases we might describe patients in terms of attributes such as Temperature, Biopsy Result, Pulse, Blood Test Results, etc. These attributes vary significantly in their costs, both in terms of monetary cost and cost to patient comfort. In such tasks, we would prefer decision trees that use low-cost attributes where possible, relying on high-cost attributes only when needed to produce reliable classifications.

ID3 can be modified to take into account attribute costs by introducing a cost term into the attribute selection measure. For example, we might divide the Gain by the cost of the attribute, so that lower-cost attributes would be preferred. While such cost-sensitive measures do not guarantee finding an optimal cost-sensitive decision tree, they do bias the search in favor of low-cost attributes.

Tan and Schlimmer (1990) and Tan (1993) describe one such approach and apply it to a robot perception task in which the robot must learn to classify different objects according to how they can be grasped by the robot's manipulator. In this case the attributes correspond to different sensor readings obtained by a movable sonar on the robot. Attribute cost is measured by the number of seconds required to obtain the attribute value by positioning and operating the sonar. They demonstrate that more efficient recognition strategies are learned, without sacrificing classification accuracy, by replacing the information gain attribute selection measure by the following measure

$$\frac{Gain^2(S, A)}{Cost(A)}$$

Nunez (1988) describes a related approach and its application to learning medical diagnosis rules. Here the attributes are different symptoms and laboratory tests with differing costs. His system uses a somewhat different attribute selection measure

$$\frac{2^{Gain(S,A)} - 1}{(Cost(A) + 1)^w}$$

where w $\in$ [0, 1] is a constant that determines the relative importance of cost versus information gain. Nunez (1991) presents an empirical comparison of these two approaches over a range of tasks.

Problem Statement

The problem statement as was provided:

```
Project 1. Health Insurance Lead Prediction using Python

Your Client FinMan is a financial services company that
provides various financial services like loan, investment
funds, insurance etc. to its customers. FinMan wishes to
cross-sell health insurance to the existing customers who
may or may not hold insurance policies with the company.
The company recommend health insurance to it's customers
based on their profile once these customers land on the
website. Customers might browse the recommended health
insurance policy and consequently fill up a form to apply.
When these customers fill-up the form, their Response
```

towards the policy is considered positive and they are classified as a lead.

Once these leads are acquired, the sales advisors approach them to convert and thus the company can sell proposed health insurance to these leads in a more efficient manner.

Now the company needs your help in building a model to predict whether the person will be interested in their proposed Health plan/policy given the information about:

Demographics (city, age, region etc.) Information regarding holding policies of the customer Recommended Policy Information

General Idea on inconsistencies from the Data Received

As per project job instructions we have three files at hand, two sets of data and a defining information depicting the data.

1. Train data
   - Filename: train_Df64byy.csv;
   - Records: 50882; Fields: 14;
   - Location: https://github.com/WolfDev8675/RepoSJX7/blob/Assign3_1/Data/train_Df64byy.csv
2. Test data
   - Filename: test_YCcRUnU.csv
   - Records: 21805; Fields: 13;
   - Location: https://github.com/WolfDev8675/RepoSJX7/blob/Assign3_1/Data/test_YCcRUnU.csv
3. Outline information on data fields
   - Filename: Metadata.docx
   - Location: https://github.com/WolfDev8675/RepoSJX7/blob/Assign3_1/Data/Metadata.docx

According to observations with the help of spreadsheet software filtering tools for the datasets received we have the following problems.

Train data:

1. "Health indicator" field has 11691 blank or missing data records.

2. "Holding policy duration" and "Holding policy type" fields have 20251 missing or blank records.
3. "Holding policy duration" field has 4335 records with a value '14+' signifying a fact of 'more than 14' by units of time but the '+' character will create an error in runtime due to parsing issues, since it is not a numerical character.

Test data:

1. "Health indicator" field has 5027 blank or missing records.
2. "Holding policy duration" and "Holding policy type" fields have 8603 missing or blank records.
3. "Holding policy duration" field has 1892 records with a value '14+' signifying a fact of 'more than 14' by units of time but the '+' character will create an error in runtime due to parsing issues, since it is not a numerical character.

Additionally (Elephant of the room): The Train data has a field (viz., "Response") available for consideration which is completely missing from the test data.

Workaround to the Data problem

The most significant of the problem faced is the missing field of data producing the inconsistency from the test to train sets. As per suggestions from the guiding faculty the Train data as provided is to be split into a train – test pair with ratio as suggested by conventionally tested methods, and the Test data provided is to be produced up to the client to their requirements and fulfilling the question they had about predicting suggestions regarding the analysis of the sentiment of the potential buyer for a certain policy pitched towards them.

The missing record issue has a more standard solution suggested. For every missing value encountered, the nature of the missing record is to be assessed according to the field in question where it is found and then the following solution is to followed.

1. If the field of data is of categorical nature, then the missing records is to field with the modal value of the non-empty data records.
2. If the field of data is of numerical continuous nature then we need to check from the records, if from the non-empty records holds any outlier points.
    a. If outliers are present then the empty records are filled up with the median value of the non-empty records.
    b. In case of no outliers, the empty records are filled up with the mean value of the non-empty records.

Alternatively, the missing values may be handled via imputation algorithms like the KNN Imputer available from Scikit learning packages in python.

The issue with "Holding policy duration" field is to be handled by randomizing the records with the abnormality. Since we are given a cap of 20 for this field then for all cases of '14+' they are inferred to hold values between 15 to 20, hence, the records are to be randomized between these values.

Description of the data fields

The data fields described as a part of the problem at hand.

Field name	Description
ID	Unique key for each row
City_Code	Code of the city where customer lives
Region_Code	Equivalent to pin code
Accomodation_Type	Type of place to live
Reco_Insurance_Type	Recommended Insurance type
Upper_Age & Lower age	Eligible range of age for this insurance
Is spouse	Whether joint account holder is spouse or not
Health Indicator	Condition of health
Holding_Policy_Duration	If there is an existing policy, then what is the duration
Holding_Policy_Type	Existing policy type
Reco_Policy_Cat	Recommended policy category
Reco_Policy_Premium	Recommended policy premium amount
Response	Class 1 means customer will show interest in buying new policy.

Interpretation

Collecting all the information imparted to us in the project briefing as well as rooting from the little intro to the problem and the problem statement the process of solving can be interpreted into the sequence of steps as follows:

1. Acquire the dataset provided and print or show the data for scrutinization purposes.
2. Carefully studying the data reveals irregularities and blanks, that might and eventually will interfere with the data

3. A first example of an irregularity is that we find that some values aren't numerical at a place where its mandatory to be one.
4. So as a very next vital step is to remove this oddity and fill the place up with a value that is suitable. We are suggested to randomize a value.
5. Next in the process is to fix the datatypes of the fields.
6. To fix the datatypes we used a population strategy for defining whether a field of data is to be categorical, numerical or object (viz., character string).
7. While auto-fixing the datatype, we get some fields that get type-casted to a datatype which it shouldn't be, here we simply fix back those datatypes by hardcoding their nature.
8. Impute the places where the 'Pandas' library while creating the DataFrame object found null or empty values and placed a Nan constant of the 'NumPy' library. This can be accomplished by any of the following ways:
    a. Conventional Median – Mode Impute: the empty values are checked if the field in which they are a part of, is numerical or categorical. For categorized fields the empty places are filled with Mode values inversely for fields where the fields are of numerical nature the empty places are filled with median. This is to be noted that the median or the mode is calculated from the remainder of the data members of the field which are not Nan or empty.
    b. KNN Impute or K – Nearest Neighbours Imputation algorithm: Nearest neighbor (NN) imputation algorithms, efficient methods to fill in missing data where each missing value on some records is replaced by a value obtained from related cases in the whole set of records. Besides the capability to substitute the missing data with plausible values that are as close as possible to the true value, imputation algorithms should preserve the original data structure and avoid to distort the distribution of the imputed variable. For KNN impute to work properly scaling the data is required.
    9. This scaling or standardization is done by a Standard Scaler object from Scikit Learner (SkLearn) package.
    10. If possible, at this stage verify the data cleaning status with the user
    11. Next step is to fulfil the requirement for the predictors and responses. This is handled by a checkbox window which interacts with the user to take in the data.
    12. Plot of the predictor parameters are followed next to the user for both the datasets, providing complete insight to the user about the nature of the data filled in the by the Nan Removing module.
    13. Following this is the encoding the categorical data so that they are understandable by the learner algorithms.
        a. Dummy encoding: Dummy coding provides one way of using categorical predictor variables in various kinds of estimation models. Dummy coding uses only ones and zeros to convey all of the necessary information on group membership.

b. Numerical encoding: Coding the categorical predictor via numerical sequences rather than the binary nature as followed by dummy encoding. Caveat this process does not increase the number of fields as is done in the dummy encoding.

14. In the case of Dummy encoding the predictor list mustn't contain the original names of the fields which are dummy encoded and conversely must contain the dummy encoded fields of those categorical fields which are encoded.
15. After all the encoding is completed, we need to generate the model schematics.
16. Follow up is to feed the data to the model and allow it to train.
17. Post training there needs to be a way to assess the status of the training of the model.
18. At step 17 if we follow Logistic Regression, we find that we face a situation named as "All No Recurrence" where the model over–learns to identify and say "No" or "False" but never learns to say "Yes" or "True" to any option.
19. As a recovery to that found in step 18, we follow Decision Tree algorithm with basic settings which seems to eradicate this problem to some extent.
20. Once the model is trained a module is kept to check if the user wants to retrain the model, drop all trained weightage to the model created and train everything as a new model.
21. If this trained or retrained model is confirmed and finalized by the user, the last trained model is now used to give the results on the secondary data (Test Data) provided to us where there is no response field and is the raw on which the algorithm is to be worked out.
22. Once the results is revealed this is shown to back to the user for scrutinizing and detailing those customers to whom the sales is to pitched, completing the process flow of the code.

Coding

Coding for this total project is divided into three sections for handling the complete job.

Type	Job nature
Documentation files	text files which provide the documentation of the code flow and the requirements
Engines	final engine or checkpoint from where the code starts running.
Operative tasks handlers	does each task separately from cleaning to decision creation

Support function packages	mainly for small assessment of data or structure creators for container windows or UIs, works in tandem with Operative task handlers and engines depending on requirement

Following is the code and progression files.

Documentation Files:

1. Readme.md

```
# RepoSJX7: branch Assign3_1
>__Problem Statement__

Project 1. Health Insurance Lead Prediction using Python

Your Client FinMan is a financial services company that provides various financial
services like loan, investment funds, insurance etc. to its customers. FinMan wishes
to cross-sell health insurance to the existing customers who may or may not hold
insurance policies with the company. The company recommend health insurance to it's
customers based on their profile once these customers land on the website. Customers
might browse the recommended health insurance policy and consequently fill up a form
to apply. When these customers fill-up the form, their Response towards the policy is
considered positive and they are classified as a lead.

Once these leads are acquired, the sales advisors approach them to convert and thus
the company can sell proposed health insurance to these leads in a more efficient
manner.

Now the company needs your help in building a model to predict whether the person will
be interested in their proposed Health plan/policy given the information about:

Demographics (city, age, region etc.)
Information regarding holding policies of the customer
Recommended Policy Information

Sample Submission

This file contains the exact submission format for the predictions. Please submit CSV
file only.

| Variable | Definition |
| ID | Unique Identifier for a row |
| Response | (Target) Probability of Customer showing interest (class 1) |

---
>__Startup point__:
>  _FinalEngine.py_
---
>__Requirements__: The code for operating properly requires the following packages, it
is recomended to check their presence and update definitions if needed.
>
> Packages required
> - io
> - matplotlib
> - numpy
> - pandas
> - pandastable
> - random
> - scikit-learn
> - sklearn
> - time
> - tkinter
>
```

```
> Out of these packages python comes shipped with _**tkinter**_, _**io**_, _**time**_
and _**random**_ packages.
> If any issues is faced with packages it is suggested to
> run command _**`python -m pip install -r requirements.txt`**_ or _**`py -m pip
install -r requirements.txt`**_ in the Developer Command prompt,
> the
[requirements.txt](https://github.com/WolfDev8675/RepoSJX7/blob/Assign3_1/requirements
.txt) will install all packages that were live in the python library when this coding
was being done.
>
> *Refer to the
__[TaskMap.txt](https://github.com/WolfDev8675/RepoSJX7/blob/Assign3_1/TaskMap.txt)__
file for specific personalized changes in the running schedule of
__[FinalEngine.py](https://github.com/WolfDev8675/RepoSJX7/blob/Assign3_1/FinalEngine.
py)__ file*

---
&copy;[Bishal Biswas(@WolfDev8675)](https://github.com/WolfDev8675)
_(b.biswas_94587@ieee.org)_
```

2. requirements.txt

```
absl-py==0.12.0
astunparse==1.6.3
cached-property==1.5.2
cachetools==4.2.1
certifi==2020.12.5
chardet==4.0.0
cycler==0.10.0
et-xmlfile==1.0.1
flatbuffers==1.12
future==0.18.2
gast==0.3.3
google-auth==1.27.1
google-auth-oauthlib==0.4.3
google-pasta==0.2.0
grpcio==1.32.0
h5py==2.10.0
idna==2.10
importlib-metadata==3.7.2
joblib==1.0.1
Keras==2.4.3
Keras-Preprocessing==1.1.2
kiwisolver==1.3.1
lxml==4.6.2
Markdown==3.3.4
matplotlib==3.3.4
numexpr==2.7.3
numpy==1.19.5
oauthlib==3.1.0
openpyxl==3.0.7
opt-einsum==3.3.0
pandas==1.2.3
pandastable==0.12.2.post1
Pillow==8.1.2
protobuf==3.15.6
pyasn1==0.4.8
pyasn1-modules==0.2.8
pyparsing==2.4.7
python-dateutil==2.8.1
pytz==2021.1
PyYAML==5.4.1
requests==2.25.1
requests-oauthlib==1.3.0
rsa==4.7.2
```

```
scikit-learn==0.24.1
scipy==1.6.1
seaborn==0.11.1
six==1.15.0
sklearn==0.0
Tcl==0.2
tensorboard==2.4.1
tensorboard-plugin-wit==1.8.0
tensorflow==2.4.1
tensorflow-estimator==2.4.0
termcolor==1.1.0
threadpoolctl==2.1.0
tk==0.1.0
typing-extensions==3.7.4.3
urllib3==1.26.3
Werkzeug==1.0.1
wrapt==1.12.1
xlrd==2.0.1
zipp==3.4.1
```

3. TaskMap.txt

```
# file: TaskMap.txt

This is a task map of all the tasks in flow as followed in the file -FinalEngine.py

1. Acquiring the data ( followed on both datasets ) :            ** lines~  20-30
2. Removing irregularities in the data ['14+' in numerical field ] ( followed on
both datasets ):        ** lines~  32-36
3. Type fixing the dataset ( followed on both datasets ):        **lines~  38-42
4. Revert fixing [overriding the typefix done by algorithm] ( followed on both
datasets ):             **lines~  44-48
5. NAN removals [Conventional Median-Mode method or KNN(Standard Scaled) ] (
followed on both datasets ):          **lines~  50-58
6. Post Cleaning data review ( followed on both datasets ):             **lines~
60-64
7. Set predict and infer fieldnames ( local for reference to both datasets pulled
from one dataset ):            **lines~  66-69
8. Plots of the data ( followed on both datasets ):       **lines~  71-75
9. Dummy encoding of the categorical predicatables ( followed on both datasets ):
         **lines~  77-81
10. Reset Predict [removing original field names from predictors which are dummy
encoded] ( local set for reference on both datasets ):           **lines~  83-87
11. Numerical Encoding Categorical Predictables ( followed on both datasets ):
   **lines~  89-93
12. Learner instance create and dataset push( followed on a primary dataset ):
   **lines~  95-101
13. Train Model and Provide Metrics ( followed on the dataset used to train
Logistic model ):       **lines~  103-108
14. Retrain options ( followed on the dataset used to train Logistic model ):
   **lines~  110-121
15. Predict Result ( followed on the dataset for result functionality ):
   **lines~  123-127


**For removal of NAN values by KNN method suitable lines need to be uncommented in
section 5 refer lines
**For following LogisticRegression path section 9 & 10 needs to be activated
   and section 11 to be deactivated also changes need to be made in section 12 in
model creation refer lines
**For following DecisionTree path section 9 & 10 needs to be deactivated
```

and section 11 to be activated also changes need to be made in section 12 in model creation refer lines

Engines

FinalEngine.py

```
#!usr/bin/python

# Code file: FinalEngine.py
# Task:
##Final code to execute all processes sequentially in a manner as
## required to complete the task  properly

# imports
import pandas as PD
import analytics as ALS
import Cosmetics as CS
import DataManager as FAS
import Cleaners as CLNS
import Interpretors as ITRS
import random as RDS
from time import process_time as PRT

# Start of Code

## Acquire Data and print primary details + show data
print(" Acquiring Data ..... ",);cT=PRT()
#1
GenDFrame=PD.read_csv(CS.uiGetFile())
CS.showTable(GenDFrame,"Primary DataFrame")
CS.showInformation("Primary DataFrame Information
",FAS.dataInfo(GenDFrame)+'\n'+GenDFrame.describe().to_string());
#2
ResDFrame=PD.read_csv(CS.uiGetFile())
CS.showTable(ResDFrame,"Result DataFrame")
CS.showInformation("Result DataFrame Information
",FAS.dataInfo(ResDFrame)+'\n'+ResDFrame.describe().to_string());
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Remove irregularities ( 14+ in 'Holding_Policy_Duration' field)
print(" Oddities Clean .... ",)
CLNS.find_Replace(GenDFrame,'Holding_Policy_Duration','14+',RDS.randint(15,20))   #1
CLNS.find_Replace(ResDFrame,'Holding_Policy_Duration','14+',RDS.randint(15,20))   #2
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Fix datatypes of fields
print(" Datatype Fixing .... ",)
GenDFrame=FAS.FixDataByPopulation(GenDFrame)    #1
ResDFrame=FAS.FixDataByPopulation(ResDFrame)    #2
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Fixing Specific field datatype shift *** field: Holding_Policy_Duration due to less
variations gets categorized
print(" Fixing Specific field datatype shift ... ")
GenDFrame=ALS.reset_columnData(GenDFrame,{'Holding_Policy_Duration':'numerical','Upper
_Age':'numerical','Lower_Age':'numerical','Holding_Policy_Type':'category','Region_Cod
e':'category'})
ResDFrame=ALS.reset_columnData(ResDFrame,{'Holding_Policy_Duration':'numerical','Upper
_Age':'numerical','Lower_Age':'numerical','Holding_Policy_Type':'category','Region_Cod
e':'category'})
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Remove NAN values  ** Comment/Uncomment required section
print(" NaN Clean .... ",)
print(" Conventional Median/Mode filler method ..... ",)
```

```
GenDFrame=CLNS.removeNAN(GenDFrame)     #1
ResDFrame=CLNS.removeNAN(ResDFrame)     #2
#print(" K Nearest Neighbour Imputation method ..... ",)
#GenDFrame=CLNS.imputeKNN(FAS.scaleData(GenDFrame))     #1
#ResDFrame=CLNS.imputeKNN(FAS.scaleData(ResDFrame))     #2
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Reprint details : post cleaning
print("Post Clean Data Review... ")
CS.showInformation("Primary DataFrame Information
",FAS.dataInfo(GenDFrame)+'\n'+GenDFrame.describe().to_string())      #1
CS.showInformation("Result DataFrame Information
",FAS.dataInfo(ResDFrame)+'\n'+ResDFrame.describe().to_string())       #2
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Setting the predict and infer field headers
print(" Parameter Set .... ",)
[predict,infer]=CS.selectFromLists(" Predictables and Inferences  "," Select
predictables  ",GenDFrame.columns.to_list()," Select inferences
",GenDFrame.columns.to_list())
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Plots of fields ** histogram or boxplot depending on failure criteria
print(" Generating Plots.... ")
FAS.showPlots(GenDFrame,predict,'Plots: Primary Data')
FAS.showPlots(ResDFrame,predict,'Plots: Secondary Data')
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

### Dummy Encoding the predicatables   ...*** Suitable for Logistic Regression
#print(" Dummy Encoding suitable predicatables....   ")
#[GenDFrame,DumbRems_G,DumbAppends_G]=FAS.encodeDummy(GenDFrame,predict)       #1
#[ResDFrame,DumbRems_R,DumbAppends_R]=FAS.encodeDummy(ResDFrame,predict)       #2
#print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

### Reset Predict field names ~ in effect to the dummy encoding  ...*** Suitable for
Logistic Regression
#print(" Resetting Predict field headers ")
#[predict.remove(col) for col in DumbRems_G if col in predict]
#predict.extend(DumbAppends_G)
#print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Numerical Encoding of the Categorical Predicatables ...*** Suitable for Decision
Tree Classifiers
print(" Numerical Encoding the Categorical Predictables ... ")
GenDFrame=FAS.encodeImpose(GenDFrame,predict)       #1
ResDFrame=FAS.encodeImpose(ResDFrame,predict)       #2
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Initiate Machine Learning Model  ** uncomment specific lines as required
print(" Initiating Learner Model ... ",)
#mods=ITRS.Learner(model=ITRS.LogisticRegression(solver='lbfgs',max_iter=100000))
#** Logistic Regression Model
mods=ITRS.Learner(model=ITRS.DecisionTreeClassifier())      #** Decision Tree
Classifier Model
mods.SplitRatio=(7.0/3)
mods.pushData(GenDFrame,predict,infer)
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Train Logistic model
print(" Training Model ... ",)
mods.generateModel()
mods.generateMetricsPlots()
CS.showInformation(" Model Metrics ",mods.generateMetricsInfo())
print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()

## Retrain options
retrain=True;attempt=0
print(" Retrain Model... ")
```

```
while(retrain):
    retrain=CS.decisionMessage(" Retrain Options "," Retrain Model ? ....")
    if retrain:
        mods.retrainModel()
        CS.showInformation(" Model Metrics ",mods.generateMetricsInfo())
        attempt+=1
        print(" Attempt = ",attempt)
    print(" Success. ... Process time (s) ",(PRT()-cT));cT=PRT()
print(" Retrain attempts = ",attempt)

## Applying Trained Model to Unknown data
print(" Applying Model to Unknown Data ")
results=mods.predictionApply(ResDFrame[predict]) #numpy.ndarray
CS.showTable(FAS.matchResponse(ResDFrame,results,'Response'),"Required Result")
print(" Success. ... Process time (s) ",(PRT()-cT))

#... End of codes
# END OF FILE: 'FinalEngine.py'
```

Operative tasks handlers

1. Cleaners.py

```
#!usr/bin/python

#Code file: Cleaners.py
#Task:
#**  Clean dataFrame depending on the type of
#**  operation chosen for the task.
#**  Cleaning can be either the Convensional type
#**  or the Imputation algorithm type

# imports
import numpy as NP
import pandas as PD
import analytics as ALS
from sklearn.impute import KNNImputer as KNI
from sklearn.preprocessing import StandardScaler as SSLC



def find_Replace(dFrame=None,columnName=None,find=None,replace=None):
    """ Function find_Replace:
        Operation: find specific value from a particular column
        in a dataframe and replace the found value with the
        replcement value
        NB: this function is not effective against numpy.nan, numpy.NAN and numpy.NaN
        since np.nan==np.nan is False which is same for other variants supplied by
numpy
        as a result fails to find the respective data """
    positions=dFrame[dFrame[columnName]==find].index.tolist() # index list of changes
    for idx in positions: dFrame.loc[idx,columnName] = replace
  #end of function


def removeNAN(dFrame=None):
    """ Function removeNAN:
        Operation: Remove NAN values or values in a pandas.DataFrame
        of the type of numpy.NAN(IEEE 754 floating point representation of Not a
Number (NaN))
        via conventional methods viz.,
            1.      If the field of data is of categorical nature, then the missing
records is to
                field with the modal value of the non-empty data records.
```

```
            2.      If the field of data is of numerical continuous nature then we
can either fill with Median or Mean,
                since Mean shifts more with data tendency shift than Median, Median is
chosen to fill.
            """
    null_info= dict(dFrame.isnull().sum())
    nullcols=[]
    for cols in null_info:
        if null_info[cols] > 0:nullcols.append(cols)
    for a_col in nullcols:
        if type(dFrame[a_col].dtype) is PD.CategoricalDtype:
            positions=dFrame[dFrame[a_col].isnull()].index.tolist() # index list of
changes
            for idx in positions: dFrame.loc[idx,a_col] =
dFrame[a_col].mode().tolist()[0] #mode fill
        elif type(dFrame[a_col].dtype) is NP.dtype:
            positions=dFrame[dFrame[a_col].isnull()].index.tolist() # index list of
changes
            for idx in positions: dFrame.loc[idx,a_col] = dFrame[a_col].median()
#median fill
        else: pass   # no NULLs or NANs or NONEs ... code not supposed to execute
(exclusive case)
    return dFrame
  #end of function


def imputeKNN(dFrame=None):
    """ Function imputeKNNN:
        Operation: Using K-Nearest Neighbour algorithm to impute and fill
        the empty places or data locations that is missing (viz., holds a numpy.NaN
constant)
        Function returns back data in encoded manner viz, coded functionality for
categorized
        values is not reverted back, this is also kept in this way keeping in mind
that any other
        learner algorithms irrespective of nature recognizes the numerical encoded
values than
        string encoded numpy.object values """

    null_info= dict(dFrame.isnull().sum()) # information on the nulls
    # Generating the Imputer Logic Object
    knnIR=KNI(n_neighbors=5,weights='uniform',metric='nan_euclidean')
    stdSLR=SSLC()
    nan_cols=[]   # empty list to collect NANs or NULLs containing column names
    for cols in null_info:
        if(null_info[cols]):
            nan_cols.append(cols)  # appending column names
    nonIntNULLS=[]
    for cols in nan_cols:
        if(ALS.numericStrength(dFrame[cols])<50 and
ALS.variety(dFrame[cols],95)[2]=='categorized'): nonIntNULLS.append(cols) # only
categorised can be in this error section
        elif(ALS.numericStrength(dFrame[cols])>50):pass
        else: nan_cols.remove(cols)    # supposed to be object(string miscellaneous)
type hence not to be considered

    mapsOfChanges=dict.fromkeys(nonIntNULLS,{})
    for a_col in nonIntNULLS:
        # Hex-Step Process
        #1: Pre Conversion Tag Listing
        pr_ctl=ALS.variety(dFrame[a_col],95)[0]
        #2: Conversion
        dFrame[a_col]=dFrame[a_col].cat.codes
        dFrame=ALS.reset_columnData(dFrame,{a_col:'category'})
        #3: Post Conversion Tag Listing
        po_ctl=ALS.variety(dFrame[a_col],95)[0]
        #4: Map index by population to Change register
        for key in po_ctl:
```

```
            contents=list(pr_ctl.items())
            for i in range(len(contents)):
                if(po_ctl[key]==contents[i][1]):
                    mapsOfChanges[a_col][key]=contents[i][0]
        #5: Return Nulls
        ret_idxs=dFrame[dFrame[a_col] == -1 ].index.tolist() #since catgorical coding
changed NULLS to (-1)
        for idx in ret_idxs: dFrame.loc[idx,a_col]= NP.nan
        #6: Scale fields


    knnIR.fit(dFrame[nan_cols])  #fitting
    dFrame[nan_cols]=knnIR.transform(dFrame[nan_cols]) # finalising imputation task
    # ..End of imputation

    # Returning Mapping (Only for Categorical values)
    for keys in mapsOfChanges:
        dFrame[keys]=round(dFrame[keys]) # correcting non categorized Decimals
        dFrame[keys].map(mapsOfChanges[keys])
        dFrame=ALS.reset_columnData(dFrame,{keys:'category'})

    return dFrame #Return Changes
  #end of function



#**End of codes
#END OF FILE 'Cleaners.py'
```

2. DataManager.py

```
#!usr/bin/python

#Code file: DataManager.py
#Task:
#**  information and assimilation of the
#**  data content of a pandas.DataFrame object

#imports
import pandas as PD
import numpy as NP
import io
import analytics as ALS
import matplotlib.pyplot as PLOT
from sklearn.preprocessing import *

def FixDataByPopulation(dFrame):
    """ Function FixDataByPopulation:
        Operation: assess the population strength and identify
         if a certain field is categorizable or not and depending on that
         try to fix the pandas.DataFrame object to specific datatypes respective of
fields """

    typer0={}  # empty dictionaries for recognising data nature
    typer1={}
    #establishing character of dataset
    for col in dFrame.columns:
        typer0[col]=[ALS.variety(dFrame[col],95)[2],ALS.numericStrength(dFrame[col])]
    # establishing fulfilment criteria of dataset
    for element in typer0:
        if typer0[element][0].startswith('random'):
            if typer0[element][1]>50: typer1[element]='numerical'
            else: typer1[element]='object'
        elif typer0[element][0].startswith('categ'):
            typer1[element]='category'
```

```
    return ALS.reset_columnData(dFrame,typer1)
  #end of function

def matchResponse(dFrame,response,responseName):
    """Function  matchResponse
       Operation: match a response input to the
       corresponding dataframe using the key or
       hashable column of the dataframe """
    dFrameResp=PD.Series(response)
    keyName=ALS.detectKeys(dFrame)[0]
    if keyName is 'index': dFrameKey=dFrame.index
    else: dFrameKey=dFrame[keyName]
    return PD.DataFrame({keyName:dFrameKey,responseName:dFrameResp})
 #end of function

def dataInfo(dFrame):
    """ Function dataInfo:
        Operation:  push the results of pandas.DataFrame.info from
        the sys.stdout buffer to a local string buffer to be used as a string function
"""
    buffer=io.StringIO()
    dFrame.info(buf=buffer)
    return buffer.getvalue()
  #end of function

def showPlots(dFrame=None,fieldNames=None,winTitle='Plot Figure'):
    """ Function  showPlots:
        Operation: plot Boxplots or Histograms of the data fields
         Histograms are plotted for Categorical fields, while,
         Boxplots are plotted for Numerical fields"""

    fieldNames_refined=fieldNames
    for field in fieldNames:
        if dFrame[field].dtype == NP.object : fieldNames_refined.remove(field)
    nosFields=len(fieldNames_refined)
    TmpNF=(nosFields-1) if (nosFields%3 and nosFields%4 and nosFields%5) else
nosFields
    axC=5 if not TmpNF%5 else 4 if not TmpNF%4 else 3 if not TmpNF%3 else None
    axR=int(TmpNF/axC); axR= axR if TmpNF is nosFields else axR+1
    pos=1
    PLOT.figure(figsize=(13,7.5)).canvas.set_window_title(winTitle)
    for col in fieldNames:
        PLOT.subplot(axR,axC,pos)
        if dFrame[col].dtype == NP.int64 or dFrame[col].dtype == NP.float64:
            dFrame.boxplot(column=col)
        elif type(dFrame[col].dtype) == PD.CategoricalDtype:
            dFrame[col].hist()
        else: pass # non executable section
        pos+=1
    PLOT.show()
  #end of function

def encodeImpose(dFrame=None,fieldNames=None):
    """ Function encodeImpose:
        Operation: Encode categorical datatypes to number coded categories """
    for field in fieldNames:
        if type(dFrame[field].dtype) == PD.CategoricalDtype:
            dFrame[field]=PD.Categorical(values=dFrame[field].cat.codes)
        else: pass
    return dFrame
  #end of function

def encodeDummy(dFrame=None,fieldNames=None):
    """ Function encodeDummy:
        Operation: Encode Categorical datatypes to dummy variables
        and add them to dataframe """
    RetFrame=dFrame
```

```
    EncodedCols=[]
    DumbColumns=[]
    for field in fieldNames:
        if type(dFrame[field].dtype) == PD.CategoricalDtype:
            oneDumbColumn=PD.get_dummies(dFrame[field])
            EncodedCols.append(field)
            #./ Fixing Dummies to Original
            DumbColumns.extend(oneDumbColumn.columns.to_list())
            RetFrame=PD.concat([RetFrame,oneDumbColumn],axis=1)
    return [RetFrame,EncodedCols,DumbColumns]
   #end of function


def scaleData(dFrame=None):
    """ Function scaleData:
        Operation: Use a standard scaler on the numerical fields of data
        and thereby standardise the data given
    """
    scaler=StandardScaler()
    allCols=dFrame.columns.to_list();modified=dFrame
    numerics=[]; [numerics.append(col) for col in allCols if (dFrame[col].dtype ==
NP.int64 or dFrame[col].dtype == NP.float64)]
    scaled=PD.DataFrame(data=scaler.fit_transform(dFrame[numerics]),columns=numerics)
    modified[numerics]=scaled
    return modified



  #** end of code
#END OF FILE 'DataManager.py'
```

3. Interpretors.py

```
#!usr/bin/python

#Code file: Interpretors.py
#Task:
#**  Interpret the solution to the data

#imports
import io
import pandas as PD
import numpy as NP
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import plot_confusion_matrix, plot_roc_curve
from sklearn.metrics import
confusion_matrix,accuracy_score,precision_score,recall_score,f1_score,roc_auc_score
import matplotlib.pyplot as PLOT

#start of codes
class Learner():
    """ class Learner  """

    Model=None
    predictors=None
    inferences=None
    SplitRatio=None
    StudyData=None
    confMatrix=None
    TrainX=None;TestX=None;TrainY=None;TestY=None;PredictY=None
    Accuracy=None;
    Precision=None
    Recall=None
```

```
    F_Score=None
    Specificity=None
    TN=None;TP=None;FN=None;FP=None;
    ROC_AUC_Score=None


    def __init__(self,model,**kwargs):
        """ Class Learner
            Operation: create Machine Learning Classifier Object """

        self.Model=model
        for var in kwargs:
            if hasattr(self.Model,var):
                setattr(self.Model,var,kwargs[var])
      #end of iniitalizer function

    def pushData(self,dFrame=None,XHeads=None,YHeads=None):
        """ Function pushData:
            Parent Class: Logistic
            Operation: push the required data to the classifier model """
        self.StudyData=dFrame
        self.predictors=XHeads
        self.inferences=YHeads
     #end of function

    def generateModel(self):
        """ Function generateModel
            Parent Class: Logistic
            Operation: Generate the respective Logistic Regression classifier model
            from the data provided to the classifier object itself """
        R=self.SplitRatio
        testfactor=abs(((1/R)-1)/((1/R)-R))
        X=self.StudyData[self.predictors]
        Y=self.StudyData[self.inferences]

self.TrainX,self.TestX,self.TrainY,self.TestY=train_test_split(X,Y,test_size=testfacto
r,random_state=0)
        self.Model.fit(self.TrainX,self.TrainY)
      #end of function


    def generateMetricsInfo(self):
        """ Function generateMetrics
            Parent Class: Logistic
            Operation: Generate Metrics informations of the model  """
        buffer=io.StringIO()
        self.PredictY=self.Model.predict(self.TestX)
        self.confMatrix=confusion_matrix(self.TestY,self.PredictY)
        self.Accuracy=accuracy_score(self.TestY,self.PredictY)*100
        self.Precision=precision_score(self.TestY,self.PredictY)*100
        self.Recall=recall_score(self.TestY,self.PredictY)*100
        self.F_Score=f1_score(self.TestY,self.PredictY)*100
        self.TN,self.FP,self.FN,self.TP=self.confMatrix.ravel()
        self.Specificity=(self.TN*100)/(self.FP+self.TN)
        self.ROC_AUC_Score=roc_auc_score(self.TestY,self.PredictY)
        buffer.write(" METRICS DETAILS \n")
        buffer.write(" True Positive Count : "+str(self.TP)+"\n\n")
        buffer.write(" True Negative Count : "+str(self.TN)+"\n\n")
        buffer.write(" False Positive Count : "+str(self.FP)+"\n\n")
        buffer.write(" False Negative Count : "+str(self.FN)+"\n\n\n\n")
        buffer.write(" Accuracy : "+str(self.Accuracy)+"\n\n")
        buffer.write(" Precision : "+str(self.Precision)+"\n\n")
        buffer.write(" Recall : "+str(self.Recall)+"\n\n")
        buffer.write(" Specificity : "+str(self.Specificity)+"\n\n")
        buffer.write(" F - Score : "+str(self.F_Score)+"\n\n")
        buffer.write(" ROC AUC Score : "+str(self.ROC_AUC_Score)+"\n\n")
        return buffer.getvalue()
     #end of function
```

```
    def generateMetricsPlots(self):
        """ Function generateMetrics
            Parent Class: Logistic
            Operation: Generate Metrics informations of the model  """
        figure,(AXPCM,AXPROC)=PLOT.subplots(1,2,figsize=(13,7.5))
        figure.canvas.set_window_title(" Model Metrics  ")
        plot_confusion_matrix(self.Model,self.TestX,self.TestY,ax=AXPCM)
        AXPCM.set_title(" Confusion Matrix ")
        plot_roc_curve(self.Model,self.TestX,self.TestY,ax=AXPROC)
        AXPROC.set_title(" Receiver Operating Charactistics (ROC) ")
        PLOT.show()
      #end of function

    def predictionApply(self,data):
        """ Function predictionApply
            Parent Class: Logistic
            Operation:  Apply the trained Model on a different data
            than on which it is trained or tested """
        return self.Model.predict(data)
       #end of function

    def retrainModel(self):
        """ Function retrain
            Parent Class: Logistic
            Operation: Retrain the previously generated model """
        self.generateModel()
        self.generateMetricsPlots()
    #end of function

 #end of class: Logistic

 #END OF FILE 'Interpretors.py'
```

Support function packages

1. analytics.py

```
#!usr/bin/python
""" Analytics packaging to support with functions that help in
analysing distribution nature of data in a pandas.Dataframe object
 functions present : numericStrength, variety, reset_columnData
 """
# code: BishalBiswas(https://github.com/WolfDev8675)

#imports
import pandas as PD

# function definitions
def numericStrength(dSeries):
    """ Function numericStrength:
    operative: numericStrength(data_column as pandas.core.series.Series)
    returns the percentage of the data contained in the population is numeric
    pointing in operator to take decision based on the majority value to set datatype
"""
    #...
    # initials
    dSeries.to_list()
    numCounts=0
    elseCounts=0
    # finding numeric hits
    for element in dSeries:
        if(type(element)==str):
            if element.isnumeric(): numCounts+=1
            else: elseCounts+=1
```

```
        else:
            if (type(element)==int or type(element)==float): numCounts+=1
            else: elseCounts+=1
    # end of for loop
    #...
    return (numCounts*100.0)/len(dSeries)  #final return
# end of function

def variety(dSeries,cutoff):
    """ Function variety:
    operative: variety(data_column as pandas.core.series.Series,cutoff value to
determine number of classes(percentage))
    returns the percentage of the variations contained in the data
    pointing in operator to take decision based on the types to set datatype
    complete return type: list of mixed datatypes
    list[0] = dictionary of the percentage of occurances with classes
    list[1] = number of variation classes
    list[2] = classification of nature based on cutoff value ( categorized or
randomized )
    classification calculation:
     percentage of difference between the data length and the number of categories
generated
     if found greater than cutoff, then it is inferred as categorised
     else assumed as randomized distribution
    ...
    """
    #...
    #intitals
    dSeries.to_list()
    variations={}
    total_length=len(dSeries)
    ret_ls=[]
    cf_dec=(cutoff*1.0)/100
    #
    for element in dSeries:
        if element not in variations: variations[element]=1
        else: variations[element]+=1
    # size class
    for element in variations:
        variations[element]=(variations[element]*100.0)/total_length
    #finalizing results
    ret_ls.append(variations)
    ret_ls.append(len(variations))
    # defining classification
    per_diff=(total_length-len(variations))*1.0/len(dSeries)
    if (per_diff>cf_dec): ret_ls.append('categorized')
    else: ret_ls.append('randomized')

    return ret_ls  # final return
# end of function

def reset_columnData(dFrame=None,typing=None):
    """ Function reset_columnData(Pandas.Dataframe,{Column_name:Datatype_to_finalize})
    returns dataframe with datatypes changed and fixed according to typing information
for the columns
    dFrame: defaults=None; possible values Pandas.Dataframe object
    typing: defaults=None; possible values 'category', 'numerical', 'object'

    returns: None; if any or both arguments are 'None';
    or pandas.Dataframe object with the changes made to the respective datatypes of
columns
    """
    # empty handler
    if (dFrame is None or typing is None):
        return None
    # inititalizing setup
    WorkerDFrame=dFrame
    # handling errors
```

```
    if type(dFrame) is not PD.DataFrame:
        raise ValueError(" dFrame must be a Pandas.DataFrame object")
    if type(typing) is not dict:
        raise ValueError(" Typing data must be a Dictionary ")
    # registering changes
    for cols in typing:
        if (typing[cols] == 'numerical'):
            WorkerDFrame[cols]=PD.to_numeric(WorkerDFrame[cols],errors='coerce')
        elif (typing[cols] == 'category'):
            WorkerDFrame=WorkerDFrame.astype({cols:typing[cols]})
        elif (typing[cols] == 'object'):
            WorkerDFrame=WorkerDFrame.astype({cols:typing[cols]})
        else:
            pass
    # end of loop
    return WorkerDFrame #return changes
# end of function

def detectKeys(dFrame=None):
    """ Function detectPrimary:
        operative: detectKeys(Pandas.DataFrame)
        returns: column name/s with all unique non repeating characters/numbers {type:
List(python native list datatype)}
        The function is to detect the column/s which can be used as the Unique key
        to the dataframe
        Methodology: Any column, if it has all values unique to itself,
        viz., different values for each rows/indexes provided that
        none of the values have repeated themselves. In all, every value of these
columns
        could be used to uniquely identify an entry or a row in a dataframe """

    # initializing primitives
    allColumns=dFrame.columns.to_list()
    keyable=[]
    # searching Keyable columns
    for a_column in allColumns:
        tempColl=PD.Series(dFrame[a_column])
        if tempColl.to_list() == list(set(tempColl)):
            keyable.append(a_column)
    if not len(keyable): keyable=['index']
    return keyable
# end of function

#** End of codes



#END OF FILE 'analytics.py'
```

2. Cosmetics.py

```
#!usr/bin/python
""" Cosmetics packaging to help with locating a file or a folder,
showing data tables, important information, or taking decisions,
in a GUI manner using the Tkinter module of python native
"""
# code: BishalBiswas(https://github.com/WolfDev8675)

#imports
from tkinter import filedialog,messagebox
from tkinter import *
import pandas as PD
import pandastable as PTS

def uiGetFile(paths=None):
```

```
    """ Function uiGetFile:
        Operation: receive file path from user using GUI functionality"""
    root=Tk()
    fBox=filedialog
    root.wm_title(" Get File: Native")
    root.geometry("500x80")
    files=fBox.askopenfile()
    root.destroy()
    return files.name
 #end of function


def showTable(dataFrame,windowName):
    """ Function showTable:
        Operation: use GUI to preview a dataframe in the form of a table """
    root=Tk()
    root.wm_title(windowName)
    root.geometry("1200x600")
    fR=Frame(root)
    fR.pack(fill='both',expand=True)
    tb=PTS.Table(fR,dataframe=dataFrame)
    tb.show()
    root.mainloop()
  #end of function


def showInformation(windowName,binderTexts):
    """ Function  showInformation:
        Operation: use GUI to show information in the form of text """
    root=Tk()
    root.wm_title(windowName)
    root.geometry("1200x600")
    tex=Text(root,width=500,height=1100)
    tex.pack()
    tex.insert(END,binderTexts)
    root.mainloop()
  #end of function

def decisionMessage(windowName,message):
    """ Function decisionMessage:
        Operation: show a message to the screen and ask to select either
        one of two options, by default command2 is selected """
    decision=False
    root=Tk()
    MsgBox=messagebox
    root.wm_title("Native Decision ")
    root.geometry("500x80")
    mbox=MsgBox.askquestion(windowName,message,icon='warning')
    root.destroy()
    if mbox=='yes': decision=True
    return decision
 #end of function

def selectFromLists(windowName,label1,listVAR1,label2=None,listVAR2=None):
    """ Function selectFromLists:
        Operation: select options from a list available ...
        section 1 of this function fills window with checkboxes
        section 2 of this function fills window with radiobuttons
        ...
    """
    root=Tk()
    root.wm_title(windowName)
    root.geometry("1200x600")
    returns1=[];returns2=None;
    dctVAR={}
    dctVAR2=dict.fromkeys(listVAR1)
    if label2 is not None: dctVAR3=dict.fromkeys(listVAR2)
    for item in listVAR1:
```

```
        dctVAR[item]=IntVar()
    r=0
    c=0
    lb1=Label(root,text=label1).grid(row=r,columnspan=6,ipady=6);r+=1
    for item in listVAR1:

dctVAR2[item]=Checkbutton(root,text=item,variable=dctVAR[item]).grid(row=r,column=c,ip
adx=2)
        if c<10: c+=1
        else: c=0;r+=1
    r+=2
    if label2 is not None:
        c=0
        lb2=Label(root,text=label2).grid(row=r,columnspan=6,ipady=6);r+=1
        response=StringVar()
        for item in listVAR2:

dctVAR3[item]=Radiobutton(root,text=item,variable=response,value=item).grid(row=r,colu
mn=c,ipadx=2)
            if c<10: c+=1
            else: c=0;r+=1

    b=Button(root,text='Confirm',command=root.destroy).grid(sticky=SW)
    root.mainloop()
    for item in dctVAR:
        dctVAR[item]=dctVAR[item].get()
        if dctVAR[item]: returns1.append(item)
    returns2=response.get()
    return [returns1,returns2]


 #** End of codes

#END OF FILE 'Cosmetics.py'
```

Window progressions

1

Primary DataFrame

	ID	City_Cod	Region_C	Accomoda	Reco_Insurance	Upper_Ag	Lower_Ag	Is_Spous	Health Indi	Holding_P	Holding_P	Reco_Poli	Reco_Poli	Response
1	1	C3	3213	Rented	Individual	36	36	No	X1	14+	3.00	22	11628.00	0
2	2	C5	1117	Owned	Joint	75	22	No	X2			22	30510.00	0
3	3	C5	3732	Owned	Individual	32	32	No		1.0	1.00	19	7450.00	1
4	4	C24	4378	Owned	Joint	52	48	No	X1	14+	3.00	19	17780.00	0
5	5	C8	2190	Rented	Individual	44	44	No	X2	3.0	1.00	16	10404.00	0
6	6	C9	1785	Rented	Individual	52	52	No	X2	5.0	1.00	22	15264.00	1
7	7	C3	679	Owned	Individual	28	28	No				17	10640.00	0
8	8	C1	3175	Owned	Joint	75	73	Yes	X4	9.0	4.00	17	29344.00	1
9	9	C15	3497	Owned	Joint	52	43	No	X1	14.0	3.00	1	27283.20	0
10	10	C1	530	Owned	Joint	59	26	Yes		7.0	4.00	18	21100.80	1
11	11	C28	600	Owned	Individual	21	21	No	X2			21	4068.00	1
12	12	C27	1097	Owned	Joint	59	47	Yes	X3	3.0	3.00	13	25043.20	0
13	13	C7	3453	Owned	Individual	66	66	No		1.0	2.00	20	17192.00	1
14	14	C5	900	Rented	Individual	20	20	No	X2			18	8364.00	0
15	15	C20	1911	Rented	Individual	27	27	No	X3	2.0	3.00	9	9440.00	0
16	16	C3	1484	Rented	Individual	20	20	No	X3			2	4912.00	0
17	17	C3	1090	Owned	Individual	34	34	No	X1	11.0	1.00	20	6660.00	0
18	18	C7	677	Owned	Individual	43	43	No	X2			19	10386.00	0
19	19	C1	1634	Owned	Individual	55	55	No	X2	1.0	3.00	21	12580.00	0
20	20	C20	973	Owned	Individual	27	27	No				4	8050.00	0
21	21	C9	3543	Owned	Individual	32	32	No	X2	3.0	3.00	16	12060.00	0
22	22	C24	1127	Rented	Individual	23	23	No	X2			16	10352.00	0
23	23	C25	787	Rented	Individual	18	18	No	X6			22	2828.00	0
24	24	C1	2862	Rented	Individual	22	22	No	X6			19	5416.00	0
25	25	C4	2182	Rented	Individual	22	22	No	X1	1.0	3.00	22	6370.00	0
26	26	C5	2276	Rented	Individual	25	25	No	X1			12	7128.00	0
27	27	C4	3283	Rented	Individual	21	21	No	X1			19	7230.00	0
28	28	C9	855	Rented	Individual	21	21	No	X5			16	3744.00	1

Preview of first dataset (Train data)

2

Primary DataFrame Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50882 entries, 0 to 50881
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      50882 non-null  int64
 1   City_Code               50882 non-null  object
 2   Region_Code             50882 non-null  int64
 3   Accomodation_Type       50882 non-null  object
 4   Reco_Insurance_Type     50882 non-null  object
 5   Upper_Age               50882 non-null  int64
 6   Lower_Age               50882 non-null  int64
 7   Is_Spouse               50882 non-null  object
 8   Health Indicator        39191 non-null  object
 9   Holding_Policy_Duration 30631 non-null  object
 10  Holding_Policy_Type     30631 non-null  float64
 11  Reco_Policy_Cat         50882 non-null  int64
 12  Reco_Policy_Premium     50882 non-null  float64
 13  Response                50882 non-null  int64
dtypes: float64(2), int64(6), object(6)
memory usage: 5.4+ MB
```

	ID	Region_Code	Upper_Age	Lower_Age	Holding_Policy_Type	Reco_Policy_Cat	Reco_Policy_Premium	Response
count	50882.000000	50882.000000	50882.000000	50882.000000	30631.000000	50882.000000	50882.000000	50882.000000
mean	25441.500000	1732.788707	44.856275	42.738866	2.439228	15.115188	14183.950069	0.239947
std	14688.512535	1424.081652	17.310271	17.319375	1.025923	6.340663	6590.074873	0.427055
min	1.000000	1.000000	18.000000	16.000000	1.000000	1.000000	2280.000000	0.000000
25%	12721.250000	523.000000	28.000000	27.000000	1.000000	12.000000	9248.000000	0.000000
50%	25441.500000	1391.000000	44.000000	40.000000	3.000000	17.000000	13178.000000	0.000000
75%	38161.750000	2667.000000	59.000000	57.000000	3.000000	20.000000	18096.000000	0.000000
max	50882.000000	6194.000000	75.000000	75.000000	4.000000	22.000000	43350.400000	1.000000

Information of first dataset (Train Data)

3

Result DataFrame

	ID	City_Cod	Region_C	Accomoda	Reco_Insurance	Upper_Ag	Lower_Ag	Is_Spous	Health Indi	Holding_P	Holding_P	Reco_Poli	Reco_Poli
1	50883	C1	156	Owned	Individual	30	30	No		6.0	3.00	5	11934.00
2	50884	C4	7	Owned	Joint	69	68	Yes	X1	3.0	3.00	18	32204.80
3	50885	C1	564	Rented	Individual	28	28	No	X3	2.0	4.00	17	9240.00
4	50886	C3	1177	Rented	Individual	23	23	No	X3	3.0	3.00	18	9086.00
5	50887	C1	951	Owned	Individual	75	75	No	X3			5	22534.00
6	50888	C1	1329	Rented	Individual	24	24	No	X2			18	6150.00
7	50889	C2	3479	Owned	Individual	56	56	No	X5	14+	4.00	17	19152.00
8	50890	C13	396	Rented	Individual	41	41	No				16	11034.00
9	50891	C18	513	Owned	Individual	22	22	No	X3			22	10784.00
10	50892	C3	957	Owned	Joint	41	37	Yes	X5	6.0	1.00	22	16934.40
11	50893	C1	916	Rented	Individual	22	22	No	X4			5	9422.00
12	50894	C16	1113	Owned	Individual	38	38	No	X4	2.0	3.00	21	14168.00
13	50895	C17	636	Owned	Individual	42	42	No	X2	5.0	2.00	17	14184.00
14	50896	C1	1112	Owned	Individual	31	31	No				19	7236.00
15	50897	C2	2371	Rented	Individual	35	35	No	X2	14+	3.00	18	9002.00
16	50898	C11	2000	Owned	Joint	46	37	No				20	18333.00
17	50899	C2	133	Owned	Individual	44	44	No	X3			11	13200.00
18	50900	C7	4535	Owned	Individual	29	29	No				20	7488.00
19	50901	C21	4336	Rented	Individual	60	60	No	X1			9	19200.00
20	50902	C34	858	Rented	Individual	54	54	No	X4	1.0	3.00	14	14586.00
21	50903	C1	3651	Rented	Individual	31	31	No	X1			13	8428.00
22	50904	C14	3329	Rented	Individual	27	27	No	X4			20	7896.00
23	50905	C1	16	Owned	Individual	71	71	No	X5	14+	3.00	18	16042.00
24	50906	C1	183	Owned	Individual	75	75	No	X1	5.0	1.00	21	18720.00
25	50907	C3	3891	Owned	Joint	68	66	Yes	X1	5.0	3.00	9	24206.00
26	50908	C2	2810	Owned	Joint	55	54	Yes	X2	4.0	3.00	3	23091.20

Preview of second dataset (Test Data)

4

Result DataFrame Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21805 entries, 0 to 21804
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      21805 non-null  int64
 1   City_Code               21805 non-null  object
 2   Region_Code             21805 non-null  int64
 3   Accomodation_Type       21805 non-null  object
 4   Reco_Insurance_Type     21805 non-null  object
 5   Upper_Age               21805 non-null  int64
 6   Lower_Age               21805 non-null  int64
 7   Is_Spouse               21805 non-null  object
 8   Health Indicator        16778 non-null  object
 9   Holding_Policy_Duration 13202 non-null  object
 10  Holding_Policy_Type     13202 non-null  float64
 11  Reco_Policy_Cat         21805 non-null  int64
 12  Reco_Policy_Premium     21805 non-null  float64
dtypes: float64(2), int64(5), object(6)
memory usage: 2.2+ MB
```

	ID	Region_Code	Upper_Age	Lower_Age	Holding_Policy_Type	Reco_Policy_Cat	Reco_Policy_Premium
count	21805.000000	21805.000000	21805.000000	21805.000000	13202.000000	21805.000000	21805.000000
mean	61785.000000	1748.737491	44.877734	42.748085	2.440085	15.138363	14220.306581
std	6294.705646	1438.358949	17.254898	17.269112	1.037627	6.302805	6497.996164
min	50883.000000	1.000000	18.000000	16.000000	1.000000	1.000000	2152.000000
25%	56334.000000	535.000000	28.000000	27.000000	1.000000	12.000000	9285.000000
50%	61785.000000	1392.000000	44.000000	41.000000	3.000000	17.000000	13244.000000
75%	67236.000000	2712.000000	59.000000	57.000000	3.000000	20.000000	18201.600000
max	72687.000000	6185.000000	75.000000	75.000000	4.000000	22.000000	43776.000000

Information on the second dataset (Test Data)

5

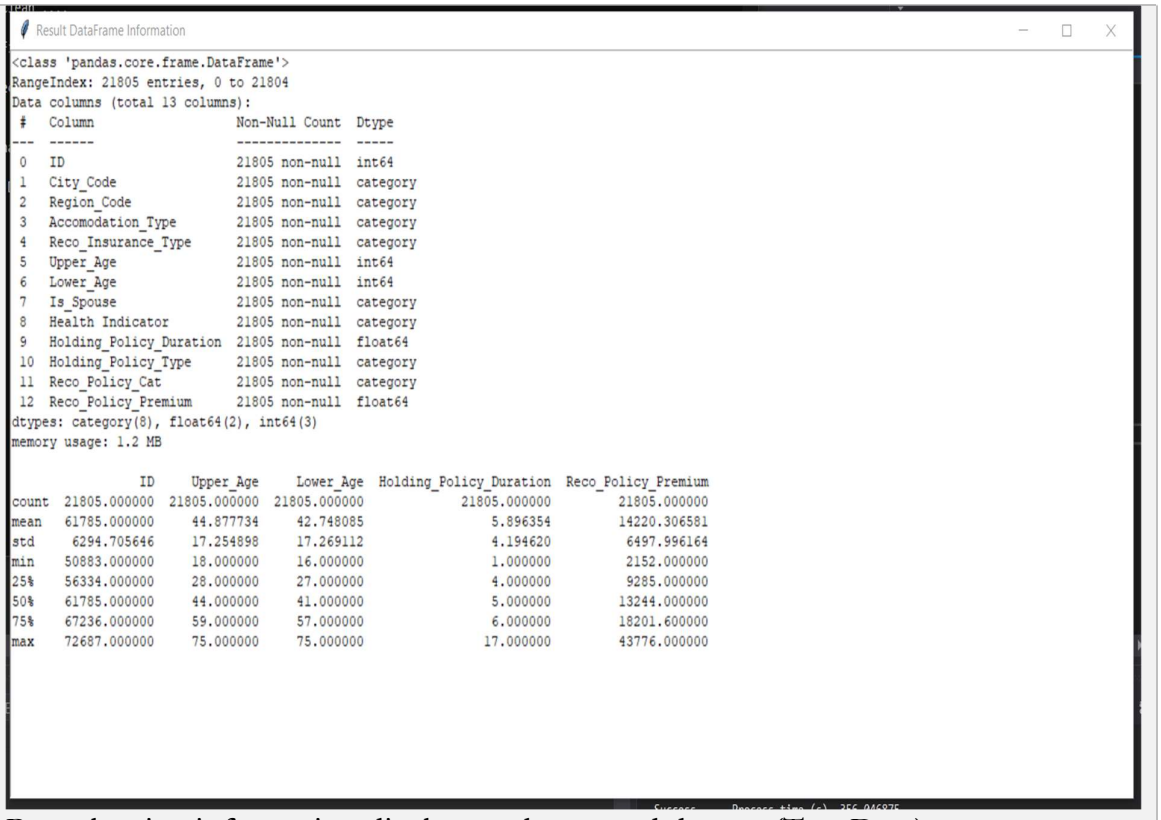
Primary DataFrame Information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50882 entries, 0 to 50881
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   ID                      50882 non-null  int64
 1   City_Code               50882 non-null  category
 2   Region_Code             50882 non-null  category
 3   Accomodation_Type       50882 non-null  category
 4   Reco_Insurance_Type     50882 non-null  category
 5   Upper_Age               50882 non-null  int64
 6   Lower_Age               50882 non-null  int64
 7   Is_Spouse               50882 non-null  category
 8   Health Indicator        50882 non-null  category
 9   Holding_Policy_Duration 50882 non-null  float64
 10  Holding_Policy_Type     50882 non-null  category
 11  Reco_Policy_Cat         50882 non-null  category
 12  Reco_Policy_Premium     50882 non-null  float64
 13  Response                50882 non-null  category
dtypes: category(9), float64(2), int64(3)
memory usage: 2.6 MB
```

	ID	Upper_Age	Lower_Age	Holding_Policy_Duration	Reco_Policy_Premium
count	50882.000000	50882.000000	50882.000000	50882.000000	50882.000000
mean	25441.500000	44.856275	42.738866	5.866770	14183.950069
std	14688.512535	17.310271	17.319375	4.175645	6590.074873
min	1.000000	18.000000	16.000000	1.000000	2280.000000
25%	12721.250000	28.000000	27.000000	4.000000	9248.000000
50%	25441.500000	44.000000	40.000000	5.000000	13178.000000
75%	38161.750000	59.000000	57.000000	6.000000	18096.000000
max	50882.000000	75.000000	75.000000	17.000000	43350.400000

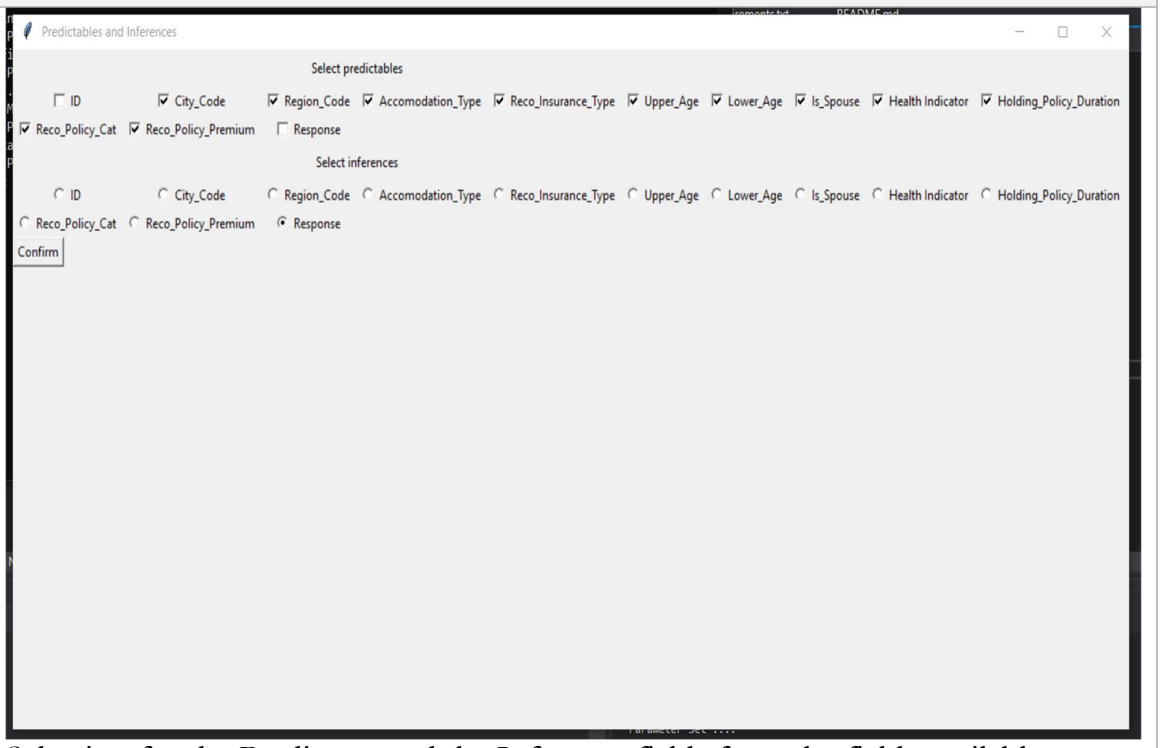
Post cleaning information display on the first dataset (Train Data)

6



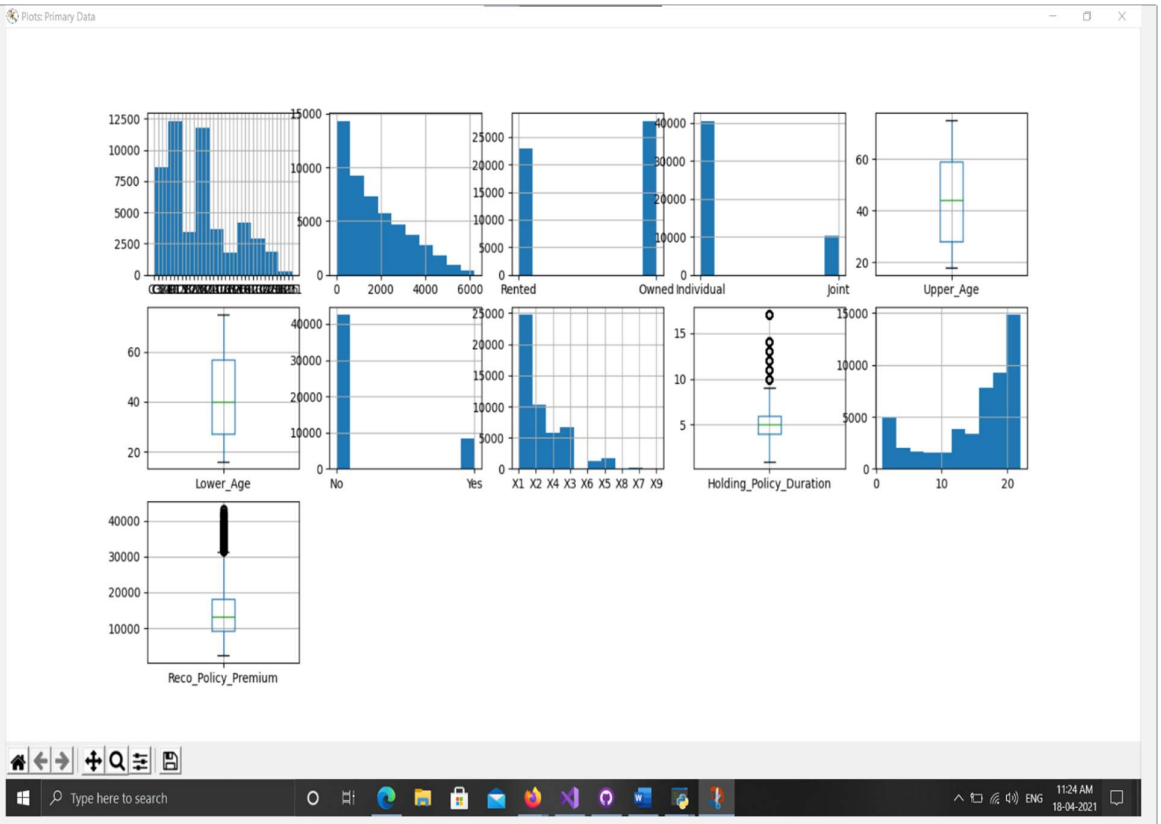
Post cleaning information display on the second dataset (Test Data)

7



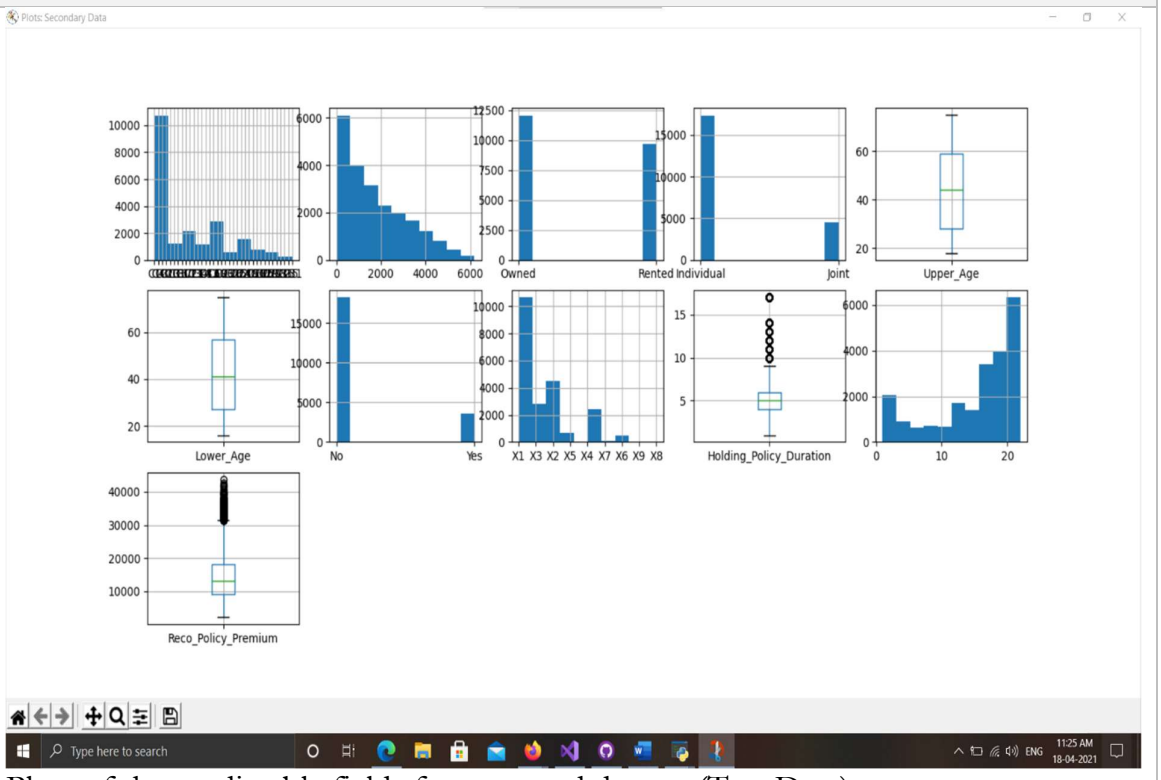
Selection for the Predictors and the Inference fields from the fields available. Worked from the first dataset (Train Data)

8



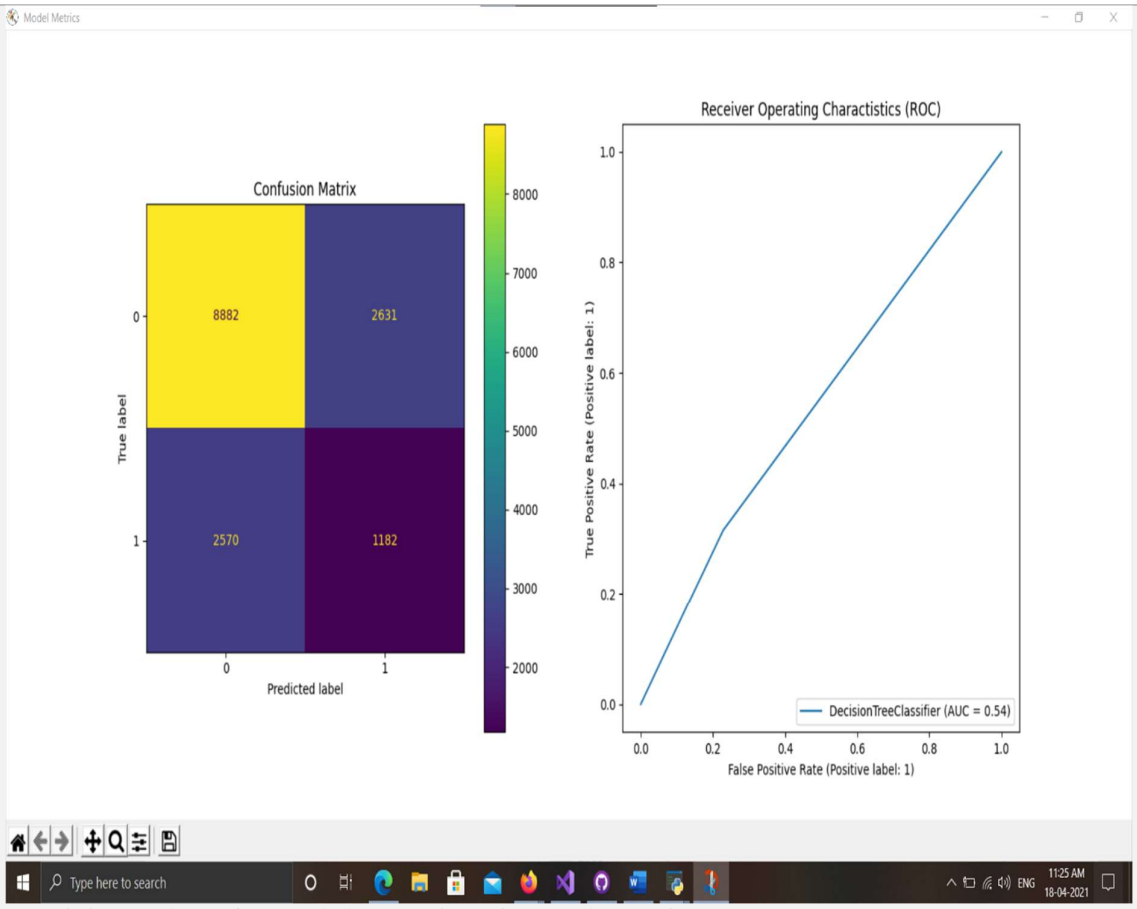
Plots of the predictable fields from first dataset (Train Data)

9



Plots of the predictable fields from second dataset (Test Data)

10

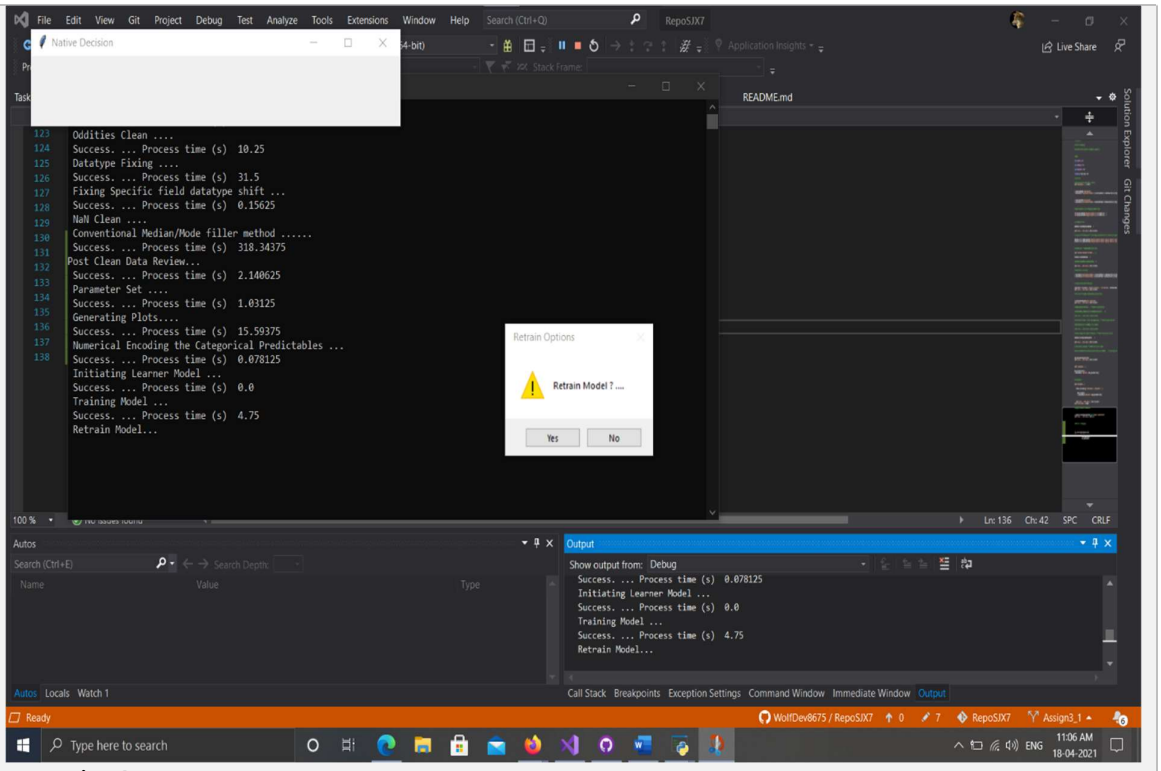


Model Metrics after training and each retrain cycles

11

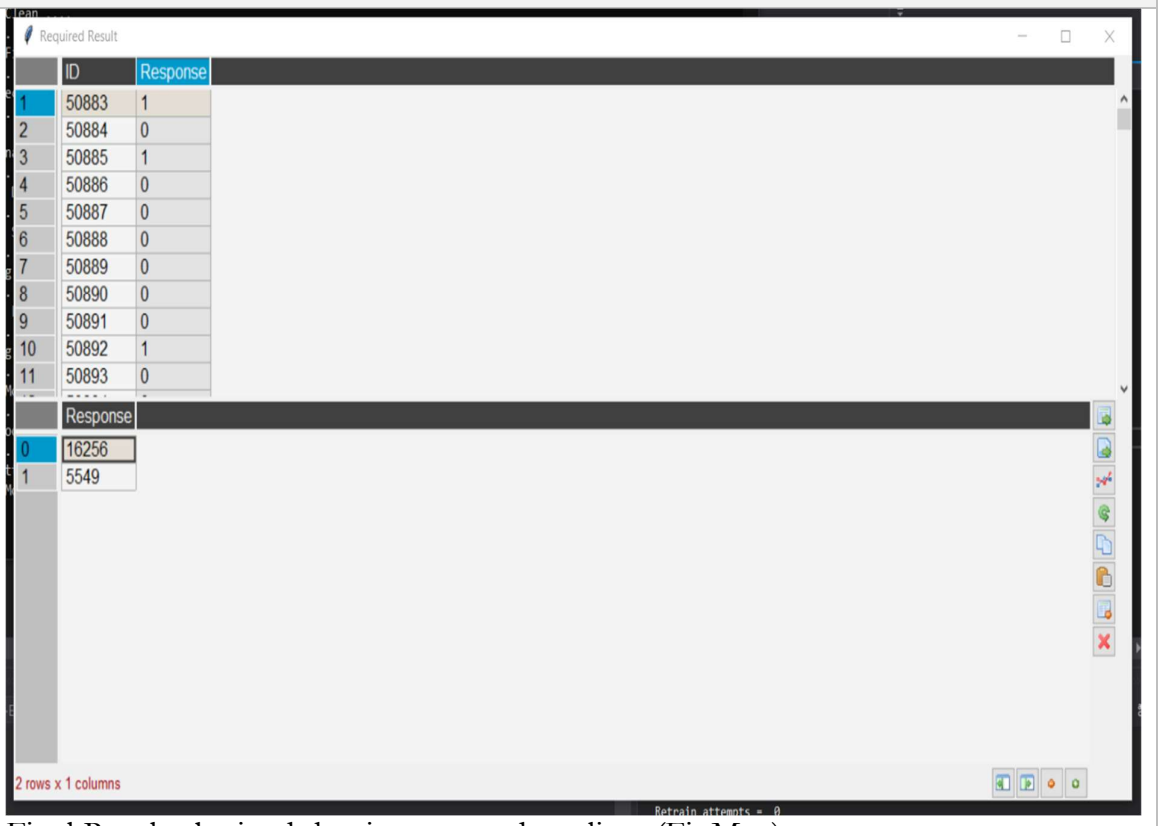
Value based model metrics after training and each retrain cycles

12



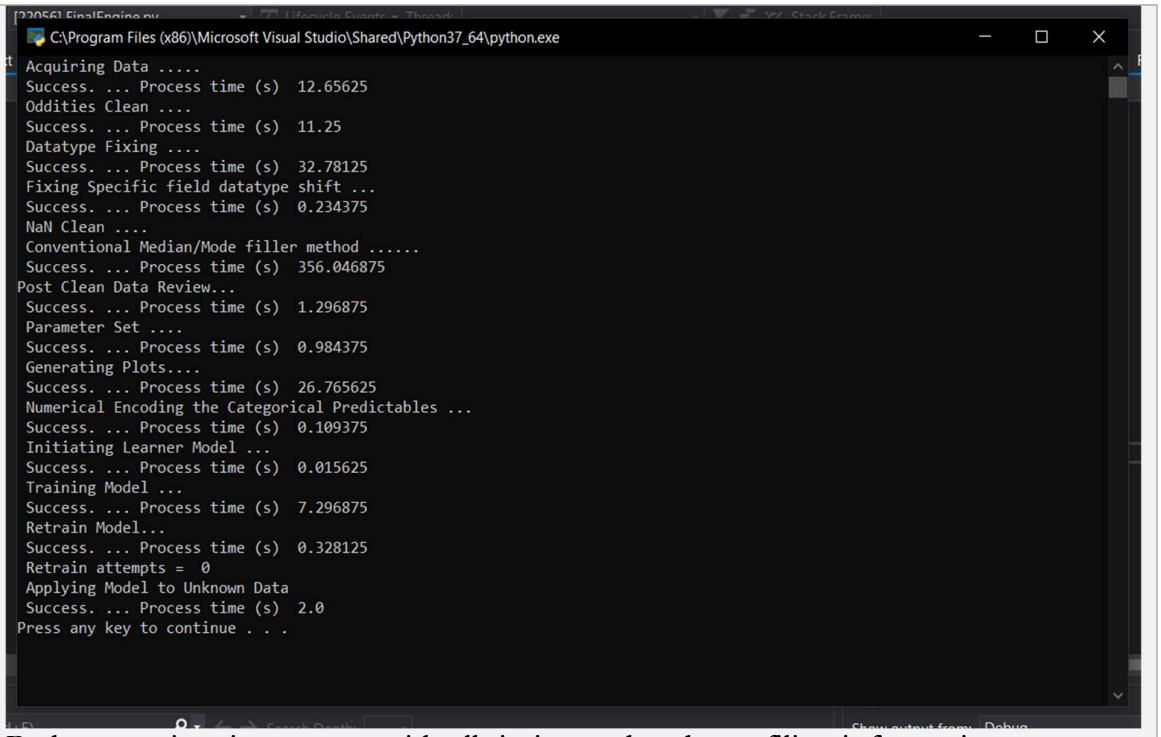
Retrain Prompts

13



Final Result obtained that is presented to client (FinMan)

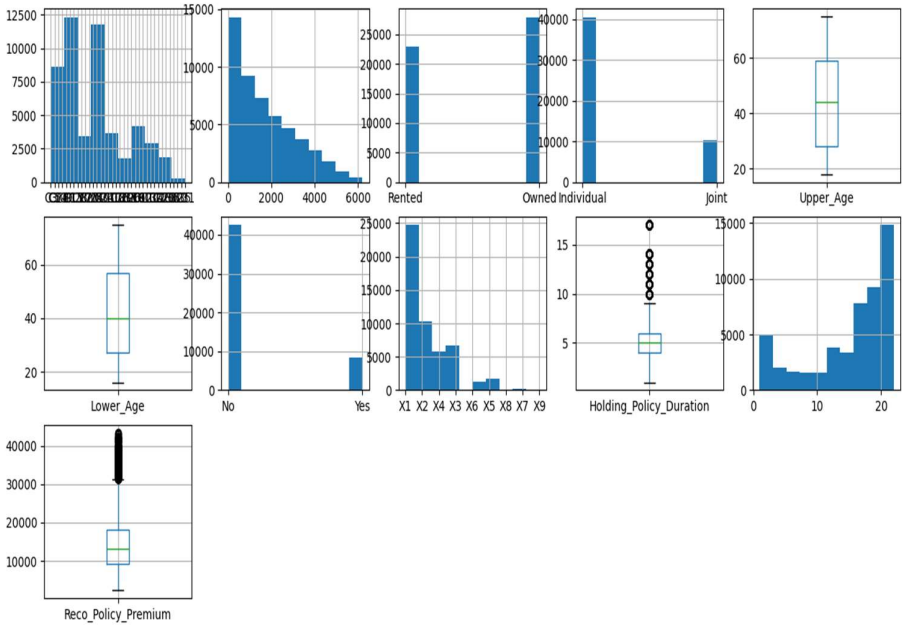
14



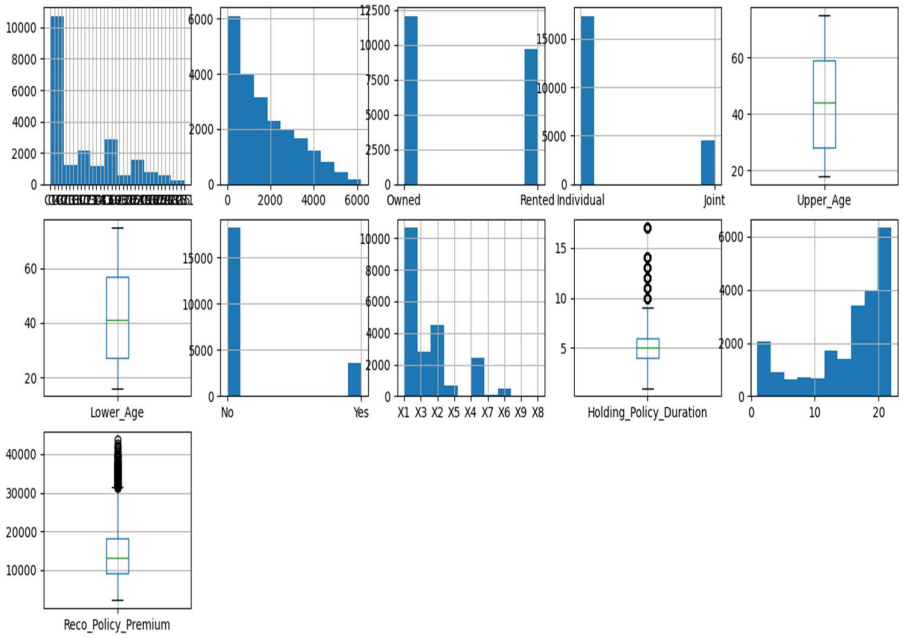
Python runtime interpreter with all timing and code profiling information

Pre – Model Generation Plots

Plot of the Predictors from the Train dataset

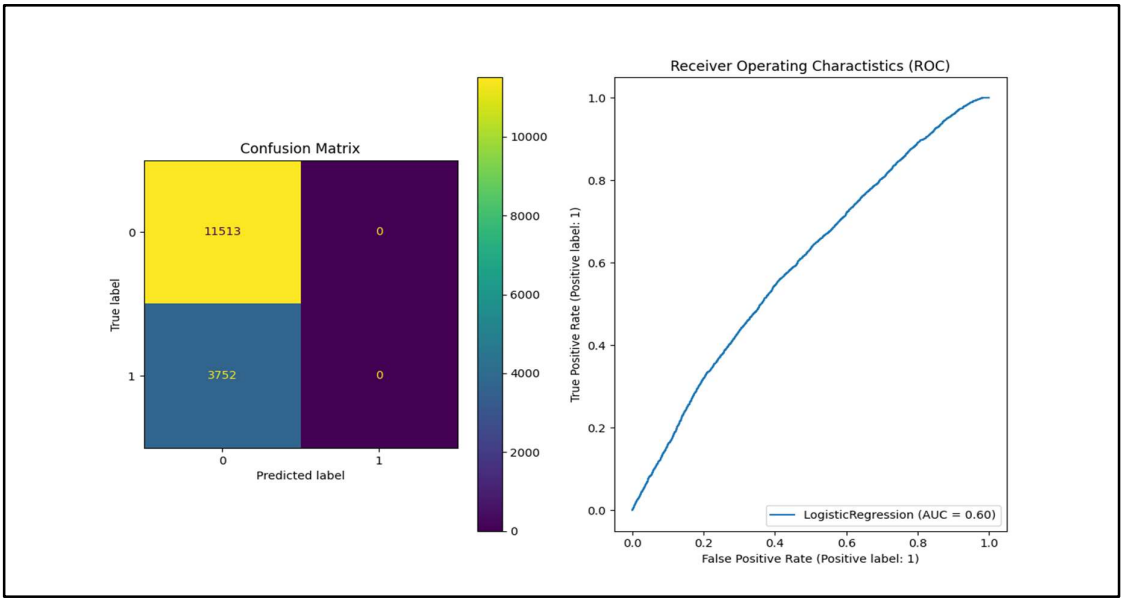


Plot of the Predictors from the Test dataset



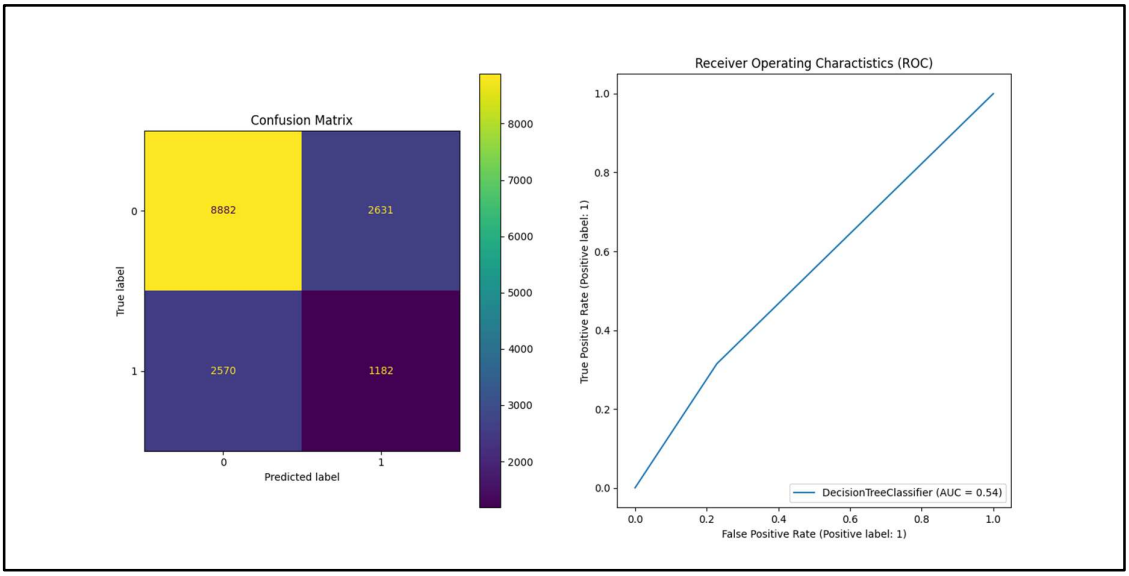
Why Logistic Regression was Avoided

The Logistic regression as a model performed with accuracy of ~75% as compared to ~65% for the decision trees, but the problem generated by the Logistic regression model is that it created an unfit model which faces an "All No Recurrence" as shown in the metrics plot for the Logistic Regression below.



As visible from the confusion matrix of the model we see that the model completely understands to respond to negatives but for positives this same model doesn't understands anything and predicts as negatives and as a result the final presentable file containing all the responses for the sales is usually predicted as **'No Sales Pitch'** for any customer of all 21805 entries almost 99% of the runs, tested on 30 consecutive logistic regression modellings each of them with variation of max 6 and min 2 retrain cycles. In all of these cases the model had 21805 zero responses, depicting a case of All No Recurrence, which directly stems from the fact that the confusion matrix displays for predicted positives all are zeros.

Contrary to this when tested with decision tree (CART – Classification and Regression Trees) model we see that this type of No Recurrence error is never encountered, as shown in the metrics of the decision trees.



When studied and with proper assessment we see that the error we encountered in Logistic Regression is overcome in this model. Although we had to tradeoff the accuracy loss ~10% by value (~75% for logistic and ~65% for Decision Trees), specificity loss ~23% by value (~100% for logistic and ~77% for Decision Trees), and AUC loss ~6% by value (~0.60 for logistic and ~0.54 for Decision Trees), the model obtained is far more suitable for handling and further operations in interests of business growth keeping in mind the fact that an algorithm is still a calculation trying to understand a person by his previous decisions and a person's brain is far more complex and can randomly shift for the reverse like an unpredictable Brownian motion**.

** **Brownian** motion is the random motion of particles suspended in a fluid (a liquid or a gas) resulting from their collision with the fast-moving atoms or molecules in the gas or liquid. This transport phenomenon is named after the botanist Robert Brown

Coding Profiling Metrics

The time taken for each process as timed by the code itself in seconds

```
Acquiring Data .....
 Success. ... Process time (s)  12.65625
 Oddities Clean ....
 Success. ... Process time (s)  11.25
 Datatype Fixing ....
 Success. ... Process time (s)  32.78125
 Fixing Specific field datatype shift ...
 Success. ... Process time (s)  0.234375
 NaN Clean ....
 Conventional Median/Mode filler method ......
 Success. ... Process time (s)  356.046875
Post Clean Data Review...
 Success. ... Process time (s)  1.296875
 Parameter Set ....
 Success. ... Process time (s)  0.984375
 Generating Plots....
 Success. ... Process time (s)  26.765625
 Numerical Encoding the Categorical Predictables ...
 Success. ... Process time (s)  0.109375
 Initiating Learner Model ...
 Success. ... Process time (s)  0.015625
 Training Model ...
 Success. ... Process time (s)  7.296875
 Retrain Model...
 Success. ... Process time (s)  0.328125
 Retrain attempts =  0
 Applying Model to Unknown Data
 Success. ... Process time (s)  2.0
Press any key to continue . . .
```

Solution Metrics

With all checks carried out on the data the solution as obtained from the training the model we get the following information

Pre-training information

- Primary data entries= 50882
- Model generation $Train:Test$ Ratio= $7:3$

Post Training Information

- True Positives = 1182
- True Negatives = 8882
- False Positives = 2631
- False Negatives = 2520
- Accuracy = 65.928595%
- Precision = 30.999213%
- Recall = 31.503198%
- Specificity = 77.147572%
- F – Score = 31.249174%
- ROC – AUC Score = 0.54325385

All values are system generated via coding and prebuilt functions available from Scikit – Learner python library package

Resultant Reached

The decision tree created as a resultant of the training is given below.

```
|--- feature_9 <= 3.50
|   |--- feature_9 <= 0.50
|   |   |--- feature_7 <= 6.50
|   |   |   |--- feature_5 <= 74.50
|   |   |   |   |--- feature_7 <= 4.50
|   |   |   |   |   |--- feature_8 <= 1.50
|   |   |   |   |   |   |--- feature_5 <= 61.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_5 >  61.50
|   |   |   |   |   |   |   |--- feature_4 <= 62.50
|   |   |   |   |   |   |   |   |--- feature_0 <= 9.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_0 >  9.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_4 >  62.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_8 >  1.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_7 >  4.50
|   |   |   |   |   |--- feature_8 <= 15.00
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_8 >  15.00
|   |   |   |   |   |   |--- feature_5 <= 56.00
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_5 >  56.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |--- feature_5 >  74.50
|   |   |   |   |--- feature_0 <= 1.00
|   |   |   |   |   |--- feature_8 <= 9.50
|   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_8 >  9.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_0 >  1.00
|   |   |   |   |   |--- feature_1 <= 4488.50
|   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_1 >  4488.50
|   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |--- feature_7 >  6.50
|   |   |   |--- feature_10 <= 5855.00
|   |   |   |   |--- feature_1 <= 3465.50
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_1 >  3465.50
|   |   |   |   |   |--- class: 1
|   |   |   |--- feature_10 >  5855.00
|   |   |   |   |--- class: 0
|   |--- feature_9 >  0.50
|   |   |--- feature_0 <= 16.50
|   |   |   |--- feature_0 <= 4.50
|   |   |   |   |--- feature_8 <= 9.50
|   |   |   |   |   |--- feature_1 <= 5073.00
|   |   |   |   |   |   |--- feature_0 <= 3.50
|   |   |   |   |   |   |   |--- feature_9 <= 1.50
|   |   |   |   |   |   |   |   |--- feature_4 <= 20.50
```

```
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 7787.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  7787.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 317.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  317.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_4 >  20.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 2.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 67.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  67.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  2.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 27.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  27.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |--- feature_9 >  1.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 788.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  788.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1777.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1777.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3304.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3304.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_0 >  3.50
|   |   |   |   |   |   |   |   |--- feature_5 <= 19.50
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_5 >  19.50
|   |   |   |   |   |   |   |   |   |--- feature_7 <= 4.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2381.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2381.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2480.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2480.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_7 >  4.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_1 >  5073.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_8 >  9.50
|   |   |   |   |   |   |--- feature_4 <= 40.50
|   |   |   |   |   |   |   |--- feature_3 <= 0.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_3 >  0.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_4 >  40.50
|   |   |   |   |   |   |   |--- feature_10 <= 17650.00
|   |   |   |   |   |   |   |   |--- feature_1 <= 110.50
|   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |   |   |   |   |--- feature_1 >  110.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_10 >  17650.00
|   |   |   |   |   |   |   |   |--- feature_7 <= 2.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 18540.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_10 >  18540.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 28955.80
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  28955.80
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_7 >  2.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_0 >  4.50
|   |   |   |   |   |--- feature_1 <= 1443.50
|   |   |   |   |   |   |--- feature_0 <= 14.50
|   |   |   |   |   |   |   |--- feature_8 <= 13.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 17.00
|   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  17.00
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 16.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_5 >  16.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 70.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 63.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  63.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  70.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_8 >  13.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 248.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  248.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 628.00
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 48.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  48.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 464.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  464.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  628.00
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 47.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  47.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  1.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_0 >  14.50
|   |   |   |   |   |   |   |--- feature_8 <= 2.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_8 >  2.50
|   |   |   |   |   |   |   |   |--- feature_5 <= 24.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 521.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  521.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_5 >  24.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 55.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  55.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |--- feature_1 >  1443.50
|   |   |   |   |   |--- feature_5 <= 20.50
|   |   |   |   |   |   |--- feature_0 <= 10.50
|   |   |   |   |   |   |   |--- feature_10 <= 7482.00
|   |   |   |   |   |   |   |   |--- feature_4 <= 19.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_4 >  19.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_10 >  7482.00
|   |   |   |   |   |   |   |   |--- feature_1 <= 1938.00
|   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  1938.00
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_0 >  10.50
|   |   |   |   |   |   |   |--- feature_10 <= 27552.00
|   |   |   |   |   |   |   |   |--- feature_1 <= 2258.00
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 21.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_4 >  21.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  2258.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_10 >  27552.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_5 >  20.50
|   |   |   |   |   |   |--- feature_0 <= 10.50
|   |   |   |   |   |   |   |--- feature_0 <= 9.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 2409.00
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_8 >  1.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 16633.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  16633.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  2409.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 5012.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4656.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  4656.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_10 >  5012.00
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |--- feature_0 >  9.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 4082.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  4082.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 4106.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  4106.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_0 >  10.50
|   |   |   |   |   |   |   |--- feature_10 <= 16620.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 16337.00
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 27.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2709.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2709.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |--- feature_5 >  27.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 9060.00
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  9060.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
|   |   |   |   |   |   |   |   |   |--- feature_10 >  16337.00
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 41.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  41.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_10 >  16620.00
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1670.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 2.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  2.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1670.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 24.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  24.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2143.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2143.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |--- feature_0 >  16.50
|   |   |   |--- feature_0 <= 33.50
|   |   |   |   |--- feature_0 <= 30.50
|   |   |   |   |   |--- feature_1 <= 54.00
|   |   |   |   |   |   |--- feature_9 <= 2.00
|   |   |   |   |   |   |   |--- feature_8 <= 7.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_8 >  7.00
|   |   |   |   |   |   |   |   |--- feature_8 <= 13.00
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_8 >  13.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_9 >  2.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_1 >  54.00
|   |   |   |   |   |   |--- feature_5 <= 16.50
|   |   |   |   |   |   |   |--- feature_7 <= 1.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_7 >  1.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_5 >  16.50
|   |   |   |   |   |   |   |--- feature_10 <= 8743.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 8702.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 6120.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 133.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  133.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |--- feature_10 >  6120.00
|   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 27.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |   |--- feature_0 >  27.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  8702.00
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 23.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  23.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_10 >  8743.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1767.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 45.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  45.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 53.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  53.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1767.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1769.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1769.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_9 <= 2.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |   |--- feature_9 >  2.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_0 >  30.50
|   |   |   |   |   |   |--- feature_10 <= 25095.20
|   |   |   |   |   |   |   |--- feature_5 <= 70.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 3890.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2338.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2143.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2143.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2338.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  3890.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 16.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_5 >  16.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 129.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 95.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  95.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  129.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 31.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  31.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |--- feature_5 >  70.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 15931.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  15931.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_10 >  25095.20
|   |   |   |   |   |   |   |--- feature_8 <= 6.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_8 >  6.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_0 >  33.50
|   |   |   |   |   |--- feature_7 <= 7.00
|   |   |   |   |   |   |--- feature_1 <= 4688.50
|   |   |   |   |   |   |   |--- feature_5 <= 44.50
|   |   |   |   |   |   |   |   |--- feature_5 <= 42.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 3921.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 600.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  600.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 3873.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  3873.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |   |   |   |   |   |--- feature_10 >  3921.00
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 14.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 56.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  56.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- feature_8 >  14.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 9479.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  9479.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_5 >  42.50
|   |   |   |   |   |   |   |   |   |--- feature_7 <= 2.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 8484.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_10 >  8484.00
|   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_7 >  2.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_5 >  44.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 3089.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 411.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 329.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  329.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  411.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  3089.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3235.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3235.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_1 >  4688.50
|   |   |   |   |   |   |   |--- feature_4 <= 41.00
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_4 >  41.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_7 >  7.00
|   |   |   |   |   |   |--- class: 1
|--- feature_9 >  3.50
|   |--- feature_9 <= 20.50
|   |   |--- feature_9 <= 19.50
|   |   |   |--- feature_9 <= 10.50
|   |   |   |   |--- feature_9 <= 4.50
|   |   |   |   |   |--- feature_0 <= 5.50
|   |   |   |   |   |   |--- feature_1 <= 28.00
|   |   |   |   |   |   |   |--- feature_1 <= 4.50
|   |   |   |   |   |   |   |   |--- feature_4 <= 74.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_4 >  74.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_1 >  4.50
|   |   |   |   |   |   |   |   |--- feature_10 <= 8190.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 7662.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  7662.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_10 >  8190.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 14647.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 14336.00
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  14336.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- feature_10 >  14647.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_1 >  28.00
|   |   |   |   |   |   |   |--- feature_1 <= 74.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_1 >  74.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 6.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 32.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3191.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3191.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_5 >  32.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 42.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  42.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |--- feature_8 >  6.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 62.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 28169.40
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  28169.40
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  62.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |--- feature_0 >  5.50
|   |   |   |   |   |   |--- feature_0 <= 28.00
|   |   |   |   |   |   |   |--- feature_1 <= 1484.50
|   |   |   |   |   |   |   |   |--- feature_7 <= 1.50
|   |   |   |   |   |   |   |   |   |--- feature_6 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_6 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 8.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  8.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_7 >  1.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 33.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  33.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  4.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  1484.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1919.00
|   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 21.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_0 >  21.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1919.00
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 27.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 25.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  25.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  27.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 7.50
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  7.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |--- feature_0 >  28.00
|   |   |   |   |   |   |   |--- feature_1 <= 171.00
|   |   |   |   |   |   |   |   |--- feature_6 <= 0.50
|   |   |   |   |   |   |   |   |   |--- feature_7 <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_7 >  1.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 57.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  57.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_6 >  0.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  171.00
|   |   |   |   |   |   |   |   |--- feature_0 <= 31.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 2599.00
|   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |--- feature_1 >  2599.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2759.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2759.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |--- feature_0 >  31.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1238.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 12803.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  12803.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1238.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 20964.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  20964.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |--- feature_9 >  4.50
|   |   |   |   |   |   |--- feature_1 <= 8.50
|   |   |   |   |   |   |   |--- feature_9 <= 5.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_9 >  5.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 7.00
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 14.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_0 >  14.50
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_8 >  7.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_1 >  8.50
|   |   |   |   |   |   |   |--- feature_1 <= 123.50
|   |   |   |   |   |   |   |   |--- feature_10 <= 3882.00
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 22.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_4 >  22.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_10 >  3882.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 19.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 74.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
```

```
|   |   |   |   |   |   |   |   |   |--- feature_5 >  74.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 13.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  13.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  19.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 23.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  23.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |--- feature_1 >  123.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 140.50
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 26.00
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 6.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 16764.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  16764.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  6.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  21.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_0 >  26.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  140.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 20.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 47.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 19.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  19.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  47.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 320.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  320.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |--- feature_5 >  20.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 5054.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 19
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  5054.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2587.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 20
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2587.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 19
|   |   |   |   |--- feature_9 >  10.50
|   |   |   |   |   |--- feature_9 <= 14.50
|   |   |   |   |   |   |--- feature_9 <= 13.50
|   |   |   |   |   |   |   |--- feature_5 <= 21.50
|   |   |   |   |   |   |   |   |--- feature_9 <= 11.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 3564.00
|   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 18.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_0 >  18.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_10 >  3564.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4716.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 87.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  87.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
```

```
|   |   |   |   |   |   |   |   |   |--- feature_10 >  4716.00
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 19.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  19.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |--- feature_9 >  11.50
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 6.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 12.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 951.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  951.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  12.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_0 >  6.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 70.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  70.00
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 17
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |--- feature_5 >  21.50
|   |   |   |   |   |   |   |   |--- feature_0 <= 0.50
|   |   |   |   |   |   |   |   |   |--- feature_9 <= 11.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3212.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 340.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  340.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3212.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3524.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3524.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |--- feature_9 >  11.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1865.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1791.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1791.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1865.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_0 >  0.50
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 1.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 27.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 22.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  22.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  27.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |--- feature_0 >  1.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3632.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 6.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  6.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 34
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3632.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 10.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  10.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 14
|   |   |   |   |   |   |--- feature_9 >  13.50
```

```
|   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 92.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 7206.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  7206.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 18248.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  18248.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 19.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  19.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |--- feature_1 >  92.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 33.50
|   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 13.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_0 >  13.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 25.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  25.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |--- feature_5 >  33.50
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 14
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 131.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  131.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |--- feature_5 <= 16.50
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 21.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_0 >  21.00
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 52.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  52.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_5 >  16.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 24587.20
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 23955.20
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 8.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 26
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  8.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  23955.20
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2365.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2365.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- feature_10 >  24587.20
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 18.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  18.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 13.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  13.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |--- feature_9 >  14.50
|   |   |   |   |   |   |--- feature_9 <= 15.50
|   |   |   |   |   |   |   |--- feature_4 <= 41.50
```

```
|   |   |   |   |   |   |   |   |--- feature_1 <= 592.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 483.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 6790.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 6720.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  6720.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  6790.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 16
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  483.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 25.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  25.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 17037.90
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  17037.90
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  592.00
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 13.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1290.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 965.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  965.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1290.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1724.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1724.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 16
|   |   |   |   |   |   |   |   |   |--- feature_0 >  13.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 641.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 34.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  34.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  641.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |--- feature_4 >  41.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 2284.50
|   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 35284.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 24
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  35284.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 65.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  65.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 10784.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 10248.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  10248.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  10784.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
```

```
|   |   |   |   |   |   |   |   |--- feature_1 >  2284.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 43.50
|   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_5 >  43.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 4864.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3568.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3568.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  4864.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 10.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  10.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_9 >  15.50
|   |   |   |   |   |   |   |--- feature_0 <= 33.50
|   |   |   |   |   |   |   |   |--- feature_9 <= 16.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 17430.00
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 20.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  20.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 17191.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 41
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  17191.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_10 >  17430.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 5081.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 8.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 19
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  8.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 19
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  5081.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_9 >  16.50
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 22.50
|   |   |   |   |   |   |   |   |   |   |--- feature_9 <= 17.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 32.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  32.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |--- feature_9 >  17.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 2692.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  2692.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 24
|   |   |   |   |   |   |   |   |   |--- feature_4 >  22.50
|   |   |   |   |   |   |   |   |   |   |--- feature_9 <= 17.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 29
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 35
|   |   |   |   |   |   |   |   |   |   |--- feature_9 >  17.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 9.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  9.00
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 35
|   |   |   |   |   |   |   |--- feature_0 >  33.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 152.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 28.00
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 6.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  6.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  28.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 8694.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 24.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  24.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  8694.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  152.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 11012.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 7854.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 7422.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  7422.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  7854.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_10 >  11012.00
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 63.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 4572.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 20
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  4572.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  63.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_9 <= 16.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_9 >  16.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |--- feature_9 >  19.50
|   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |--- feature_0 <= 0.50
|   |   |   |   |   |   |--- feature_1 <= 3956.50
|   |   |   |   |   |   |   |--- feature_1 <= 1626.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 221.00
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 61.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 59.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  59.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 15696.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  15696.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- feature_5 >  61.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 6.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  6.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 74.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  74.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
```

```
|   |   |   |   |   |   |   |   |--- feature_1 >  221.00
|   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 10.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 13277.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  13277.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  10.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 61.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  61.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 65.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  65.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 798.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  798.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |--- feature_1 >  1626.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 2236.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1821.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1821.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2004.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1967.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1967.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2004.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  2236.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 74.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 20913.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 18086.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  18086.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  20913.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_5 >  74.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3526.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3526.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  3956.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_0 >  0.50
|   |   |   |   |   |   |--- feature_0 <= 30.50
|   |   |   |   |   |   |   |--- feature_1 <= 311.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 33614.40
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 34.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4193.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  4193.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  34.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 10737.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  10737.00
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
|   |   |   |   |   |   |   |   |--- feature_10 >  33614.40
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 24.00
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 60.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  60.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 11.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  11.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  24.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_1 >  311.00
|   |   |   |   |   |   |   |   |--- feature_1 <= 360.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 355.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 329.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  329.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  355.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  360.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 31011.60
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 30888.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 69.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 25
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  69.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  30888.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_10 >  31011.60
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 71.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  71.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_0 >  30.50
|   |   |   |   |   |   |   |--- feature_1 <= 595.00
|   |   |   |   |   |   |   |   |--- feature_1 <= 319.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 21040.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 13392.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 4.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  4.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  13392.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 34.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  34.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_10 >  21040.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 11.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  11.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 73.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  73.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |--- feature_1 >  319.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 354.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  354.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 24.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
```

```
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  24.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  5.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 417.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  417.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  595.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 7500.00
|   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 7302.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  7302.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 22.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  22.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 33.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  33.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |--- feature_10 >  7500.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 4593.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 11889.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 29.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  29.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  11889.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 12348.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  12348.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |--- feature_1 >  4593.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 40.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  40.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |--- feature_10 <= 5300.00
|   |   |   |   |   |   |--- feature_0 <= 3.00
|   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |--- feature_10 <= 4347.00
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_10 >  4347.00
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 25.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4558.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  4558.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_4 >  25.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |--- feature_5 <= 25.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_5 >  25.50
|   |   |   |   |   |   |   |   |--- feature_0 <= 1.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_0 >  1.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_0 >  3.00
|   |   |   |   |   |   |   |--- feature_0 <= 34.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 1632.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1617.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1455.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 2.50
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  2.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1455.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1487.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1487.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1617.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  1632.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_0 >  34.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 2801.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  2801.50
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 24.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_4 >  24.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_10 >  5300.00
|   |   |   |   |   |   |--- feature_10 <= 5316.00
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_10 >  5316.00
|   |   |   |   |   |   |   |--- feature_1 <= 5009.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 6004.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 5866.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1808.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 4.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  4.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1808.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2223.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2223.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_10 >  5866.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 5965.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  5965.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  6004.00
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 15.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 73.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 61.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 22
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  61.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  73.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 21199.00
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  21199.00
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  15.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 31.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 64.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  64.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  31.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  5009.00
|   |   |   |   |   |   |   |   |--- class: 1
```

```
|   |--- feature_9 >  20.50
|   |   |--- feature_1 <= 4670.50
|   |   |   |--- feature_1 <= 1707.50
|   |   |   |   |--- feature_1 <= 51.50
|   |   |   |   |   |--- feature_0 <= 6.50
|   |   |   |   |   |   |--- feature_4 <= 49.50
|   |   |   |   |   |   |   |--- feature_10 <= 12146.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 4876.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_10 >  4876.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 8.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  8.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 27.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  27.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |--- feature_10 >  12146.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 23655.00
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 27.00
|   |   |   |   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_5 >  27.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_10 >  23655.00
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_4 >  49.50
|   |   |   |   |   |   |   |--- feature_1 <= 10.50
|   |   |   |   |   |   |   |   |--- feature_10 <= 26628.00
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 60.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_4 >  60.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 69.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  69.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |--- feature_10 >  26628.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  10.50
|   |   |   |   |   |   |   |   |--- feature_10 <= 33518.40
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 19.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_5 >  19.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 8.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  8.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |--- feature_10 >  33518.40
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |--- feature_0 >  6.50
|   |   |   |   |   |   |--- feature_1 <= 12.50
|   |   |   |   |   |   |   |--- feature_0 <= 32.50
|   |   |   |   |   |   |   |   |--- feature_10 <= 6095.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_10 >  6095.00
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 37.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_4 >  37.00
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  5.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |--- feature_0 >  32.50
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_1 >  12.50
```

```
|   |   |   |   |   |   |   |   |--- feature_1 <= 14.00
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 12134.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  12134.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 12456.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  12456.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_1 >  14.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 47.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 70.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 56.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 11
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  56.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  70.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  47.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_1 >  51.50
|   |   |   |   |   |   |--- feature_7 <= 5.50
|   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |--- feature_4 <= 70.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 3633.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_10 >  3633.00
|   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 32.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 30.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 23
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  30.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 8
|   |   |   |   |   |   |   |   |   |   |--- feature_0 >  32.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 64.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  64.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |--- feature_4 >  70.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1080.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 85.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 69.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  69.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  85.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 34.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  34.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1080.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1293.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  5.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1293.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 73.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  73.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |--- feature_5 <= 25.50
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 31.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 393.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 360.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  360.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
```

```
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  393.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 17
|   |   |   |   |   |   |   |   |   |--- feature_0 >  31.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 3786.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  3786.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 10616.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  10616.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |--- feature_5 >  25.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 119.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 61.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  61.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  119.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 995.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_0 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 18
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  995.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 14
|   |   |   |   |   |   |--- feature_7 >  5.50
|   |   |   |   |   |   |   |--- feature_10 <= 13420.00
|   |   |   |   |   |   |   |   |--- feature_10 <= 9086.00
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 12.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_0 >  12.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_10 >  9086.00
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_10 >  13420.00
|   |   |   |   |   |   |   |   |--- feature_0 <= 22.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 66.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  66.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_0 >  22.50
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 63.50
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_4 >  63.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |--- feature_1 >  1707.50
|   |   |   |   |   |--- feature_1 <= 1802.50
|   |   |   |   |   |   |--- feature_5 <= 72.50
|   |   |   |   |   |   |   |--- feature_8 <= 15.00
|   |   |   |   |   |   |   |   |--- feature_5 <= 70.50
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 1729.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 49.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_5 >  49.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 57.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  57.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  1729.50
```

```
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 15407.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  1.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  15407.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |--- feature_5 >  70.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_8 >  15.00
|   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_5 >  72.50
|   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_1 >  1802.50
|   |   |   |   |   |   |--- feature_0 <= 12.50
|   |   |   |   |   |   |   |--- feature_0 <= 10.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 2181.50
|   |   |   |   |   |   |   |   |   |--- feature_5 <= 21.50
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_5 >  21.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 29209.60
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 9168.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  9168.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 9
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  29209.60
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  2181.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4134.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3733.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3733.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  4134.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 3129.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2976.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 22
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2976.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  3129.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |--- feature_0 >  10.50
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 20.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 10936.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  10936.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_4 >  20.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 36.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1837.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1837.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 35.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  35.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_5 >  36.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 56.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 20336.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  20336.00
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |--- feature_5 >  56.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 60.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  60.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |   |   |--- feature_0 >  12.50
|   |   |   |   |   |   |--- feature_0 <= 29.00
|   |   |   |   |   |   |   |--- feature_5 <= 37.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 3.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 6240.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  6240.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1918.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1918.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 6
|   |   |   |   |   |   |   |   |   |--- feature_8 >  3.50
|   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 20.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2294.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  2294.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_4 >  20.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1904.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1904.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 10
|   |   |   |   |   |   |   |   |--- feature_5 >  37.50
|   |   |   |   |   |   |   |   |   |--- feature_4 <= 45.50
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 39.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  39.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 1.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |   |   |--- feature_8 >  1.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_4 >  45.50
|   |   |   |   |   |   |   |   |   |   |--- feature_8 <= 11.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  2.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |--- feature_8 >  11.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 69.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |   |   |   |   |--- feature_5 >  69.50
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_0 >  29.00
|   |   |   |   |   |   |   |   |--- feature_1 <= 2634.50
|   |   |   |   |   |   |   |   |   |--- feature_0 <= 34.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 1907.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 40.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 >  40.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_1 >  1907.50
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 26700.80
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 12
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 >  26700.80
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |   |--- feature_0 >  34.50
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2413.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 17934.00
```

```
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  17934.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |--- feature_1 >  2413.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  2634.50
|   |   |   |   |   |   |   |   |   |--- feature_10 <= 5552.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4089.00
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 1.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 >  1.00
|   |   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  4089.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_10 >  5552.00
|   |   |   |   |   |   |   |   |   |   |--- feature_1 <= 2682.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_1 >  2682.50
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 8540.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  8540.00
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 13
|   |   |   |--- feature_1 >  4670.50
|   |   |   |   |--- feature_4 <= 71.00
|   |   |   |   |   |--- feature_7 <= 3.50
|   |   |   |   |   |   |--- feature_1 <= 4887.50
|   |   |   |   |   |   |   |--- feature_1 <= 4767.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 4737.50
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  4737.50
|   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_1 >  4767.50
|   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |--- feature_1 >  4887.50
|   |   |   |   |   |   |   |--- feature_2 <= 0.50
|   |   |   |   |   |   |   |   |--- feature_1 <= 5239.00
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 4920.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_1 >  4920.00
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 51.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_5 >  51.50
|   |   |   |   |   |   |   |   |   |   |--- feature_5 <= 56.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |   |--- feature_5 >  56.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |--- feature_1 >  5239.00
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 4.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_8 >  4.00
|   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |--- feature_2 >  0.50
|   |   |   |   |   |   |   |   |--- feature_8 <= 11.00
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 2.00
|   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |--- feature_8 >  2.00
|   |   |   |   |   |   |   |   |   |   |--- feature_10 <= 4729.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |   |   |   |   |   |--- feature_10 >  4729.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |   |   |--- feature_8 >  11.00
|   |   |   |   |   |   |   |   |   |--- class: 1
|   |   |   |   |   |--- feature_7 >  3.50
|   |   |   |   |   |   |--- feature_7 <= 5.50
|   |   |   |   |   |   |   |--- class: 0
|   |   |   |   |   |   |--- feature_7 >  5.50
|   |   |   |   |   |   |   |--- class: 1
```

```
|   |   |   |--- feature_4 >  71.00
|   |   |   |   |--- feature_1 <= 5309.00
|   |   |   |   |   |--- class: 0
|   |   |   |   |--- feature_1 >  5309.00
|   |   |   |   |   |--- class: 1
```

Assessment of the Targets met

At the commencement of the project, we had few of the targets that we set out with in path of solving the problem at hand.

1. Cleaning the data
2. EDA on the data
3. Provide graphical insight into the data
4. Standardize and Normalize numerical data fields if and whenever required.
5. Solve the business problem of the client and provide back vital details.

Checking back all the stages and processes involved in the project completion, we can confirm that the project has completed and exploited all the targets that it had set out with, leaving only scopes where further improvements can be made with improving code patterning or with using advanced systems and equipment for running this code.

Client presentability

The presentability of the solution as required in the problem statement is provided in a comma separated variable(csv) file in the project repository in addition to that a set of first 26 lines of the file in question is provided as an example.

	ID	Response
0	50883	1
1	50884	0
2	50885	1
3	50886	0
4	50887	0
5	50888	0
6	50889	0
7	50890	0
8	50891	0
9	50892	1
10	50893	0
11	50894	0
12	50895	0
13	50896	0
14	50897	0
15	50898	1

16	50899	0
17	50900	0
18	50901	1
19	50902	0
20	50903	0
21	50904	0
22	50905	0
23	50906	0

The total data points in this complete file are 21805, an on application of the tree algorithm we get the number of customers to whom the sale of insurances is to be pitched is of 5549 or 25.44829% of the test customer base and the rest 16256 or 74.55171% has a high chance of turning down any possible sale pitched to them.

Conclusion and Inferences

From the Metrics obtained by calculations we can infer the following solutions

1. The Model is accurate on mediocre terms, more work could be done to raise the accuracy levels.
2. The AUC of the model parameters is at a value (0.54) which is borderline good at best but not the perfect.
3. Avoidance of "All No Recurrence" which was an issue we faced at the model generation using Logistic Regression. This kind of error is characterized by a model being heavily biased to a certain type of value rather than taking a unanimous decision.
4. KNN is a heavy algorithm to go with and takes a lot of time to start with although a comparatively cleaner imputer
5. Some sections of the code are too slow and buggy in filtering operations within a DataFrame, this can be better overcome with grouping and aggregate or some smart filtering alternatives if provided by Pandas package. Research on this field alone is needed to reduce time constraints of the code execution.
6. Using more effective and powerful packages in the python library might help in adding more perfection to the obtained solution. In perspective of this issue its is suggestable that in place of the Sklearn package we could use
    a. Caffe 1
    b. Caffe 2
    c. Keras
    d. TensorFlow, etc.

Bibliography

The following materials were referenced for bringing the project to fruition.

- https://www.lotame.com/what-are-the-methods-of-data-collection/
- https://www.guru99.com/what-is-data-analysis.html
- https://www.investopedia.com/terms/d/data-analytics.asp
- https://web.archive.org/web/20201205235717/https://www.ibm.com/in-en/analytics/hadoop/big-data-analytics
- https://www.talend.com/resources/what-is-data-preparation/
- https://www.zarantech.com/blog/importance-of-data-science/
- https://www.trifacta.com/data-cleansing/
- https://www.sisense.com/glossary/data-cleaning/
- https://www.analyticsvidhya.com/blog/2020/07/knnimputer-a-robust-way-to-impute-missing-values-using-scikit-learn/
- https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825
- https://www.import.io/post/business-data-analysis-what-how-why/
- https://www.itl.nist.gov/div898/handbook/eda/eda.htm
- https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15
- https://www.formpl.us/blog/data-interpretation
- https://searchbusinessanalytics.techtarget.com/definition/data-visualization
- https://www.tableau.com/learn/articles/data-visualization
- https://web.archive.org/web/20201101004343/https://machinelearningmastery.com/logistic-regression-for-machine-learning/
- http://faculty.cas.usf.edu/mbrannick/regression/Logistic.html
- https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Logistic_Regression.pdf
- https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/
- https://realpython.com/logistic-regression-python/
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://numpy.org/doc/stable/reference/constants.html?highlight=nan#numpy.NAN
- https://datatofish.com/message-box-python/
- https://stackabuse.com/decision-trees-in-python-with-scikit-learn/
- https://dataaspirant.com/decision-tree-algorithm-python-with-scikit-learn/
- https://pandas.pydata.org/docs/reference/api/pandas.Series.to_list.html?highlight=to_list#pandas.Series.to_list
- https://www.cin.ufpe.br/~cavmj/Machine%20-%20Learning%20-%20Tom%20Mitchell.pdf

- https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-dummy-coding/
- https://mljar.com/blog/visualize-decision-tree/
- https://www.ke.tu-darmstadt.de/lehre/archiv/ws0809/mldm/dt.pdf
- https://www.askpython.com/python/examples/standardize-data-in-python
- https://scikit-learn.org/stable/modules/preprocessing.html
- https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html
- https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/
- https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/
- https://docs.python.org/3/library/tkinter.ttk.html?highlight=checkbutton#standard-options
- https://web.archive.org/web/20210407233302/https://docs.python.org/3/library/io.html
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html?highlight=decisiontreeclassifier#sklearn.tree.DecisionTreeClassifier.predict
- https://web.archive.org/web/20210122100556/https://scikit-learn.org/stable/modules/generated/sklearn.metrics.ConfusionMatrixDisplay.html#sklearn.metrics.ConfusionMatrixDisplay
- https://scikit-learn.org/stable/modules/generated/sklearn.tree.plot_tree.html#sklearn.tree.plot_tree
- https://datascience.stackexchange.com/questions/28512/train-new-data-to-pre-trained-model

* * *

Complete collection of the project files is safely kept at

https://github.com/WolfDev8675/RepoSJX7/tree/Assign3_1

* * *