# Chapter 3

# Binary Classification

# Introduction

You were introduced to two broad categories of machine learning. One of them is supervised learning and the other is “unsupervised learning”. Supervised learning is further divided into two types of problem cases:

* Regression
* Classification.

The previous chapter was a brief introduction to “Regression”. This chapter is revolving around another concept, “Classification”. Binary classification involves the task of classifying the aspects of a given set into two groups based on a classification rule. While performing Binary Classification, categories are predefined and it is utilized to categorize new probabilistic observations into said categories.

We face many real-world problems that involve Classifications at various levels. Unlike regression problems, where a real numbered value is predicted, classification problems deal with the association of an example to a category. They take the forms such as:

* Predicting whether an email is spam or not.
* Identifying images of single digits 0 – 9 correctly.
* Identification of a satellite image into man-made or natural regions.
* Identifying disease or tissue types based on the gene expression levels.
* Predicting whether a customer will buy a certain product or not.
* Identifying whether a credit transaction is fraudulent.

From the above-mentioned examples, 1,5, and 6 are those examples that involve binary decisions.

Such problems are known as “**Binary Classification Problems**”. The other examples stated above deal with the multiple classes or categories. Such problems are called “**Multiclass classification problems**”, which would be discussed later.

### Comprehending the Business Context:

Business context is essential for solving purposes and for classifying the data upon which the context is based as it will help in arriving at various trails that can be utilized for exploratory analysis. Let us explore the data related to a Bank Note Authentication.

# Exercise 3.01

The Dataset “BankNote Authentication” will be loaded into the Jupyter Lab notebook, and the basic explorations are to be done on it. The printing of the dataset using the .shape() function and generating the summary of the dataset using the .describe() function are seen here.

Open the following link to get started:

<https://bit.ly/3MQTNTH>

For your practice, an end-of-chapter exercise is given. Practice these concepts there.

# Activity 3.01 - Hypothesis Testing

Based upon the practice exercises in this chapter, you are required to perform activities. Activity 01 is related to a general phase of Hypothesis testing.

Open the following link to get started:

<https://bit.ly/3MZbOPS>

# Testing Business Hypothesis Using Exploratory Data Analysis

Here we are at the stage to explore the dataset in terms of authentication of the bank notes whether they are fraudulent or authentic. So, you would be able to approach the problem statement from a domain perspective. Evolving some hypotheses about the relationship of data related to the notes and the outcomes you have set to achieve are some of the factors to be visualized here. Such hypotheses need to be verified through the data you have. This is where “Exploratory Data Analysis” plays a big role in the data science life cycle.

# Visualization for Exploratory Data Analysis

Effective visualization is essential for deriving business intuitions from the data. In this section, we will introduce some of the visualization techniques that will be utilized for EDA:

* Line Graphs
* Histograms
* Density Plots
* Stacked bar Charts

### Line Graphs

Line graphs are the most common and preferred method for revealing data trends. They are mostly used for continuous data.

### Histograms

The proportion of data along with some specified intervals can be seen in the histogram. It represents the plots of the distributions of data. Histograms are very effective for identifying whether data distribution is symmetric and for identifying outliers in data.

### Density Plots

Density plots are utilized for visualizing the distribution of data. But they represent a smoother distribution.

### Stacked bar Charts

They help you to visualize the various classes of data, one on top of the other, to give you a sense of the proportion of the categories.

Further exercises will let you practice such visuals with data distribution of “BankNote Authentication”.

# Exercise 3.02

We are going to define a hypothesis to check the authentication of notes against the collected skewness. We will be using a line graph for the same purpose. First of all, we need to state the hypothesis about the relationship. A hypothesis can be domain knowledge. Let’s define our hypothesis for this scenario as: *The authentication of a note being real is more with less skewness*. This is our hypothesis. There is a target variable in data, known as “Class”. A class contains two values 0 representing a genuine note and 1 representing a fake note.

It is time to verify the veracity of the hypothesis with data. One of the best ways is by taking the cross-sections of data and visualizing them. It can be visualized in this exercise.

Open the following link to get started:

<https://bit.ly/34La8It>

# Feature Engineering

The process of EDA was traversed in the previous section. In this section, “Feature Engineering” is traversed. It is the process of transforming raw variables to generate new variables in the data. This process is one of the most important steps that influence the accuracy of the models that we build.

# Types

* Raw variables are transformed based on intuitions from a business perspective.
* The transformation of raw variables is done from a statistical and data normalization perspective.

# Business-driven Feature Engineering

It is the process of transforming raw variables based on business intuitions that were derived during the exploratory analysis. It covers transforming data and creating new variables based on business factors that influence a business problem.

Exploratory analysis was involved in previous exercises as the relationship of a single variable with the dependent variable was explored. In the upcoming exercise, we will combine multiple variables and then drive new features from them.

The task is divided into 2 exercises, depending upon the relationship between “skewness” and the class of note along with transforming individual variables and then combining them to form a new feature.

# Exercise 3.03

In this exercise, we have explored the relationship between two variables, skewness of a note and whether a note is real or fake. The steps which are followed to perform this can be accessed from exercise 3.03.

Open this link to get started:

<https://bit.ly/361SKjh>

# Exercise 3.04

In this exercise, we will combine the individual variables analyzed in Exercise 3.02 to derive a new feature called an “Index”. Weights have been assigned in terms of transformed skewness. The same can be practiced through this exercise.

Open the link to get started:

<https://bit.ly/3waDZVN>

# Data-driven Feature Engineering

This section involves the transformation of data through feature engineering from the perspective of data structures. Different methods of identifying data structures will be seen here.

Data types, such as categorical or numeric would be seen, and then deriving summary statistics is a good way to take a quick peek into data before you do some of the downstream feature engineering steps. The following example can be seen from our data set:

print(notes\_Data. types)

print(notes\_Data.describe())

Similarly, the descriptive summary statistic can be seen. It displays some of the basic features such as mean, standard deviation, count, etc. The main aim of a descriptive summary is to acquire a complete description of the data concerning the distribution and some basic statistics.

### Correlation Matrix and Visualization

Correlation is a measure that involves how one set of data may correspond to another set. It tells how one or more variables are related to each other. It is a statistical technique that determines how one variable changes about the other variable. Highly correlated variables can sometimes be damaging for the veracity of models and, in many cases, we make decisions to eliminate such variables or to combine them to form composite variables.

Let us visualize how data correlation can be generated in the upcoming exercise.

# Exercise 3.05

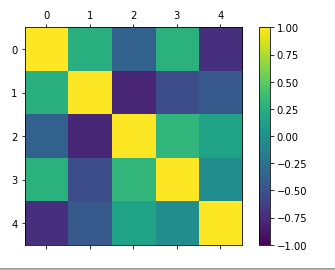
In this exercise, we will be generating a correlation plot based upon the provided dataset. We will be analyzing the results from them. We have to import the set\_option library from the “pandas” of Python. The set\_option function is used to define the display options for many operations. Following syntax is to be used for the same:

from pandas import set\_option

Open the following link to get started:

<https://bit.ly/3KJkg3G>

You should get the following output after solving the exercise:



### Skewness of Data

Feature engineering involves “skewness”. Skewed data means data is shifted in one direction or the other. It can cause machine learning models to underperform. Many models assume normally distributed data to follow the Gaussian structure. Any deviation from the assumed structure can affect model performance.

An area where feature engineering can be applied is by looking at the skewness of data and then correcting the skewness through normalization of the data. Skewness can be visualized by plotting the data using histograms and density plots. Let us then look into these techniques.

Any numeric data can be skewed through the following syntax:

numeric.skew()

### Histograms

The distribution of data can be plotted through the histogram and the skewness can be identified from the same. The histogram is plotted with the ***pyplot*** package from **matplotlib** using the .**hist()** function. Moreover, the number of subplots can also be included and are controlled by the .**subplots**() function. The titles are set by the ***set\_title()*** function.

### Density Plots

The visualization of data distribution can be achieved from Density Plots. Density plots can be created using the **kind = 'density'** parameter. They help in obtaining a smoother visualization of the data distribution.

# Other Feature Engineering Methods

We have looked at many methods for transformations of data so far. It is the right time to look into other similar data transformation techniques, namely **scaler, and normalizer.**

**Standard Scaler:** It standardizes data to a mean of 0 and a standard deviation of 1. The mean here is the average of the data and the standard deviation is a measure of the spread of data.

**Normalizer Function:** It normalizes the length of data, which means that each value in a row is divided by the normalization of the row vector to normalize the row. This function is applied on the rows while the standard scaler is applied columnwise.

The following code uses the normalizer data transmission techniques:

from sklearn.preprocessing import Normalizer

normaliser = Normalizer().fit(<*any numeric data*>)

normalisedNum = normaliser.transform(<*any numeric data*>) )

set\_printoptions(precision = 3)

print(normalisedNum)

# Building a Binary Classification Model using the Logistic Regression Function

In the previous section, we have seen how we do the essential transformation on the data elements. The right transformation of the data elements can highly influence the creation of the right business outcomes by the downstream modeling process. The learning of the traits is achieved through machine learning. You have realized in previous chapters as well that the goal of machine learning is to estimate a mapping function between an output variable and input variables. In mathematical terms, it can be written as:

**Y = f(X)**

In this equation, “Y" is a dependent variable, which is our prediction as to whether a banknote is real or fake

X is the independent variable that includes skewness, curtosis, and entropy.

f() is a function that connects various attributes of the data to the probability or whether a note is real or fake. This function is learned during the machine learning process. It is a combination of different coefficients applied to the attributes to get the probability of authenticity of a banknote.

Let us assume that we are having two attributes, namely “*variance*” and *skewness.* Using these, we have to identify whether a note is real or not. Let the variance be **20.1** and the skewness is **5.8**. With these attribute values, let us assume that the mapping equation is as follows:

**Y = M0+M1variance \*Variance +M2skewness\*skewness**

Based on certain assumptions related to “Y”, it can be said that the note is real or not. There is another term called “Model”. A model is an approximation of the data generation process. During machine learning, we are trying to get as close to the real model that has generated the data we are analyzing. To learn or estimate the data generating models, we use different machine learning algorithms. Machine learning models can be divided into two types:

* Parametric Models
* Non-parametric models

Parametric models are those where we assume the form of the function we are trying to learn and then learn the coefficients from the training data. Some examples of Parametric models include:

* Linear and logistic regression
* Naïve Bayes
* Linear support vector machines
* Perceptron

Those machine learning models that do not make strong assumptions on the function are called non-parametric models. Non-parametric models are free to learn any functional form from the data in the absence of an assumed form. Some examples are:

* Decision trees
* K –nearest neighbors
* Neural networks
* Support vector machines with Gaussian kernels

# Logistic regression Demystified:

It is similar to linear regression. At the core of logistic regression is the sigmoid function, which quashes any real-valued number to a value between 0 and 1, which renders this function ideal for predicting probabilities.

# Metrics for evaluating Model Performance

Based on the various metrics on the predictions, decisions are made and evaluations are done on the models you build. Some of the important metrics used for evaluating the performance of models are:

**Confusion Matrix:** A confusion matrix is built to generate the resultant comparison between prediction and the ground truth. The generation of a confusion matrix is used for calculating many of the matrices such as **accuracy** and **classification** **reports**. It is based on metrics such as accuracy or other detailed metrics shown in the classification report such as precision or recall of the models for testing.

Accuracy: Accuracy is the level of evaluation, which will be resorted to have a quich=k check on model performance.Accuracy can be represented as follows:

**Accuracy = T P + T N/ (T P + T N + F P + F N )**

Where

TP = True Positives,

TN = True Negatives,

FP = False Positives, and

FN = False Negatives.

Classification Report: A classification report gives us three key metrics: **precision, recall,** and the **F1 score**. **Precision** is the ratio of true positives to the sum of true positives and false positives:

**Precision = tp/(tp+fp)**

**The recall** is the ratio of true positives to the sum of true positives and false negatives:

Recall = tp/(tp+fn)

There is another measure, known as the F1 score. It is a weighted score of both precision and recall. An F1 score of 1 indicates the best performance and 0 indicates the worst performance.

# Data Preprocessing

The preprocessing steps include the following:

**Data Loading:** The loading of the data from different sources into the notebook**.**

**Data Cleaning:** It involves the removal of anomalies from the data.

**Data Imputation:** Filling the missing data with new data points.

**Converting data types:** Datasets will have different types of data such as numerical data, categorical data, and character data. Running models will necessitate the transformation of data types.

# Exercise 3.06

In this exercise, we will build a logistic regression model for predicting the note validity. This exercise would be having three parts. Preprocessing of data, then the training process, and then the prediction, analysis of metrics will be covered here:

Open the link to get started:

<https://bit.ly/3i9GlvZ>

# Activity 3.02 - Logistic Regression Model with Feature Engineered Variables

This activity is based upon Logistic Regression. Once the new features are created using some of the categorical variables, we don't have to use those raw variables anymore. The formation of dummy variables from categorical data will be seen in this activity. The dummy variables are created using:

*pd.get\_dummies()*

Open the following link to get started:

<https://bit.ly/34OaWwf>

# Next Steps

Further improvements can also be done by using these features:

**Class Imbalance:** Class imbalance involves the scenarios in which one class outnumbers another class in the dataset. It can be seen that many of the examples in a dataset does not belong to any class. When there are class imbalances, there is a likelihood that the classifier overfits the majority class. There are many scenarios in real life where the class imbalance is very prevalent. These include fraud detection, medical diagnosis, and customer churn.

**Feature Engineering:** In data science, achieving the desired outcome will depend on the diversity of experiments we take. Making changes to the raw materials through feature engineering is one big area to make improvements in the initial model. We dealt with feature engineering and built a model using feature-engineered variables. Identification of trails would depend on extending the business knowledge applied through the hypothesis we formulate and the exploratory analysis done to validate the business hypothesis.

**Model Selection Strategy:** When we discussed parametric and non-parametric models, we had seen that if the real data generation process is not similar to the model that we have assumed, we will get poor results. Thus we assumed linear models in that case. There are some questions related to it like what if the real data generation process is not linear? Or, what if there are other parametric or non-parametric models that are much more suitable for this use case? Such considerations are utilized when we try to analyze results and to improve the model. A strategy known as model spot-checking must be adopted that can entail working out the use case with different models and checking the initial metrics before adopting a model for the use case.

# Classification using Logistic Regression with Scikit-Learn:

In the Logistic Regression Classifier, the probabilities defining the possible outcomes of a single trial are modeled using a logistic function. It is implemented through:

from sklearn.linear\_model import LogisticRegression

The sklearn LR implementation can fit the binary, One-vs-Rest, or multinomial logistic regression with optional L2 or L1 regularization. For example, in the case of a binary classification on a sample sklearn dataset:

from sklearn.datasets import make\_hastie\_10\_2

X,y = make\_hastie\_10\_2(n\_samples=1000)

Using the Train-test split to divide the input data into training and test sets:

from sklearn.model\_selection import train\_test\_split

#sklearn.cross\_validation in older scikit versions

data\_train, data\_test, labels\_train, labels\_test = train\_test\_split(X,y, test\_size=0.3)

LRC = LogisticRegression()

LRC.fit(data\_train, labels\_train)

predicted = LRC.predict(data\_test)

**For Confusion Matrix:**

from sklearn.metrics import confusion\_matrix

confusion\_matrix(predicted, labels\_test)

# Summary

In this chapter, we have studied binary classification, mainly concerned with categorization with binary decisions. We have used logistic regression from the perspective of solving a use case. We were introduced to classification problems. We also looked at the Feature Engineering process with its business-driven and data-driven phases.

Intuitions obtained from EDA were utilized to generate new features from the raw variables. A logistic regression model was built and the metrics were analyzed to predict a future course of action. In the next chapter, you will deal with multiclass classification problems and will be going to handle those.