# Twitter Sentiment Analysis of Airline Reviews

**A Project Report submitted in partial fulfillment of the requirements for the award of the**

## Table of Contents

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO |  | CONTENT | PGNO |
| 1 | **MACHINE LEARNING**  1.1. Introduction  1.2. Importance of Machine Learning  1.3. Uses of Machine Learning  1.4. Types of Learning Algorithms  1.4.1 Supervised Learning  1.4.2 Unsupervised Learning  1.4.3 Semi Supervised Learning |  | 01 |
| 2 | **PYTHON**  2.1. Introduction To Python  2.1.1. History Of Python  2.2. How to Setup Python  2.2.1. Installation (using python IDLE)  2.2.2. Installation (using Anaconda)  2.3. Features Of Python  2.4. Python Variable Types  2.4.1. Python Numbers  2.4.2. Python Strings  2.4.3. Python Lists  2.4.4. Python Tuples  2.4.5. Python Dictionary  2.5. Python Functions  2.5.1. Defining a Function  2.5.2. Calling a Function  2.6. Python Using OOPs Concepts  2.6.1. Class  2.6.2. init Method in Class |  | 06 |
| 3 | **ENSEMBLE METHODS**  3.1. Introduction  3.1.1 Averaging Method  3.1.2 Max Voting |  | 14 |

|  |  |  |
| --- | --- | --- |
|  | 3.1.3 Bagging  3.1.4 Boosting  3.1.5 KNORA-E |  |
| 4 | **TWITTER SENTIMENT ANALYSIS**  4.1.Project Requirements  4.1.1. Packages used  4.1.2. Algorithms used  4.2. Problem Statement  4.3. Dataset Description  4.4. Objective of the Case Study | 16 |
| 5 | **DATA PREPROCESSING/FEATURE ENGINEERING AND**  **EDA**  5.1. Loading The Data  5.2. Balancing The Dataset  5.3. Noise and Stopword Removal  5.4. Word vectorization | 19 |
| 6 | **MODEL BUILDING AND EVALUATION**  6.1. Brief About the Algorithms Used and Building Model 6.2. Model Evaluation | 23 |
| 7 | **CONCLUSION** | 30 |
| 8 | **REFERENCES** | 31 |

**List of Figures**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FIG.NO |  |  | FIG.NAME | PG.NO |
| 1.2.1 | The Process Flow |  |  | 02 |
| 1.4.1 | Supervised Learning |  |  | 03 |
| 1.4.2 | Unsupervised Learning |  |  | 04 |
| 1.4.3 | Semi Supervised Learning |  |  | 05 |
| 2.2.1 | Python download |  |  | 07 |
| 2.2.2 | Anaconda download |  |  | 08 |
| 4.1.1 | Packages used |  |  | 17 |
| 5.1.1 | Loading data |  |  | 19 |
| 5.1.2 | Reading Data |  |  | 20 |
| 5.2.1 | Balancing the Dataset |  |  | 21 |
| 5.3.1 | Noise and stopword Removal |  |  | 21 |
| 5.4.1 | Word Vectorization |  |  | 22 |
| 6.1.1 | Averaging Method |  |  | 23 |
| 6.1.2 | Max Voting |  |  | 24 |
| 6.1.3 | Bagging |  |  | 25 |
| 6.1.4 | Boosting |  |  | 26 |
| 6.1.5 | KNORA-E |  |  | 27 |
| 6.2.1 | Model Evaluation |  |  | 28 |

## CHAPTER 1: MACHINE LEARNING

**1.1. INTRODUCTION**:

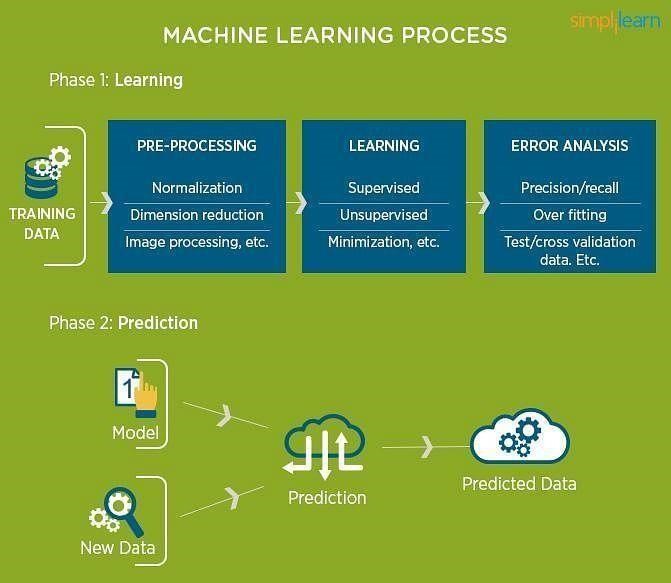
Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

**1.2 IMPORTANCE OF MACHINE LEARNING:**

Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps analyse those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques. The process flow depicted here represents how machine learning works.



### Fig 1.2.1: The Process Flow

**1.3 USES OF MACHINE LEARNING:**

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let’s consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data.

Traditionally, data analysis was always characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

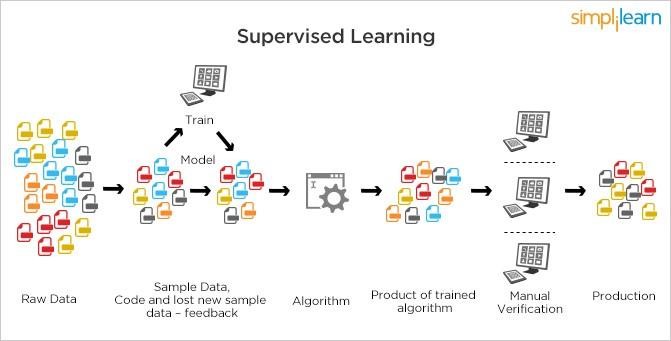
**1.4 TYPES OF LEARNING ALGORITHMS:**

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

**1.4.1 Supervised Learning:**

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

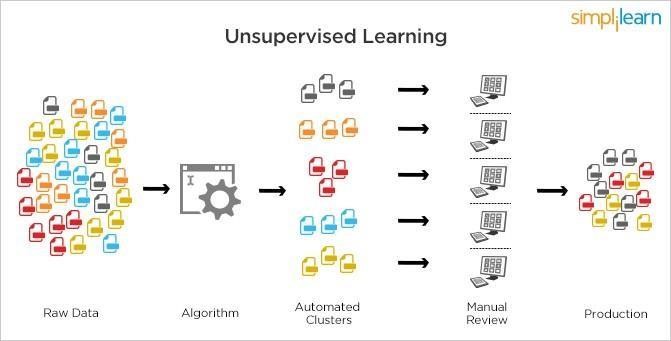
Supervised machine learning algorithms uncover insights, patterns, and relationships from a labeled training dataset – that is, a dataset that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to “learn” how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.



### Fig 1.4.1: Supervised Learning

**1.4.2 Unsupervised Learning:**

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.



### Fig 1.4.2: Unsupervised Learning

Popular techniques where unsupervised learning is used also include self-organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

**1.4.3 Semi Supervised Learning:**

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabelled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabelled data.

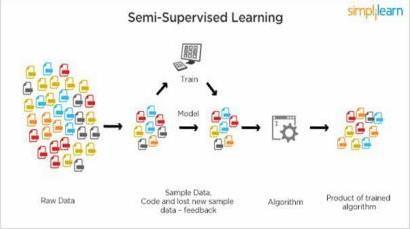


Fig 1.4.3: Semi Supervised Learning

## CHAPTER 2: PYTHON

Basic programming language used for machine learning is: PYTHON

**2.1. INTRODUCTION TO PYTHON:**

∙ Python is a high-level, interpreted, interactive and object-oriented scripting language.

∙ Python is a general-purpose programming language that is often applied in scripting roles

∙ Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP. ∙ Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.

∙ Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

**2.1.1 History of python:**

Python was developed by GUIDO VAN ROSSUM in early 1990’s ∙ Its latest version is 3.7.3, it is generally called as python3

**2.2 HOW TO SETUP PYTHON:**

∙ Python is available on a wide variety of platforms including Linux and Mac OS X. Let’s understand how to set up our Python environment.

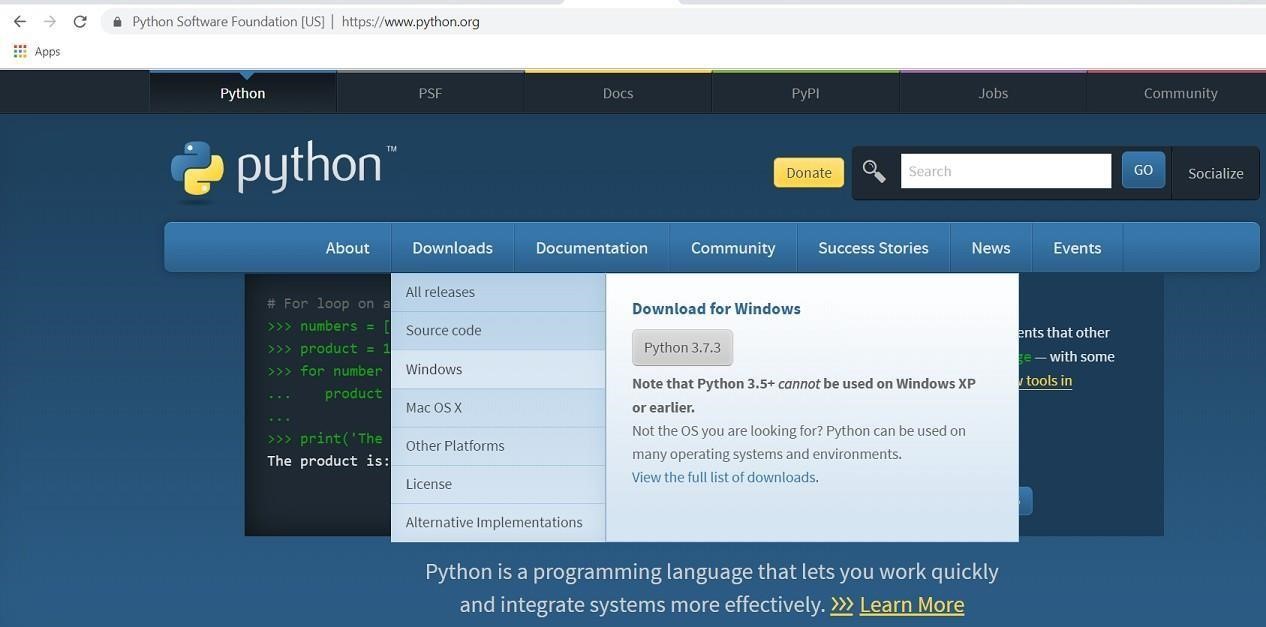
∙ The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

**2.2.1 Installation (using python IDLE):**

∙ Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.

∙ Download python from [www.python.org](http://www.python.org/)

When the download is completed, double click the file and follow the instructions to install it. ∙ When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.



### Fig 2.2.1: Python download

**2.2.2 Installation (using Anaconda):**

∙ Python programs are also executed using Anaconda.

∙ Anaconda is a free open-source distribution of python for large scale data processing, predictive analytics and scientific computing.

∙ Conda is a package manager that quickly installs and manages packages.

∙ In WINDOWS:

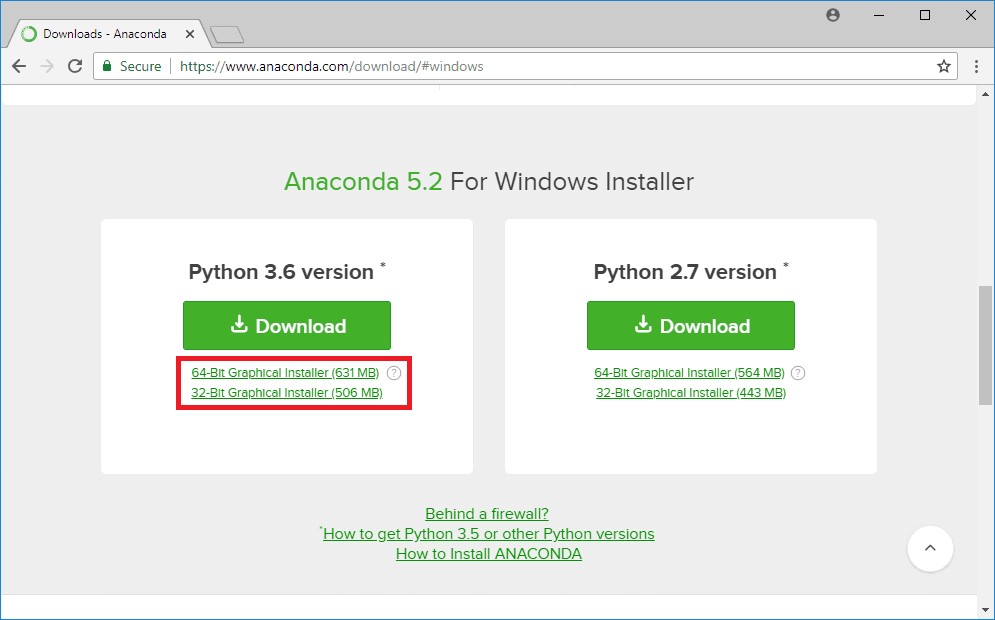
∙ Step 1: Open Anaconda.com/downloads in the web browser.

∙ Step 2: Download python 3.4 version for (32-bits graphic installer/64 -bit graphic installer)

∙ Step 3: select installation type (all users)

∙ Step 4: Select path (i.e. add anaconda to path & register anaconda as default python 3.4) next click install and next click finish

∙ Step 5: Open Jupiter notebook (it opens in default browser)



### Fig 2.2.2: Anaconda download

**2.3 FEATURES OF PYTHON:**

∙ Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

∙ Easy-to-read: Python code is more clearly defined and visible to the eyes. ∙ Easy-to-maintain:

Python's source code is fairly easy-to-maintaining.

∙ A broad standard library: Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.

∙ Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

∙ Extendable: You can add low-level modules to the Python interpreter.

∙ These modules enable programmers to add to or customize their tools to be more efficient.

∙ Databases: Python provides interfaces to all major commercial databases.

GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**2.4 PYTHON VARIABLE TYPES:**

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory. Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Python has five standard data types are:

∙ Numbers

∙ Strings

∙ Lists

∙ Tuples

∙ Dictionary

**2.4.1 Python Numbers:**

∙ Number data types store numeric values. Number objects are created when you assign a value to them.

∙ Python supports four different numerical types − int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

**2.4.2 Python Strings:**

∙ Strings in Python are identified as a contiguous set of characters represented in the quotation marks.

∙ Python allows for either pairs of single or double quotes.

∙ Subsets of strings can be taken using the slice operator ([ ] and [:] ) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

∙ The plus (+) sign is the string concatenation operator and the asterisk (\*) is the repetition operator.

**2.4.3 Python Lists:**

∙ Lists are the most versatile of Python's compound data types.

∙ A list contains items separated by commas and enclosed within square brackets ([]).

∙ To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data types.

∙ The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. ∙ The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

**2.4.4 Python Tuples:**

∙ A tuple is another sequence data type that is similar to the list.

∙ A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.

∙ The main differences between lists and tuples are: Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated.

∙ Tuples can be thought of as read-only lists.

∙ For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple.

Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

**2.4.5 Python Dictionary:**

∙ Python's dictionaries are a kind of hash table type.

∙ They work like associative arrays or hashes found in Perl and consist of key-value pairs. ∙ A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.

∙ Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

∙ You can use numbers to "index" into a list, meaning you can use numbers to find out what’s in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.

∙ What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

**2.5 PYTHON FUNCTION:**

**2.5.1 Defining a Function:**

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.

The code block within every function starts with a colon (:) and is indented. The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

**2.5.2 Calling a Function:**

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

**2.6 PYTHON USING OOPs CONCEPTS:**

**2.6.1 Class:**

A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

∙ Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.

∙ Data member: A class variable or instance variable that holds data associated with a class and its objects.

∙ Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.

∙ Defining a Class:

∙ We define a class in a very similar way to how we define a function.

∙ Just like a function ,we use parentheses and a colon after the class name(i.e. ():). Fig 3.6.1.1: Defining a Class

**2.6.2 init method in Class:**

* The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
* The init method has a special name that starts and ends with two underscores: init ().

## CHAPTER 3: ENSEMBLE METHODS

**3.1.Introduction:**

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model. To better understand this definition lets take a step back into ultimate goal of machine learning and model building. This is going to make more sense as I dive into specific examples and why Ensemble methods are used.

### **3.1.1. Averaging Method**

It is mainly used for regression problems. The method consists of building multiple models independently and returning the average of the prediction of all the models. In general, the combined output is better than an individual output because variance is reduced.

#### 3.1.2. Max Voting

It is mainly used for classification problems. The method consists of building multiple models independently and getting their individual output called ‘vote’. The class with maximum votes is returned as output.

#### 3.1.3. Bagging

It is also known as a bootstrapping method. Base models are run on bags to get a fair distribution of the whole dataset. A bag is a subset of the dataset along with a replacement to make the size of the bag the same as the whole dataset. The final output is formed after combining the output of all base models.

#### 3.1.4. Boosting

Boosting is a sequential method–it aims to prevent a wrong base model from affecting the final output. Instead of combining the base models, the method focuses on building a new model that is dependent on the previous one. A new model tries to remove the errors made by its previous one. Each of these models is called weak learners. The final model (aka strong learner) is formed by getting the weighted mean of all the weak learners.

#### 3.1.5. KNORA-E

This method searches for a local Oracle, which is a base classifier that correctly classify all samples belonging to the region of competence of the test sample. All classifiers with a perfect performance in the region of competence are selected (local Oracles). In the case that no classifier achieves a perfect accuracy, the size of the competence region is reduced (by removing the farthest neighbor) and the performance of the classifiers are re-evaluated. The outputs of the selected ensemble of classifiers is combined using the majority voting scheme. If no base classifier is selected, the whole pool is used for classification.

## CHAPTER 4: TWITTER SENTIMENT ANALYSIS

**4.1. PROJECT REQUIREMENTS:**

**4.1.1.Packages used:**

**Numpy:** In Python we have lists that serve the purpose of arrays, but they are slow to process. Numpy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in Numpy is called ndata, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

**Pandas:** Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Scikit-learn:** also called Sklearn, is a robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling, including classification, regression, clustering, and dimensionality reduction via a consistent interface.

**Collections:** The collection Module in Python provides different types of containers. A Container is an object that is used to store different objects and provide a way to access the contained objects and iterate over them. Some of the built-in containers are [Tuple,](https://www.geeksforgeeks.org/python-tuples/) [List,](https://www.geeksforgeeks.org/python-list/) [Dictionary,](https://www.geeksforgeeks.org/python-dictionary/) etc.

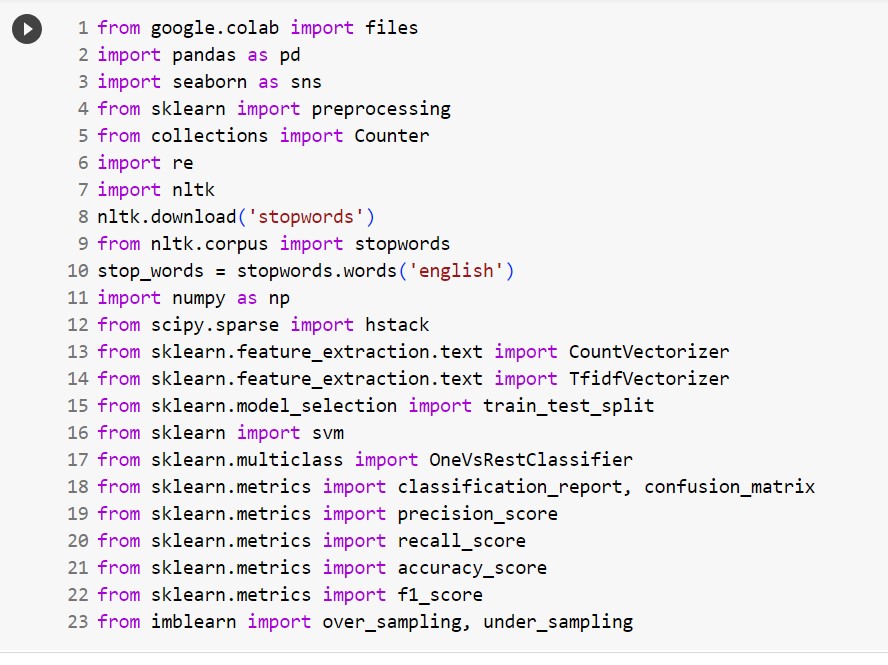
**Requests:** The collection Module in Python provides different types of containers. A Container is an object that is used to store different objects and provide a way to access the contained objects and iterate over them. Some of the built-in containers are [Tuple,](https://www.geeksforgeeks.org/python-tuples/) [List,](https://www.geeksforgeeks.org/python-list/) [Dictionary,](https://www.geeksforgeeks.org/python-dictionary/) etc.

**NLTK:** The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.

**Imblearn:** Imbalanced-Learn is a Python module that helps in balancing the datasets which are highly skewed or biased towards some classes. Thus, it helps in resampling the classes which are otherwise oversampled or undesampled. If there is a greater imbalance ratio, the output is biased to the class which has a higher number of examples.

**Seaborn:** is a library for making statistical graphics in Python. It builds on top of matplotlib and

integrates closely with pandas’ data structures. Seaborn helps you explore and understand your data.



### Fig 4.1.1: Packages used

**4.1.2. Algorithms Used**

∙ Random Forest

∙ XGBoost

∙ Linear Regression

∙ Averaging Method

∙ Max Voting

∙ Bagging

∙ Boosting

∙ KNORA-E

**4.2 PROBLEM STATEMENT:**

The aim of this dataset is to predict the sentiment of the passenger or traveller of the airlines

through their comments in twitter.

**4.3 DATASET DESCRIPTION:**

The dataset is also distributed as a CSV formatted file hearts.csv. The loaded object contains

‘tweet\_id’, ‘airline\_sentiment’, ‘airline\_sentiment\_confidence’, ‘negativereason’,

‘negativereason\_confidence’, ‘airline’, ‘airline\_sentiment\_gold’, ‘name’, ‘negativereason\_gold’,

‘retweet\_count’, ‘text’, ‘tweet\_coord’, ‘tweet\_created’, ‘tweet\_location’, ‘user\_timezone’.

**4.4 OBJECTIVE OF THE CASE STUDY:**

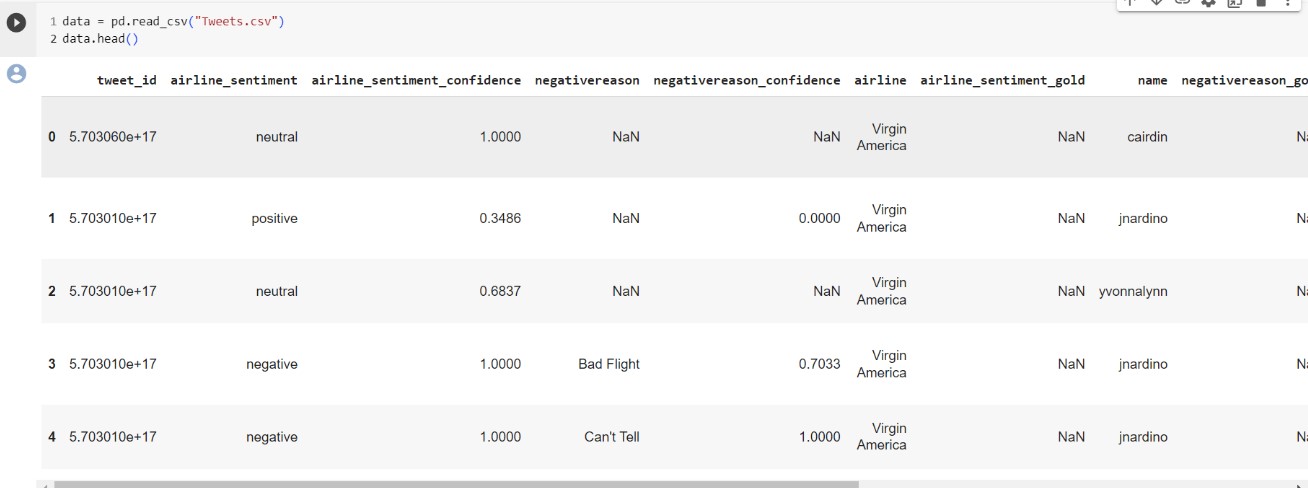
Objective of the problem is To analyze Twitter sentiment related to airlines with the goal of understanding customer perceptions, identifying prevalent issues, and gauging overall customer satisfaction. By employing sentiment analysis techniques, this study aims to categorize tweets into positive, negative, or neutral sentiments, enabling airlines to proactively address concerns, enhance customer experience, and optimize their services based on real-time feedback.

## CHAPTER 5: DATA PREPROCESSING/ FEATURE ENGINEERING AND EDA

**5.1 LOADING THE DATA:**

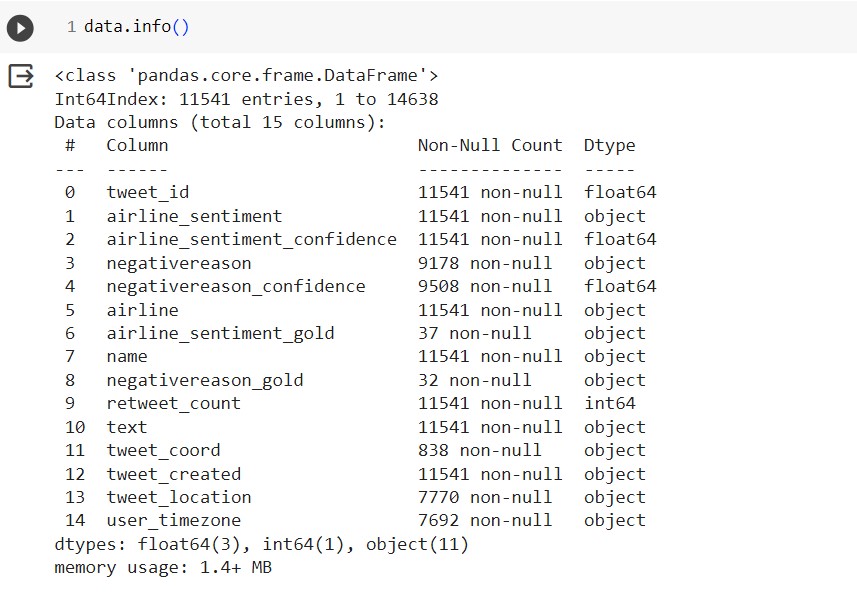
Python has a built-in head () function to read a file. This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.

We can specify the mode while opening a file. In mode, we specify whether we want to read, write or append to the file. We can also specify if we want to open the file in text mode or binary mode.



### Fig 5.1.1: Loading data

Pandas DataFrame to which all the operations can be performed which helps us to access each and every row as well as columns and each and every value can be accessed using the dataframe. Any missing value or NaN value have to be cleaned.



### Fig 5.1.2: Reading Data

**5.2 BALANCING THE DATASET:**

Imbalanced data means the number of rows or frequency of data points of one class is much more than the other class. In other words, the ratio of the value counts of classes is much higher. Such data set is known as an imbalanced dataset in which the class having more data points is the majority class and the other is the minority class.

The imbalance makes the classification more challenging. Whenever we build a classifier with such data, it works well with the majority class but gives a poor performance with the minority class. Although, the model performance concerning the minority class matters the most. Some Machine Learning algorithms are more sensitive toward imbalanced data, such as Logistic Regression and Support Vector Machine. However, some algorithms tackle this issue themselves, such as Random Forest and XGBoost.

There are two sampling techniques available to handle the imbalanced data:

1. Under Sampling
2. Over Sampling

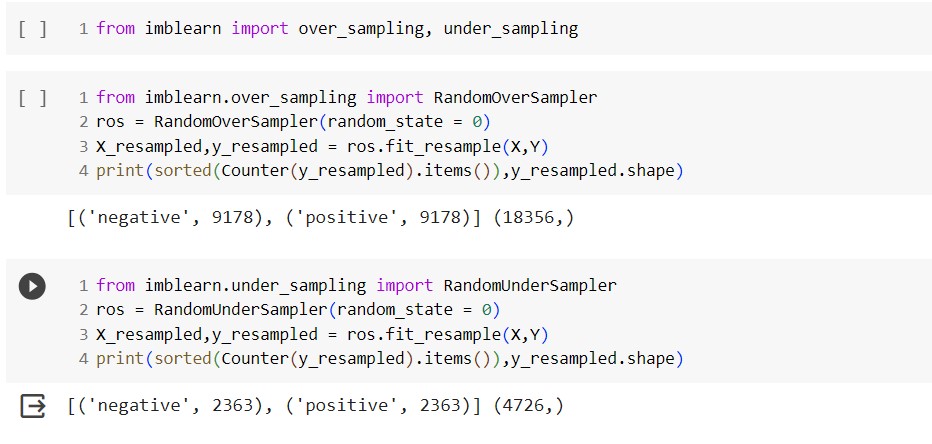
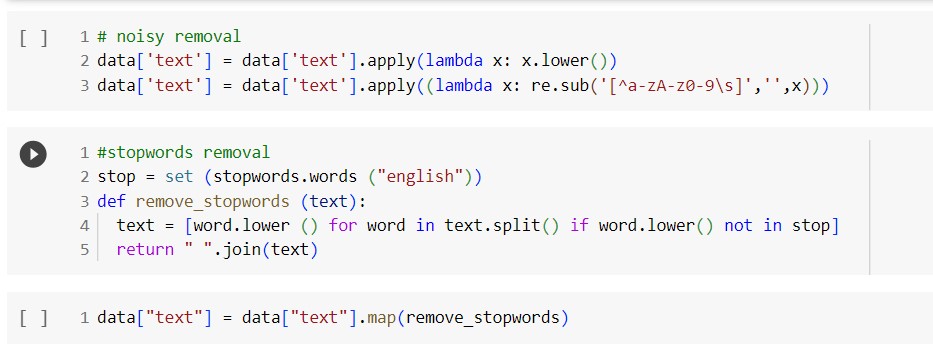


Fig 5.2.1: Balancing the Dataset

**5.3 NOISE AND STOPWORD REMOVAL:**

Noise removal in the context of text data refers to the process of eliminating irrelevant or unnecessary information that does not contribute to the analysis or understanding of the text. In the context of Twitter sentiment analysis, noise can include special characters, numbers, URLs, and other elements that do not carry significant meaning regarding sentiment.

Stopword removal is a common step in text preprocessing for natural language processing tasks, including sentiment analysis. Stopwords are words that are considered to be of little value in terms of content analysis because they are very common and do not carry significant meaning. Examples of stopwords in English include "and," "is," "the," "in," etc. Removing stopwords can reduce the dimensionality of the data and improve the efficiency of algorithms, especially in tasks like sentiment analysis where the sentiment-carrying words are often adjectives, verbs, or specific nouns.



### Fig 5.3.1: Noise and Stopword removal

**5.4 WORD VECTORIZATION:**

Word vectorization techniques like Bag of Words (BoW) and Term Frequency-Inverse Document

Frequency (TF-IDF) are essential in natural language processing tasks, including sentiment analysis.

They transform text data into numerical vectors that machine learning algorithms can understand.



Fig 5.4.1: Word vectorization

## CHAPTER 6: MODEL BUILDING AND EVALUATION

**6.1. BRIEF ABOUT THE ALGORITHMS USED AND BUILDING MODEL**

We have used ensemble methods to predict whether the data in twitter about the airlines review given is accurate or not. We have used five different models to check the accuracy and all evaluation metrics for the given dataset. We will be explaining the methods below:

**1.Averaging method:**

It is mainly used for regression problems. The method consists of building multiple models independently and returning the average of the prediction of all the models. In general, the combined output is better than an individual output because variance is reduced. Here we used three methods, then we will average the predictions of all models.



### Fig 6.1.1: Averaging Method

**2.Max Voting:**

It is mainly used for classification problems. The method consists of building multiple models independently and getting their individual output called ‘vote’. The class with maximum votes is returned as output.



### Fig 6.1.2: Max Voting

**3. Bagging:**

It is also known as a bootstrapping method. Base models are run on bags to get a fair distribution of the whole dataset. A bag is a subset of the dataset along with a replacement to make the size of the bag the same as the whole dataset. The final output is formed after combining the output of all base models.



### Fig 6.1.3: Bagging

**4.Boosting:**

Boosting is a sequential method–it aims to prevent a wrong base model from affecting the final output. Instead of combining the base models, the method focuses on building a new model that is dependent on the previous one. A new model tries to remove the errors made by its previous one. Each of these models is called weak learners. The final model (aka strong learner) is formed by getting the weighted mean of all the weak learners.



### Fig 6.1.4: Boosting

**5.KNORA-E:**

This method searches for a local Oracle, which is a base classifier that correctly classify all samples belonging to the region of competence of the test sample. All classifiers with a perfect performance in the region of competence are selected (local Oracles). In the case that no classifier achieves a perfect accuracy, the size of the competence region is reduced (by removing the farthest neighbor) and the performance of the classifiers are re-evaluated. The outputs of the selected ensemble of classifiers is combined using the majority voting scheme. If no base classifier is selected, the whole pool is used for classification.



Fig 6.1.5: KNORA-E

**6.2. MODEL EVALUATION**

Model Evaluation can give us a brief information on the accuracies of the machine learning models we have used to predict the correct information. In all the models which has high accuracy can give us a correct prediction of heart disease.



Fig 6.2.1 Model Validation

* Hence, here the accuracy of the Max Voting is high.
* So, it gives us an accurate and reliable prediction on sentiment of the tweets of the airline passengers.

## CONCLUSION

Using ensemble techniques, specifically Max Voting, produced remarkably accurate results in our sentiment analysis of airline passengers on Twitter. This cooperative strategy, which combined various models, gave rich insights regarding passenger attitude that are essential for improving airline services. Airlines were able to address customer issues and build closer relationships by leveraging the strength of ensemble approaches, which demonstrated an ability to interpret complicated feelings. This study highlights Max Voting's efficacy and emphasizes the value of continuous analysis in order to adjust to changing consumer emotions in the ever-changing world of social media.

## REFERENCES

* <https://en.wikipedia.org/wiki/Machine_learning>

* <https://en.wikipedia.org/wiki/Python_(programming_language)>

* <https://www.geeksforgeeks.org/ensemble-methods-in-python/>

* [https://deslib.readthedocs.io/en/latest/modules/des/knora\_e.html#:~:text=k%2DNearest%20Or acles%20Eliminate%20(KNORA,are%20selected%20(local%20Oracles)](https://deslib.readthedocs.io/en/latest/modules/des/knora_e.html#:~:text=k%2DNearest%20Oracles%20Eliminate%20(KNORA,are%20selected%20(local%20Oracles).

* [https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-missalgorithm-in-python/](https://www.geeksforgeeks.org/ml-handling-imbalanced-data-with-smote-and-near-miss-algorithm-in-python/)

* **Base Paper:**

https://drive.google.com/file/d/1Iwp25pPSH1aE6DxbH5etUTmBIUTyUcM0/view?usp=drive\_link