**Compute the customer behavioral segmentation on shopping mall data.**

**Abstract:**

The use of machine learning can be seen almost everywhere around us, be it Facebook recognizing you or your friends, or YouTube recommending you a video or two based on your history — Machine Learning is everywhere!However, the ‘magic’ of machine learning is not just limited to only these areas.Machine Learning is broadly categorized as Supervised and Unsupervised Learning.Supervised Learning is one in which we teach the machine by providing both independent and dependent variables, for example, Classifying or predicting values.Unsupervised Learning mainly deals with identifying the structure or pattern of the data. In this type of algorithms, we do not have labeled data(or the dependent variable is absent), for example, clustering data, recommendation systems, etc.Unsupervised Learning provides amazing results as one can deduce many hidden relations between different attributes or features.In this article, I will be discussing a specific problem based on clustering techniques(Unsupervised Learning). However, my main aim in this article is to discuss the opulent use of machine learning in business and profit enhancement.

**Introduction:**

Machine Learning techniques are broadly divided into two parts :

* Supervised Machine Learning.
* Unsupervised Machine Learning.

In Supervised Machine Learning, the data is labelled and the algorithm learns from labelled training data. Examples of this method are Classification and Regression.In Unsupervised Machine Learning, we do not need to supervise the model. Such a method deals with unlabelled data. Unsupervised machine learning helps us find hidden and unknown patterns in data. Often it easier to get unlabelled data as compared to labelled data, and in such cases, we can use unsupervised machine learning to work on the data. Data, which needs categorization can be categorized with the help of unsupervised machine learning.Clustering is a type of unsupervised machine learning in which the algorithm processes our data and divided them into “clusters”.

**Existing System:**

Supervised algorithms has already implemented  for our data but the results was not to the satisfied level.

**Disadvantages:-**

* Accuracy level is  not good
* Time consumption is more

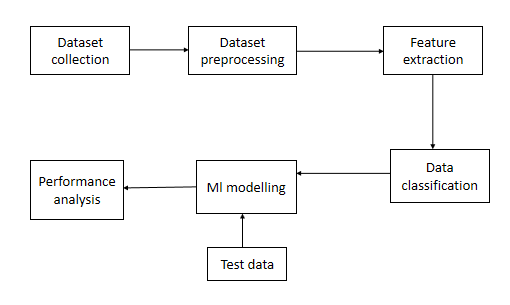
**Proposed System:**

In the proposed model of system , we are about to implement the k-means clustering algorithm and check the best accurate results with this approach.

**Advantages:**

* Accuracy level is good
* Time consumption is less
* Comparison of different algorithms can be observed .

**System Architecture:**

****

**System Requirements:**

**Hardware:**

* Operating system of windows  7 , 8, 10 (32-bit or 64-bit ).
* RAM-4GB

**Software:**

* Anaconda navigator software tool ,
* jupyter notebook editor window ,
* in python language.

**METHODOLOGY**

**3.2 Modules:**

* **DATA COLLECTION**
* **DATA PRE-PROCESSING**
* **FEATURE EXTRACTION**
* **EVALUATION MODEL**

**3.2.1DATA COLLECTION:**

Data used in this paper is a set of records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*. Collected in kaggal website.

**3.2.2DATA PRE-PROCESSING**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

1. Formatting
2. Cleaning
3. Sampling

**Formatting:** The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

**Cleaning:** Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonym zed or removed from the data entirely.

**Sampling:** There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

**3.2.3 FEATURE EXTRACTION**

Next thing is to do Feature extraction is an attribute extension we created more columns from shopping mall. Finally, our models are trained using Classifier algorithm. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen clustering were k means.

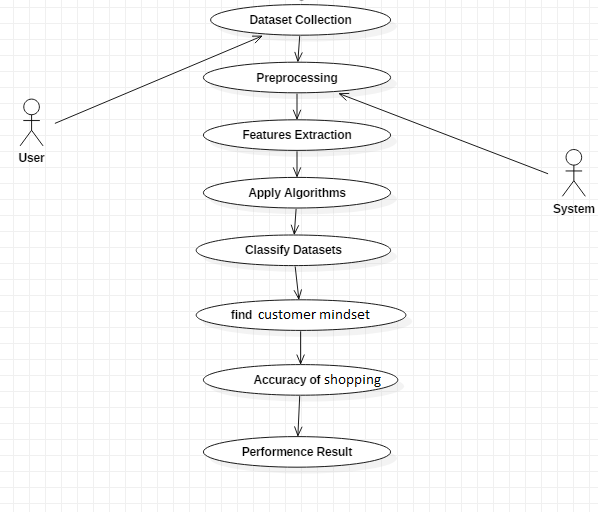
**3.2.4EVALUATION MODEL**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs. **Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions. We predict the accuracy over actual and predicted output and calculate accuracy as –



**3.3 UML DIAGRAMS**

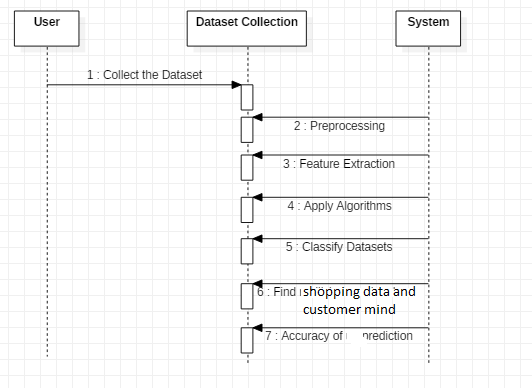
USE CASE DIAGRAM



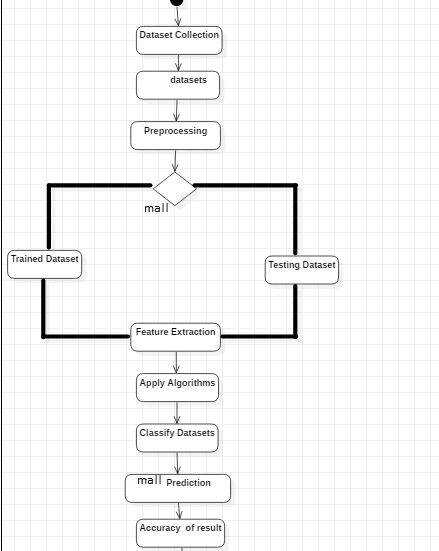
CLASS DIAGRAM



SEQUENCE DIAGRAM



ACTIVITY DIAGRAM



**3.4 ALGORITHM**

**3.4.1 RANDOM FOREST**

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of trees*, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

**3.4.2 HOW RANDOM FOREST WORKS**

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

**3.4.3 ADVANTAGES OF USING RANDOM FOREST**

pros of using random forest for classification and regression.

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features.
4. The random forest algorithm also works well when data has missing values or it has not been scaled we.

**3.5 Domain Specification**

**\**

motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

**Reinforcement Learning**

Reinforcement learningis a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

* Q-learning
* Deep Q network
* State-Action-Reward-State-Action (SARSA)
* Deep Deterministic Policy Gradient (DDPG)

**Applications/ Examples of deep learning applications**

**AI in Finance:**The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

**AI in HR:**Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

**AI in Marketing:**AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

**Artificial Intelligence**

ML

**Machine Learning**

**Deep Learning**

## Difference between Machine Learning and Deep Learning

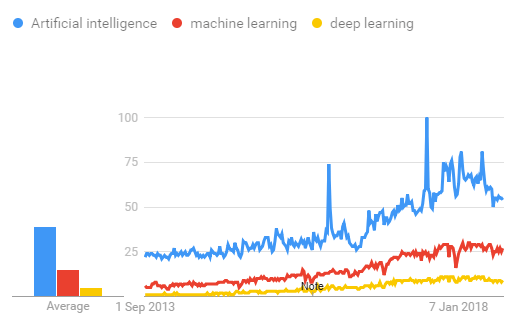
|  |  |  |
| --- | --- | --- |
|  | **Machine Learning** | **Deep Learning** |
| **Data Dependencies** | Excellent performances on a small/medium dataset | Excellent performance on a big dataset |
| **Hardware dependencies** | Work on a low-end machine. | Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication |
| **Feature engineering** | Need to understand the features that represent the data | No need to understand the best feature that represents the data |
| **Execution time** | From few minutes to hours | Up to weeks. Neural Network needs to compute a significant number of weights |
| **Interpretability** | Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost) | Difficult to impossible |

## When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning.

|  |  |  |
| --- | --- | --- |
|  | **Machine learning** | **Deep learning** |
| **Training dataset** | Small | Large |
| **Choose features** | Yes | No |
| **Number of algorithms** | Many | Few |
| **Training time** | Short | Long |

With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.



## TensorFlow

the most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

* Researchers
* Data scientists
* Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

In this tutorial, you will learn

**TensorFlow Architecture**

Tensorflow architecture works in three parts:

* Preprocessing the data
* Build the model
* Train and estimate the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

**Where can Tensorflow run?**

TensorFlow can hardware, and software requirements can be classified into

Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done Tensorflow can be run on many different platforms. You can run it on

* Desktop running Windows, macOS or Linux
* Cloud as a web service
* Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

**List of Prominent Algorithms supported by TensorFlow**

* Linear regression: tf.estimator.LinearRegressor
* Classification:tf.estimator.LinearClassifier
* Deep learning classification: tf.estimator.DNNClassifier
* Deep learning wipe and deep: tf.estimator.DNNLinearCombinedClassifier
* Booster tree regression: tf.estimator.BoostedTreesRegressor
* Boosted tree classification: tf.estimator.BoostedTreesClassifier

**3.6REQUIREMENTS ANALYSIS**

**SOFTWARE REQUIREMENTS**

* Python
* Anaconda Navigator
* Python built-in modules
* Numpy
* Pandas
* Matplotlib
* Sklearn
* Seaborm

**ANACONDA NAVIGATOR**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

## Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

## **WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR**?

The following applications are available by default in Navigator:

* JupyterLab
* Jupyter Notebook
* QTConsole
* Spyder
* VSCode
* Glueviz
* Orange 3 App
* Rodeo
* RStudio

Advanced conda users can also build your own Navigator applications

## How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

## What’s new in 1.9?

* Add support for **Offline Mode** for all environment related actions.
* Add support for custom configuration of main windows links.
* Numerous bug fixes and performance enhancements.

**PYTHON OVERVIEW**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted:** Python is processed at runtime by the interpreter.You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.
* **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.
* **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

**Python Features**

Python's features include:

* **Easy-to-learn:** Python has few keywords, simple structure, and a clearlydefined 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-maintain.
* **A broad standard library:** Python's bulk of the library is very portable andcross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.
* **Portable:** Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.
* **Extendable:** You can add low-level modules to the Python interpreter. Thesemodules 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 andported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable:** Python provides a better structure and support for large programsthan shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

* IT supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**PYTHON ENVIRONMENT**

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.

**Python’s standard library**

* Pandas
* Numpy
* Sklearn
* seaborn
* matplotlib
* Importing Datasets

**PANDAS**

Pandas is quite a game changer when it comes to analyzing data with Python and it is one of the most preferred and widely used tools in [data munging/wrangling](https://en.wikipedia.org/wiki/Data_wrangling) if not THE most used one. Pandas is an open source What’s cool about Pandas is that it takes data (like a CSV or TSV file, or a SQL database) and creates a Python object with rows and columns called data frame that looks very similar to table in a statistical software (think Excel or SPSS for example. People who are familiar with R would see similarities to R too). This is so much easier to work with in comparison to working with lists and/or dictionaries through for loops or list comprehension.

**Installation and Getting Started**

In order to “get” Pandas you would need to install it. You would also need to have Python 2.7 and above as a pre-requirement for installation. It is also dependent on other libraries (like [NumPy](http://www.numpy.org/)) and has optional dependancies (like Matplotlib for plotting). Therefore, I think that the easiest way to get Pandas set up is to install it through a package like the [Anaconda distribution](https://www.continuum.io/downloads), “a cross platform distribution for data analysis and scientific computing.”

In order to use Pandas in your Python IDE ([Integrated Development Environment](https://en.wikipedia.org/wiki/Integrated_development_environment)) like [Jupyter Notebook](http://jupyter.org/) or [Spyder](https://pythonhosted.org/spyder/) (both of them come with Anaconda by default), you need to import the Pandas library first. Importing a library means loading it into the memory and then it’s there for you to work with. In order to import Pandas all you have to do is run the following code:

* **import pandas as pd**
* **import numpy as np**

Usually you would add the second part (‘as pd’) so you can access Pandas with ‘pd.command’ instead of needing to write ‘pandas.command’ every time you need to use it. Also, you would import numpy as well, because it is very useful library for scientific computing with Python. Now Pandas is ready for use! Remember, you would need to do it every time you start a new Jupyter Notebook, Spyder file etc.

**Working with Pandas**

Loading and Saving Data with Pandas

When you want to use Pandas for data analysis, you’ll usually use it in one of three different ways:

* Convert a Python’s list, dictionary or Numpy array to a Pandas data frame
* Open a local file using Pandas, usually a CSV file, but could also be a delimited text file (like TSV), Excel, etc
* Open a remote file or database like a CSV or a JSONon a website through a URL or read from a SQL table/database

There are different commands to each of these options, but when you open a file, they would look like this:

* **pd.read\_filetype()**

As I mentioned before, there are different filetypes Pandas can work with, so you would replace “filetype” with the actual, well, filetype (like CSV). You would give the path, filename etc inside the parenthesis. Inside the parenthesis you can also pass different arguments that relate to how to open the file. There are numerous arguments and in order to know all you them, you would have to read the documentation (for example, the [documentation for pd.read\_csv()](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_csv.html)would contain all the arguments you can pass in this Pandas command).

In order to convert a certain Python object (dictionary, lists etc) the basic command is:

* **pd.DataFrame()**

Inside the parenthesis you would specify the object(s) you’re creating the data frame from. This command also has [different arguments](http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.html).

You can also save a data frame you’re working with/on to different kinds of files (like CSV, Excel, JSON and SQL tables). The general code for that is:

* **df.to\_filetype(filename)**

**Viewing and Inspecting Data**

Now that you’ve loaded your data, it’s time to take a look. How does the data frame look? Running the name of the data frame would give you the entire table, but you can also get the first n rows with df.head(n) or the last n rows with df.tail(n). df.shape would give you the number of rows and columns. df.info() would give you the index, datatype and memory information. The command s.value\_counts(dropna=False) would allow you to view unique values and counts for a series (like a column or a few columns). A very useful command is df.describe() which inputs summary statistics for numerical columns. It is also possible to get statistics on the entire data frame or a series (a column etc):

* df.mean() Returns the mean of all columns
* df.corr() Returns the correlation between columns in a data frame
* df.count() Returns the number of non-null values in each data frame column
* df.max()Returns the highest value in each column
* df.min()Returns the lowest value in each column
* df.median()Returns the median of each column
* df.std()Returns the standard deviation of each column

**Selection of Data**

One of the things that is so much easier in Pandas is selecting the data you want in comparison to selecting a value from a list or a dictionary. You can select a column (df[col]) and return column with label col as Series or a few columns (df[[col1, col2]]) and returns columns as a new DataFrame. You can select by position (s.iloc[0]), or by index (s.loc['index\_one']) . In order to select the first row you can use df.iloc[0,:] and in order to select the first element of the first column you would run df.iloc[0,0] . These can also be used in different combinations, so I hope it gives you an idea of the different selection and indexing you can perform in Pandas.

**Filter, Sort and Groupby**

You can use different conditions to filter columns. For example, df[df[year] > 1984] would give you only the column year is greater than 1984. You can use & (and) or | (or) to add different conditions to your filtering. This is also called boolean filtering.

It is possible to sort values in a certain column in an ascending order using df.sort\_values(col1) ; and also in a descending order using df.sort\_values(col2,ascending=False). Furthermore, it’s possible to sort values by col1 in ascending order then col2 in descending order by using df.sort\_values([col1,col2],ascending=[True,False]).

The last command in this section is groupby. It involves splitting the data into groups based on some criteria, applying a function to each group independently and combining the results into a data structure. df.groupby(col) returns a groupby object for values from one column while df.groupby([col1,col2]) returns a groupby object for values from multiple columns.

**Data Cleaning**

Data cleaning is a very important step in data analysis. For example, we always check for missing values in the data by running pd.isnull() which checks for null Values, and returns a boolean array (an array of true for missing values and false for non-missing values). In order to get a sum of null/missing values, run pd.isnull().sum(). pd.notnull() is the opposite of pd.isnull(). After you get a list of missing values you can get rid of them, or drop them by using df.dropna() to drop the rows or df.dropna(axis=1) to drop the columns. A different approach would be to fill the missing values with other values by using df.fillna(x) which fills the missing values with x (you can put there whatever you want) or s.fillna(s.mean()) to replace all null values with the mean (mean can be replaced with almost any function from the statistics section).

It is sometimes necessary to replace values with different values. For example, s.replace(1,'one') would replace all values equal to 1 with 'one'. It’s possible to do it for multiple values: s.replace([1,3],['one','three'])would replace all 1 with 'one' and 3 with 'three'. You can also rename specific columns by running: df.rename(columns={'old\_name': 'new\_ name'})or use df.set\_index('column\_one') to change the index of the data frame.

**Join/Combine**

The last set of basic Pandas commands are for joining or combining data frames or rows/columns. The three commands are:

df1.append(df2)— add the rows in df1 to the end of df2 (columns should be identical)

* df.concat([df1, df2],axis=1) — add the columns in df1 to the end of df2 (rows should be identical)
* df1.join(df2,on=col1,how='inner') — SQL-style join the columns in df1with the columns on df2 where the rows for colhave identical values. how can be equal to one of: 'left', 'right', 'outer', 'inner'

**NUMPY**

Numpy is one such powerful library for array processing along with a large collection of high-level mathematical functions to operate on these arrays. These functions fall into categories like Linear Algebra, Trigonometry, Statistics, Matrix manipulation, etc.

### Getting NumPy

NumPy’s main object is a homogeneous multidimensional array. Unlike python’s array class which only handles one-dimensional array, NumPy’s ndarray class can handle multidimensional array and provides more functionality. NumPy’s dimensions are known as axes. For example, the array below has 2 dimensions or 2 axes namely rows and columns. Sometimes dimension is also known as a rank of that particular array or matrix.

#### Importing NumPy

NumPy is imported using the following command. Note here np is the convention followed for the alias so that we don't need to write numpyevery time.

* import numpy as np

NumPy is the basic library for scientific computations in Python and this article illustrates some of its most frequently used functions. Understanding NumPy is the first major step in the journey of machine learning and deep learning.

#### 

**Sklearn**

In python, scikit-learn library has a pre-built functionality under sklearn. Pre processing.

Next thing is to do feature extraction Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally our models are trained using Classifier algorithm.. We use nltk . classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered . The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre processed data. The chosen classifiers were Decision tree , Support Vector Machines and Random forest. These algorithms are very popular in text classification tasks.

**SEABORN**

# **Data Visualization in Python**

Data visualization is the discipline of trying to understand data by placing it in a visual context, so that patterns, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. No matter if you want to create interactive, live or highly customized plots python has a excellent library for you.

**To get a little overview here are a few popular plotting libraries:**

* **[Matplotlib:](https://matplotlib.org/)**low level, provides lots of freedom
* **[Pandas Visualization:](https://pandas.pydata.org/pandas-docs/stable/visualization.html)**easy to use interface, built on Matplotlib
* **[Seaborn:](https://seaborn.pydata.org/)**high-level interface, great default styles
* **[ggplot:](http://ggplot.yhathq.com/)**based on R’s ggplot2, uses [Grammar of Graphics](https://www.amazon.com/Grammar-Graphics-Statistics-Computing/dp/0387245448)
* **[Plotly:](https://plot.ly/python/)**can create interactive plots

In this article, we will learn how to create basic plots using Matplotlib, Pandas visualization and Seaborn as well as how to use some specific features of each library. This article will focus on the syntax and not on interpreting the graphs.

### Matplotlib

Matplotlib is the most popular python plotting library. It is a low level library with a Matlab like interface which offers lots of freedom at the cost of having to write more code.

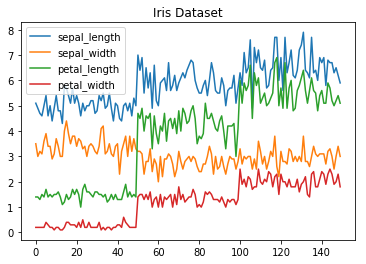
1. To install Matplotlib pip and conda can be used.
2. pip install matplotlib
3. conda install matplotlib

Matplotlib is specifically good for creating basic graphs like line charts, bar charts, histograms and many more. It can be imported by typing:

* **import matplotlib.pyplot as plt**

#### Line Chart

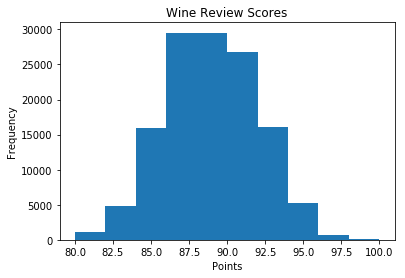
In Matplotlib we can create a line chart by calling the plot method. We can also plot multiple columns in one graph, by looping through the columns we want, and plotting each column on the same axis.



**Line Chart**

#### Histogram

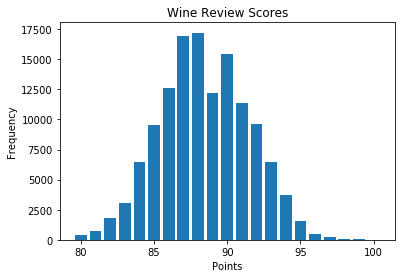
In Matplotlib we can create a Histogram using the hist method. If we pass it categorical data like the points column from the wine-review dataset it will automatically calculate how often each class occurs.



**Histogram**

#### Bar Chart

A bar-chart can be created using the bar method. The bar-chart isn’t automatically calculating the frequency of a category so we are going to use pandas value\_counts function to do this. The bar-chart is useful for categorical data that doesn’t have a lot of different categories (less than 30) because else it can get quite messy.



**Bar-Chart**

### Pandas Visualization

Pandas is a open source high-performance, easy-to-use library providing data structures, such as dataframes, and data analysis tools like the visualization tools we will use in this article.

Pandas Visualization makes it really easy to create plots out of a pandas dataframe and series. It also has a higher level API than Matplotlib and therefore we need less code for the same results.

1. **Pandas can be installed using either pip or conda.**
2. **pip install pandas**
3. **conda install pandas**

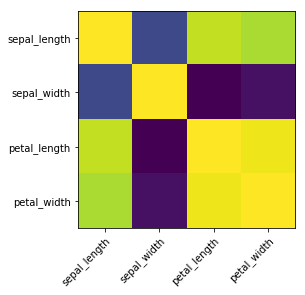
#### Heatmap

A Heatmap is a graphical representation of data where the individual values contained in a [matrix](https://en.wikipedia.org/wiki/Matrix_%28mathematics%29) are represented as colors. Heatmaps are perfect for exploring the correlation of features in a dataset.

To get the correlation of the features inside a dataset we can call <dataset>.corr() , which is a Pandas dataframe method. This will give use the [correlation matrix](https://www.displayr.com/what-is-a-correlation-matrix/).

We can now use either Matplotlib or Seaborn to create the heatmap.

**Matplotlib:**



**Heatmap without annotations**

Data visualization is the discipline of trying to understand data by placing it in a visual context, so that patterns, trends and correlations that might not otherwise be detected can be exposed.

Python offers multiple great graphing libraries that come packed with lots of different features. In this article we looked at Matplotlib, Pandas visualization and Seaborn.

**3.7 TESTING**

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

* Meets the business and technical requirements that guided its design and Development.
* Works as expected and can be implemented with the same characteristics.

**3.7.1 TESTING METHODS**

* **Functional Testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

* Functions: Identified functions must be exercised.
* Output: Identified classes of software outputs must be exercised.
* Systems/Procedures: system should work properly

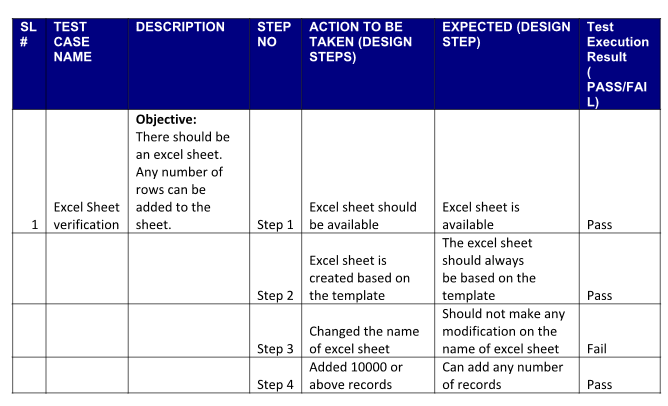
**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

Test Case for Excel Sheet Verification:

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on classification will work on the respective columns of dataset .

Test Case 1 :



**4.Results**

1. **CONCLUSION**

In this paper, we describe our large-scale system for automatically classifying phishing pages which maintains a false positive rate below 0.1%. Our classiﬁcation system examines millions of potential phishing pages daily in a fraction of the time of a manual review process. By automatically updating our blacklist with our classiﬁer, we minimize the amount of time that phishing pages can remain active before we protect our users from them. Even with a perfect classiﬁer and a robust system, we recognize that our blacklist approach keeps us perpetually a step behind the phishers. We can only identify a phishing URL and normal URL using machine learning algorithm. Result we got in terms of accuracy metric.

**FUTURE SCOPE**

As shopping culture increase day by day, some features may be included or replaced with new ones to detect people mindset them.

1. **REFERENCES**

**1. Muhammad A, Nabil AM, Ariff LS, Abdullah A-M (2019) Customer relationship management and big data enabled: Personalization & customization of services. Appl Comput Inf 15(2):94–101**

**2. Beheshtian-Ardakani A, Fathian M, Gholamian M (2018) A novel model for product bundling and direct marketing in e-commerce based on market segmentation. Decis Sci Lett 7(1):39–54**

**3. Yoshua B, Aaron C, Pascal V (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828**

**4. Blum A, Mitchell T (1998) Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory, pages 92–100,**

**5. Carnein M, Trautmann H (2019) Customer segmentation based on transactional data using stream clustering. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp 280–292. Springer**

**6. Daqing C, Laing SS, Kun G (2012) Data mining for the online retail industry: a case study of RFM model-based customer segmentation using data mining. J Database Market Customer Strat Manag 19(3):197–208**