A PROJECT ON E-MAIL MARKETTING AND DATA PROPENSITY MODELLING

Project submitted and prepared by Bishal Biswas
under guidance of Prof. Arghya Roy

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. Arghya Roy

Acknowledgement

Our project and everything started during the ending rule of SARS – CoVID19, virtually crippling the society and world as a whole sending everything into a lockdown, although at better stage but still this 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. Arghya Roy for guiding with every step from imparting knowledge about the subject to the intricacies of the Data analytics, clearing doubts and issues faced.

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.
PGDDSPJULY2020/1
b.biswas_94587@ieee.org

Contents

Objective and Purpose	4
Introduction	5
Data Analytics	6
Why Data Analytics?	6
Why Data Analytics Matter	7
Types of Data Analytics	7
Applications of Data Analytics	8
Data Analysis Tools	9
Data Analysis Techniques	10
A little intro to the problem at hand	13
Logistic Regression	13
The Logit and Logistic Transformations	14
The Log Odds Ratio Transformation	15
The Logistic Regression and Logit Models	15
Solving the Likelihood Equations	17
Interpretation of Regression Coefficients	18
Apache Spark	20
Problem statement	23
Interpretation	25
PySpark Code for Pre-Processing	27
Example of the data	28
Equation of Logit	29
Sigmoid obtained	29
Final metrics	30
Target reached from Test data with the Objective at hand	32
Conclusion and Inferences	32
Bibliography	33

Objective and Purpose

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. According to estimates, in 2021, there will be 74 zetabytes of generated data. That's expected to double by 2024.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.

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 Analytics and subsequently Big Data Analytics is based on.

Data Analytics

Data analytics is the science of analyzing raw data in order to make conclusions about that information. Many of the techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.

Data analytics techniques can reveal trends and metrics that would otherwise be lost in the mass of information. This information can then be used to optimize processes to increase the overall efficiency of a business or system.

Why Data Analytics?

Data analytics is a broad term that encompasses many diverse types of data analysis. Any type of information can be subjected to data analytics techniques to get insight that can be used to improve things.

For example, manufacturing companies often record the runtime, downtime, and work queue for various machines and then analyze the data to better plan the workloads so the machines operate closer to peak capacity.

Data analytics can do much more than point out bottlenecks in production. Gaming companies use data analytics to set reward schedules for players that keep the majority of players active in the game. Content companies use many of the same data analytics to keep you clicking, watching, or re-organizing content to get another view or another click.

The process involved in data analysis involves several different steps:

1. The first step is to determine the data requirements or how the data is grouped. Data may be separated by age, demographic, income, or gender. Data values may be numerical or be divided by category.

2. The second step in data analytics is the process of collecting it. This can be done through a variety of sources such as computers, online sources, cameras, environmental sources, or through personnel.
3. Once the data is collected, it must be organized so it can be analyzed. Organization may take place on a spreadsheet or other form of software that can take statistical data.
4. The data is then cleaned up before analysis. This means it is scrubbed and checked to ensure there is no duplication or error, and that it is not incomplete. This step helps correct any errors before it goes on to a data analyst to be analyzed.

Hence, key takeaways from the definition of Data Analytics can be summarized into:

- Data analytics is the science of analyzing raw data in order to make conclusions about that information.
- The techniques and processes of data analytics have been automated into mechanical processes and algorithms that work over raw data for human consumption.
- Data analytics help a business optimize its performance.

Why Data Analytics Matters

Data analytics is important because it helps businesses optimize their performances. Implementing it into the business model means companies can help reduce costs by identifying more efficient ways of doing business and by storing large amounts of data.

A company can also use data analytics to make better business decisions and help analyze customer trends and satisfaction, which can lead to new—and better—products and services.

Types of Data Analytics

Data analytics is broken down into four basic types.

- Descriptive analytics describes what has happened over a given period of time.
    - Examples:
        - Have the number of views gone up?
        - Are sales stronger this month than last?

- Diagnostic analytics focuses more on why something happened. This involves more diverse data inputs and a bit of hypothesizing.
    - Examples:
        - Did the weather affect beer sales?
        - Did that latest marketing campaign impact sales?
- Predictive analytics moves to what is likely going to happen in the near term.
    - Examples:
        - What happened to sales the last time we had a hot summer?
        - How many weather models predict a hot summer this year?
- Prescriptive analytics suggests a course of action.
    - Examples:
        - If the likelihood of a hot summer is measured as an average of these five weather models is above 58%, we should add an evening shift to the brewery and rent an additional tank to increase output.

Data analytics underpins many quality control systems in the financial world, including the ever-popular Six Sigma program. If you aren't properly measuring something—whether it's your weight or the number of defects per million in a production line—it is nearly impossible to optimize it.

Applications of Data Analytics

- Healthcare

  The main challenge for hospitals is to treat as many patients as they efficiently can, while also providing a high. Instrument and machine data are increasingly being used to track and optimize patient flow, treatment, and equipment used in hospitals. It is estimated that there will be a one percent efficiency gain that could yield more than $63 billion in global healthcare savings by leveraging software from data analytics companies.
- Travel

  Data analytics can optimize the buying experience through mobile/weblog and social media data analysis. Travel websites can gain insights into the customer's preferences. Products can be upsold by correlating current sales to the subsequent browsing increase in browse-to-buy conversions via customized packages and offers. Data analytics that is based on social media data can also deliver personalized travel recommendations.
- Gaming

  Data analytics helps in collecting data to optimize and spend within and across games. Gaming companies are also able to learn more about what their users like and dislike.

- Energy Management

  Most firms are using data analytics for energy management, including smart-grid management, energy optimization, energy distribution, and building automation in utility companies. The application here is centered on the controlling and monitoring of network devices and dispatch crews, as well as managing service outages. Utilities have the ability to integrate millions of data points in the network performance and gives engineers the opportunity to use the analytics to monitor the network.

Data Analysis Tools

There are several data analysis tools available in the market, each with its own set of functions. The selection of tools should always be based on the type of analysis performed, and the type of data worked. Here is a list of a few compelling tools for Data Analysis.

1. Excel

It has a variety of compelling features, and with additional plugins installed, it can handle a massive amount of data. So, if you have data that does not come near the significant data margin, then Excel can be a very versatile tool for data analysis.

2. Tableau

It falls under the BI Tool category, made for the sole purpose of data analysis. The essence of Tableau is the Pivot Table and Pivot Chart and works towards representing data in the most user-friendly way. It additionally has a data cleaning feature along with brilliant analytical functions.

3. Power BI

It initially started as a plugin for Excel, but later on, detached from it to develop in one of the most data analytics tools. It comes in three versions: Free, Pro, and Premium. Its PowerPivot and DAX language can implement sophisticated advanced analytics similar to writing Excel formulas.

4. Fine Report

Fine Report comes with a straightforward drag and drops operation, which helps to design various styles of reports and build a data decision analysis system. It can directly connect to all kinds of databases, and its format is similar to that of Excel. Additionally, it also provides a variety of dashboard templates and several self-developed visual plug-in libraries.

5. R & Python

These are programming languages which are very powerful and flexible. R is best at statistical analysis, such as normal distribution, cluster classification algorithms, and regression analysis. It also performs individual predictive analysis like customer behavior, his spend, items preferred by him based on his browsing history, and more. It also involves concepts of machine learning and artificial intelligence.

6. SAS

It is a programming language for data analytics and data manipulation, which can easily access data from any source. SAS has introduced a broad set of customer profiling products for web, social media, and marketing analytics. It can predict their behaviors, manage, and optimize communications.

Data Analysis Techniques

There are different techniques for Data Analysis depending upon the question at hand, the type of data, and the amount of data gathered. Each focuses on strategies of taking onto the new data, mining insights, and drilling down into the information to transform facts and figures into decision making parameters. Accordingly, the different techniques of data analysis can be categorized as follows:

1. Techniques based on Mathematics and Statistics
    - Descriptive Analysis: Descriptive Analysis takes into account the historical data, Key Performance Indicators, and describes the performance based on a chosen benchmark. It takes into account past trends and how they might influence future performance.
    - Dispersion Analysis: Dispersion in the area onto which a data set is spread. This technique allows data analysts to determine the variability of the factors under study.
    - Regression Analysis: This technique works by modeling the relationship between a dependent variable and one or more independent variables. A regression model can be linear, multiple, logistic, ridge, non-linear, life data, and more.
    - Factor Analysis: This technique helps to determine if there exists any relationship between a set of variables. In this process, it reveals other factors or variables that describe the patterns in the relationship among the original variables. Factor Analysis leaps forward into useful clustering and classification procedures.
    - Discriminant Analysis: It is a classification technique in data mining. It identifies the different points on different groups based on variable

measurements. In simple terms, it identifies what makes two groups different from one another; this helps to identify new items.

- Time Series Analysis: In this kind of analysis, measurements are spanned across time, which gives us a collection of organized data known as time-series.

2. Techniques based on Artificial Intelligence and Machine Learning
   - Artificial Neural Networks: a Neural network is a biologically-inspired programming paradigm that presents a brain metaphor for processing information. An Artificial Neural Network is a system that changes its structure based on information that flows through the network. ANN can accept noisy data and are highly accurate. They can be considered highly dependable in business classification and forecasting applications.
   - Decision Trees: As the name stands, it is a tree-shaped model that represents a classification or regression models. It divides a data set in smaller subsets simultaneously developing into a related decision tree.
   - Evolutionary Programming: This technique combines the different types of data analysis using evolutionary algorithms. It is a domain-independent technique, which can explore ample search space and manages attribute interaction very efficiently.
   - Fuzzy Logic: It is a data analysis technique based on probability which helps in handling the uncertainties in data mining techniques.

3. Techniques based on Visualization and Graphs
   - Column Chart, Bar Chart: Both these charts are used to present numerical differences between categories. The column chart takes to the height of the columns to reflect the differences. Axes interchange in the case of the bar chart.
   - Line Chart: This chart is used to represent the change of data over a continuous interval of time.
   - Area Chart: This concept is based on the line chart. It additionally fills the area between the polyline and the axis with color, thus representing better trend information.
   - Pie Chart: It is used to represent the proportion of different classifications. It is only suitable for only one series of data. However, it can be made multi-layered to represent the proportion of data in different categories.
   - Funnel Chart: This chart represents the proportion of each stage and reflects the size of each module. It helps in comparing rankings.

- Word Cloud Chart: It is a visual representation of text data. It requires a large amount of data, and the degree of discrimination needs to be high for users to perceive the most prominent one. It is not a very accurate analytical technique.
- Gantt Chart: It shows the actual timing and the progress of activity in comparison to the requirements.
- Radar Chart: It is used to compare multiple quantized charts. It represents which variables in the data have higher values and which have lower values. A radar chart is used for comparing classification and series along with proportional representation.
- Scatter Plot: It shows the distribution of variables in the form of points over a rectangular coordinate system. The distribution in the data points can reveal the correlation between the variables.
- Bubble Chart: It is a variation of the scatter plot. Here, in addition to the x and y coordinates, the area of the bubble represents the 3rd value.
- Gauge: It is a kind of materialized chart. Here the scale represents the metric, and the pointer represents the dimension. It is a suitable technique to represent interval comparisons.
- Frame Diagram: It is a visual representation of a hierarchy in the form of an inverted tree structure.
- Rectangular Tree Diagram: This technique is used to represent hierarchical relationships but at the same level. It makes efficient use of space and represents the proportion represented by each rectangular area.
- Map
    - Regional Map: It uses color to represent value distribution over a map partition.
    - Point Map: It represents the geographical distribution of data in the form of points on a geographical background. When the points are the same in size, it becomes meaningless for single data, but if the points are as a bubble, then it additionally represents the size of the data in each region.
    - Flow Map: It represents the relationship between an inflow area and an outflow area. It represents a line connecting the geometric centers of gravity of the spatial elements. The use of dynamic flow lines helps reduce visual clutter.
    - Heat Map: This represents the weight of each point in a geographic area. The color here represents the density.

A little intro to the problem at hand

A sales data of an UK based company is provided for analysis and target was to determine at the final stage potential customers identified by their ids' who are likely to buy if any new sales or renewed information of their purchase is pitched on them. Data of this establishment is obtained from Kaggle.com and modified via python codes to a state where statistical analysis to the target requirement is obtainable. The partial solution obtained from the Python code pertaining to Apache Spark ecosystem and related libraries (viz., Pyspark) revealed the data in such a way where we needed to regress logistically to the problem. Since our final requirement stands at a point where we either decide if to mail the customer or not to mail the customer.

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

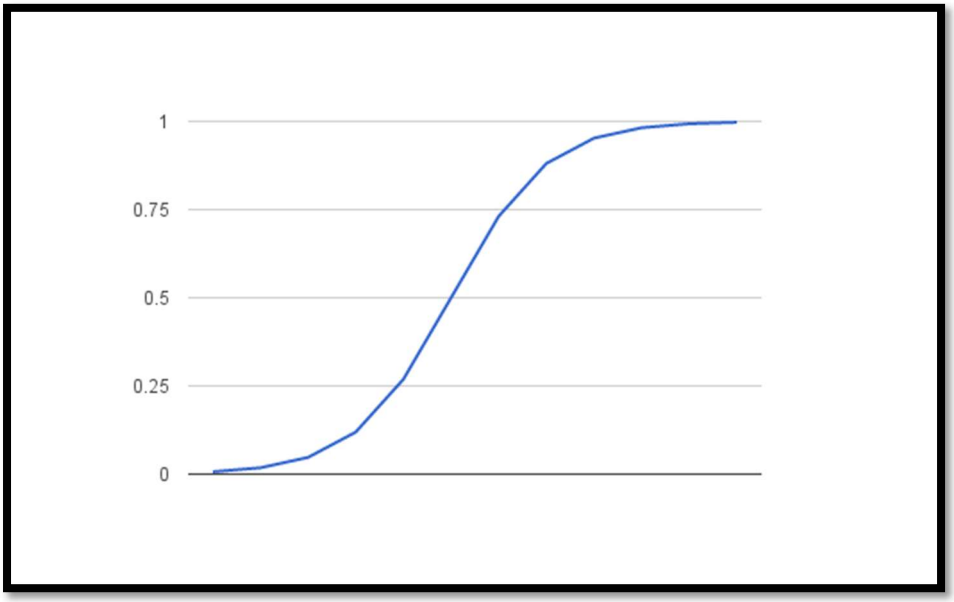
In multiple regression, a mathematical model of a set of explanatory variables is used to predict the mean of a continuous dependent variable. In logistic regression, a mathematical model of a set of explanatory variables is used to predict a logit transformation of the dependent variable. Suppose the 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

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

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,

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

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

The logistic regression model is given by the G equations

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

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

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

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

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

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

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

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

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

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

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

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

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,

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

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

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

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.

Apache Spark

Apache Spark is a unified analytics engine for large-scale data processing. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, MLlib for machine learning, GraphX for graph processing, and Structured Streaming for incremental computation and stream processing.

Apache Spark has its architectural foundation in the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way. The Dataframe API was released as an abstraction on top of the RDD, followed by the Dataset API. In Spark 1.x, the RDD was the primary application programming interface (API), but as of Spark 2.x use of the Dataset API is encouraged even though the RDD API is not deprecated. The RDD technology still underlies the Dataset API.

Spark and its RDDs were developed in 2012 in response to limitations in the MapReduce cluster computing paradigm, which forces a particular linear dataflow structure on distributed programs: MapReduce programs read input data from disk, map a function across the data, reduce the results of the map, and store reduction results on disk. Spark's RDDs function as a working set for distributed programs that offers a (deliberately) restricted form of distributed shared memory.

Spark facilitates the implementation of both iterative algorithms, which visit their data set multiple times in a loop, and interactive/exploratory data analysis, i.e., the repeated database-style querying of data. The latency of such applications may be reduced by several orders of magnitude compared to Apache Hadoop MapReduce implementation. Among the class of iterative algorithms are the training algorithms for machine learning systems, which formed the initial impetus for developing Apache Spark.

Apache Spark requires a cluster manager and a distributed storage system. For cluster management, Spark supports standalone (native Spark cluster, where you can launch a cluster either manually or use the launch scripts provided by the install package. It is also possible to run these daemons on a single machine for testing), Hadoop YARN, Apache Mesos or Kubernetes. For distributed storage, Spark can interface with a wide variety, including Alluxio, Hadoop Distributed File System (HDFS), MapR File System (MapR-FS), Cassandra, OpenStack Swift, Amazon S3, Kudu, Lustre file system, or a custom solution can be implemented. Spark also supports a pseudo-distributed local mode, usually used only for development or testing purposes, where distributed storage is not required and the local file system can be used instead; in such a scenario, Spark is run on a single machine with one executor per CPU core.

Spark Core

Spark Core is the foundation of the overall project. It provides distributed task dispatching, scheduling, and basic I/O functionalities, exposed through an application programming interface (for Java, Python, Scala, .NET and R) centered on the RDD abstraction (the Java API is available for other JVM languages, but is also usable for some other non-JVM languages that can connect to the JVM, such as Julia). This interface mirrors a functional/higher-order model of programming: a "driver" program invokes parallel operations such as map, filter or reduce on an RDD by passing a function to Spark, which then schedules the function's execution in parallel on the cluster. These operations, and additional ones such as joins, take RDDs as input and produce new RDDs. RDDs are immutable and their operations are lazy; fault-tolerance is achieved by keeping track of the "lineage" of each RDD (the sequence of operations that produced it) so that it can be reconstructed in the case of data loss. RDDs can contain any type of Python, .NET, Java, or Scala objects.

Besides the RDD-oriented functional style of programming, Spark provides two restricted forms of shared variables: broadcast variables reference read-only data that needs to be available on all nodes, while accumulators can be used to program reductions in an imperative style.

Spark SQL

Spark SQL is a component on top of Spark Core that introduced a data abstraction called DataFrames, which provides support for structured and semi-structured data. Spark SQL provides a domain-specific language (DSL) to manipulate DataFrames in Scala, Java, Python or .NET. It also provides SQL language support, with command-line interfaces and ODBC/JDBC server.

Although DataFrames lack the compile-time type-checking afforded by RDDs, as of Spark 2.0, the strongly typed DataSet is fully supported by Spark SQL as well.

Spark Streaming

Spark Streaming uses Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD transformations on those mini-batches of data. This design enables the same set of application code written for batch analytics to be used in streaming analytics, thus facilitating easy implementation of lambda architecture. However, this convenience comes with the penalty of latency equal to the mini-batch duration. Other streaming data engines that process event by event rather than in mini-batches include Storm and the streaming component of Flink. Spark Streaming has support built-in to consume from Kafka, Flume, Twitter, ZeroMQ, Kinesis, and TCP/IP sockets.

In Spark 2.x, a separate technology based on Datasets, called Structured Streaming, that has a higher-level interface is also provided to support streaming. Spark can be deployed in a traditional on-premises data center as well as in the cloud.

MLlib Machine Learning Library

Spark MLlib is a distributed machine-learning framework on top of Spark Core that, due in large part to the distributed memory-based Spark architecture, is as much as nine times as fast as the disk-based implementation used by Apache Mahout (according to benchmarks done by the MLlib developers against the alternating least squares (ALS) implementations, and before Mahout itself gained a Spark interface), and scales better than Vowpal Wabbit. Many common machine learning and statistical algorithms have been implemented and are shipped with MLlib which simplifies large scale machine learning pipelines, including:

- summary statistics, correlations, stratified sampling, hypothesis testing, random data generation
- classification and regression: support vector machines, logistic regression, linear regression, naive Bayes classification, Decision Tree, Random Forest, Gradient-Boosted Tree
- collaborative filtering techniques including alternating least squares (ALS)
- cluster analysis methods including k-means, and latent Dirichlet allocation (LDA)
- dimensionality reduction techniques such as singular value decomposition (SVD), and principal component analysis (PCA)
- feature extraction and transformation functions

- optimization algorithms such as stochastic gradient descent, limited-memory BFGS (L-BFGS)

GraphX

GraphX is a distributed graph-processing framework on top of Apache Spark. Because it is based on RDDs, which are immutable, graphs are immutable and thus GraphX is unsuitable for graphs that need to be updated, let alone in a transactional manner like a graph database.[26] GraphX provides two separate APIs for implementation of massively parallel algorithms (such as PageRank): a Pregel abstraction, and a more general MapReduce-style API.[27] Unlike its predecessor Bagel, which was formally deprecated in Spark 1.6, GraphX has full support for property graphs (graphs where properties can be attached to edges and vertices).

GraphX can be viewed as being the Spark in-memory version of Apache Giraph, which utilized Hadoop disk-based MapReduce. Like Apache Spark, GraphX initially started as a research project at UC Berkeley's AMPLab and Databricks, and was later donated to the Apache Software Foundation and the Spark project.

Pyspark

Apache Spark is written in Scala programming language. PySpark has been released in order to support the collaboration of Apache Spark and Python, it actually is a Python API for Spark. In addition, PySpark, helps you interface with Resilient Distributed Datasets (RDDs) in Apache Spark and Python programming language. This has been achieved by taking advantage of the Py4j library. Py4J is a popular library which is integrated within PySpark and allows python to dynamically interface with JVM objects. PySpark features quite a few libraries for writing efficient programs.

Problem statement

The Problem statement as was provided:

```
Project 1: Email Marketing / Retail Propensity Modelling

Data Set: OnlineRetail as received from Kaggle

Data Length: 10,48,575.

Fields to be considered: CustID, Quantity, Price, Invoice
Date, Country and Stock Code.
```

- X variables: Count, Recency, Quantity, Price, Country

- Y variable: IF(Count > 2,1,0)

Problem Statement:

Find out the probability of buying a specific stock on 15-April-2011 for a Customer. If the probability is

greater than 0.8, then we email the Customer.

Workaround:

Build the Logistics Regression model on data of 4147 Customers. Freeze the coefficients.

Use those coefficients to predict Y for remaining 1777 Customers.

Expected Output:

1. Performance Metrics of the Model.

2. Based on recency, who are the hot prospects. (out of 1777 Customers)

3. Based on probability, create 5 group of prospects who are to be targeted with Group 1 should

consists of highest probability prospects. (On 1777 Customers)

. . .

Additional Clarifications and changes

1. Sort of recency of test data => Low to high => predicted probabilities high to low

2. Divide by 5 groups G to R color coded

* . * . * . *

** G to R is color green to red in a gradient method

Interpretation

Pulling in ideas as was imparted in the brief intro into problem as well as the problem statement itself. We followed the following steps to workout our way to the end solution of our problem.

PySpark Pre-solution.

1. Data found to have discrepancies in irregularities.
2. So primarily we needed filtering.
3. Process done as a simple check that fields weren't empty or irregular valued.
4. Those with such problems were promptly scrapped.
5. We were also briefed about working on customers who were buying a particular stock "85123A".
6. Hence, we also added this stock code into our filtering code.
7. Next to the cleaning we needed to set up a 'Y' variable which was specifically pointed to work on 'count' and 'gt' values if they were greater than '2' then we switch the resultant Y as equal to 1(numerical/categorical-boolean) if this condition isn't matched, we switch the resultant Y as equal to 0(numerical/categorical-boolean).
8. This is done using the "group-by" module of Spark on the cleaned data and thus a 'Y' variable was custom – made according to the problem shifting the solution to a stage where any decision if taken must be on logistical domains of either true or false values.

Logistical Regression Solution

1. The solution which stood as a result of the PySpark pre-process gave us a data of 1490 units or 1490 customers with sum-price of the item, total quantity of items, purchase-count of the customer with the seller and the particular stock, max -date of the recency of purchase, recency by time the customer has bought the item.
2. All these fields are debriefed as possible contenders in setting up and determining Y value. Also, we need a training set and a test set to determine the perfection of our model.
3. Conventionally we have 7:3 ratio for "train: test" breaking of the data units' collection, as for us,

   7:3 => 70%: 30%
   Therefore, for train we have,
   0.7*1489=1042.3,
   keeping 1490-1042.3=446.7 as the test data points
   but we have to finalize our test data into 5 groups
   where 446.7/5=89.34 which is fractional

Since we can't take a fractional part of a data field, viz., we either take it complete or don't take it at all into consideration.
The test data is finalized into 440 units where if grouped finally we can have 440/5=88 elements in each group mathematically
Thereby, for training we have 1489-440=1049 elements
Hence, we get a ratio of 1049/1489=0.704499 => 70.4499% for training and
440/1489=0.295500 => 30.5500% for testing which goes very closely according to conventional ratio for distribution of train to test data sets.
Thereby, concluding that our assumption of the intake numbers for each set is within limits of the rules.

4. By method of basic Logistic regression, we set up an equation with intercepts and coefficients alike at 0.0010 (decimal /float/double) to find logit, $e^{(logit)}$, sigmoid-p and the sum of sigmoid according to the formulas.
5. In the next step we used the Solver tool at 95% internal confidence to solve the sum of sigmoids to a max value by GRG Non-Linear method and preventing any changes to replace unconstrained variables as non-negative results.
6. Small errors in limits generated is corrected as a static value and the Solver tool is re-run to finer correction stages with the same settings.
7. The solution now obtained is taken as final and binding.
8. The changed coefficients is again used to find the Y predicted values which is easily obtained by using sigmoid-p value to either touch floor or ceiling depending on a 0.5 threshold.
9. The equation obtained in now used on the test data to find out the Y predicted values for the test data which is easily found out.
10. From the test data we copy the numerical values of the Logit and Sigmoid values separately as a table and sort the table in the ascending order by values of logit. Plotting this Sigmoid vs Logit, we get our own sigmoid curve.
11. Then on we calculate the metrics of our model namely the Confusion matrix and the other important values required to finalize our model.
12. Next and final step is the copy of the test data and we work on the additional changes to the problem description done and color coding the results to determine the most possible to least possible choices where a threshold value of 0.8 played a part, to determine suitability of mailing the customer.

PySpark Code for Pre-Processing

```
import org.apache.spark._

import org.apache.log4j._

import org.apache.spark.sql.Row

import org.apache.spark.sql.types._

import org.apache.spark.sql.functions._

import org.apache.spark.sql._

import org.apache.spark.sql._

object BSE_Project_Retail {

  case class RETAIL_SCHEMA(

Invoice:String, StockCode:String, Description:String, Quantity:Int,
InvoiceDate:String, Price:Double, Customer_ID:String, Country:String)

  def main(args: Array[String]) {

    Logger.getLogger("org").setLevel(Level.ERROR)

    val sc = new SparkContext("local[*]", "OnlineRetails")

    val spark =
SparkSession.builder.appName("OnlineRetail").master("local[*]").getOrCreat
e()

    import spark.implicits._

    val retail_ds = spark.read.option("header",
"true").option("inferSchema",
"true").csv("data/online_retail_headers.csv").cache().as[RETAIL_SCHEMA]

    val retail_fil = retail_ds.filter($"Price" >= 0 and $"Quantity" > 0
and $"StockCode" === "85123A" and

        $"Country".isNotNull and

        $"Invoice".isNotNull and

        $"StockCode".isNotNull and

        $"InvoiceDate".isNotNull and

        $"Customer_ID".isNotNull)

    val final_filtered_data = retail_fil.select($"Customer_ID",
$"Quantity", $"Price", $"InvoiceDate", $"Country", $"StockCode")



val group_by = final_filtered_data.groupBy($"Customer_ID").

                                    agg(sum("Price").as("SUM_PRICE"),
```

```
sum("Quantity").as("TOTAL_QUANTITY"),


count("Quantity").as("PURCHASE_COUNT"),

                                        max("InvoiceDate").as("MAX_DATE")).
withColumn("USER_DATE",to_timestamp(lit("20111204"),"yyyyMMdd")).


withColumn("mdate",date_format(unix_timestamp(col("MAX_DATE"),"dd-MM-yyyy
HH:mm").cast(TimestampType), "yyyy-MM-dd HH:mm:ss"))

val final_output = group_by.select($"Customer_ID", $"SUM_PRICE",
$"TOTAL_QUANTITY", $"PURCHASE_COUNT",$"MAX_DATE",
(datediff(col("USER_DATE"),col("mdate"))/365).as("RECENCY")).

                withColumn("Y", when(col("PURCHASE_COUNT") >
2,1).otherwise(0))

    final_output.show()

    print(final_output.count())

    final_output.repartition(1).write.format("csv").option("header",
"true").save("data/retail_final_filter_data")

  }

}


**End of codes...
```

Example of the data

Right after completing the Pyspark process code we get a data of the form as given below.

N.B.: this is just an example of the data hence the number of data points is limited to 10.

Customer_ID	SUM_PRICE	TOTAL_QUANTITY	PURCHASE_COUNT	MAX_DATE	RECENCY	Y
12940	2.95	6	1	13	0.224657534	0
13289	2.95	6	1	16	1.967123288	0
13623	17.7	16	6	24	1.02739726	1
15727	14.75	54	5	25	0.528767123	1
14423	7.65	256	3	28	1.183561644	1
14465	2.95	3	1	7	0.660273973	0
14514	20.65	112	7	29	1.180821918	1
14876	5.9	24	2	27	1.523287671	0
15003	5.9	4	2	26	1.854794521	0
15967	20.65	22	7	26	1.189041096	1

Equation of Logit

The equation of logit was primarily set as

Logit= 0.001+ 0.001*Sum_Price + 0.001*Total_Quantity + 0.001*Purchase_Count +0.001*Max_Date + 0.001*Recency

After the solver is run, we get the following equations at each stage:

Stage 1. Logit= 0.0008 + 0.0038*Sum_Price + 0.0138*Total_Quantity + 0.002*Purchase_Count - 0.0001*Max_Date + 0.0008*Recency

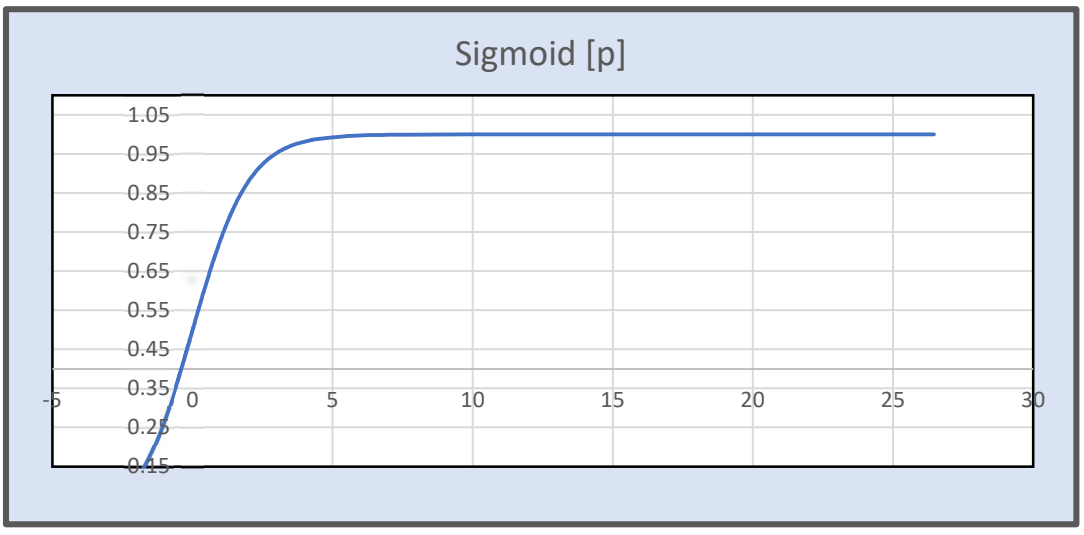
Stage 2. Logit= -0.3830 + 0.1418*Sum_Price + 0.0018*Total_Quantity + 0.0656*Purchase_Count - 0.0406*Max_Date - 0.4004*Recency

All solutions obtained and further processes performed on the test data as well as checking the train data were done using the equation at stage 2.

Sigmoid obtained

With the equation obtained at Stage 2 of Solver operations, we take the testing data form the logit and the corresponding sigmoid (p) value which in addition to deciding the solution (Y) also is a contender for the Sigmoid curve. To plot the curve, we followed the following processes:

1. Copy over values of the logit and the sigmoid columns to a newer sheet to prevent any data going bad due to sorting processes.
2. Sort the complete table in ascending order for the values only in the Logit column.
3. Plot a Linear Scatter curve for this sorted table with the Sigmoid as the Ordinate and Logit as the Abscissa giving us the following curve as given below



Final metrics

With the solution standing for itself, we first find out four variables separately after generating the confusion matrix

1. TP = True positive; cases where the model's prediction for a truth corroborates with the actual findings.
2. FP = False positive; cases where the model predicts for a truth, where in actuality the result was a failure.
3. TN = True negative; cases where the model's prediction for a fallacy corroborates with the actual findings.
4. FN = False negative; cases where the model predicts for a false, where in actuality the result was a truth.

And consequently, calculate the metrics of our model with the following formulas.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

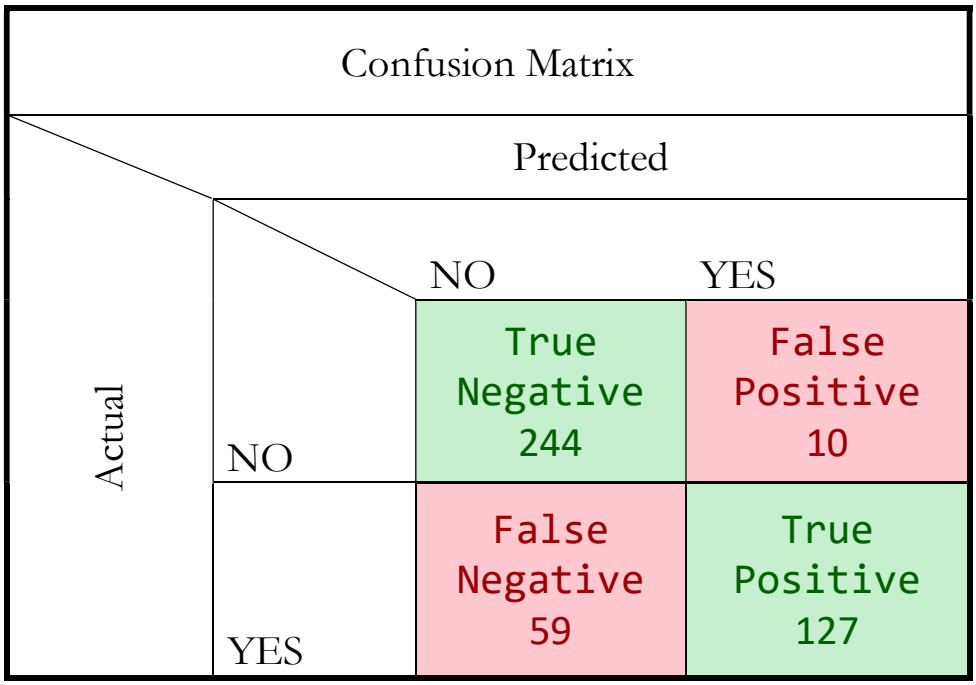
$$\text{Recall} = \frac{TP}{TP+FN}$$

$$\text{Sensitivity} = \frac{TP}{\text{Count of all the observed positives}}$$

$$\text{Specificity} = \frac{TN}{TN+FP}$$

$$\text{f1\_score} = \frac{2*TP}{2*TP+FP+FN}$$

Confusion matrix (Test data)



Metrics obtained

Accuracy	84%
Precision	93%
Recall	68%
Sensitivity	68%
Specificity	96%
f1_score	79%

Target reached from Test data with the Objective at hand

According to the preset rules set at the objective of the project problem the following results were obtained.

Mail Sending Count	
To be Sent	58
Not to be Sent	382
Total Count	440

Conclusion and Inferences

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

1. There is scope for a little development since we have accuracy at 84%, we could do things to raise the accuracy to the 90s
2. The test data had lesser true positives than true negatives, that gives us an idea for development by taking more data points in the test or by overall increase the size of dataset taken.
3. The sigmoid obtained must have points which are completely on the zero line instead we had points plunging below the zero, signifying either of the two options:
    a. We need more iterations of solver (only 2 stages were done – it needs to be increased to at least 30 for manual calculations). Best solutions were obtained at 10000 to around a million iterations levels
    b. Data is skewed at many places in spite of corrections and filtering in the preprocess at the PySpark code.
4. Some data fields which were custom generated could have been more accurate if taken at the collection level than being generated using simple mathematical calculations; maybe complex equations may have been involved to generate results closer to life.
5. This problem could have been solved using python language (** was forbidden for this project) using the "Pandas" package, where we could have more solver stage iterations at around 10,000,000 generating more correct results.

Bibliography

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

- 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.zarantech.com/blog/importance-of-data-science/
- https://www.import.io/post/business-data-analysis-what-how-why/
- https://towardsdatascience.com/a-brief-introduction-to-pyspark-ff4284701873
- https://spark.apache.org/docs/latest/api/python/
- https://pypi.org/project/pyspark/
- https://databricks.com/glossary/pyspark
- 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/

* * *

Complete collection of the project files is safely kept at

https://github.com/WolfDev8675/RepoSJX7/tree/Assign2_1

* * *