Data Analyst

A Training Report

Submitted in partial fulfilment of the requirements for the award of a degree of

Bachelor of Technology In Computer Science

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

From 14/08/2023 to 21/12/2023

SUBMITTED BY

Name of the student: vanum Jeevan Jyothi kumar

Registration Number : 12018267

Signeture of the student :

Student Declaration

I’m **vanum Jeevan Jyothi kumar** with registration number **12018267**, here by declare that the work done in the position “**Data Analyst**” from August, 2023 to December, 2023 under the supervision of Team lead at HackersDaddy Cyber security solutions is a record of original work for the partial fulfillment of the requirements for the award of the degree, Computer Science and Engineering.

Vanum Jeevan Jyothi kumar

(12018267)

Dated: 21-12-2023

Declaration by the supervisor

To whom so ever it may concern

This is to certify that vanum Jeevan Jyothi kumar, 12018267 from Lovely Professional University, Punjab, has worked as a Intern in HackersDaddy cyber security solutions on “Data Analyst” under my supervision from August,2023 to December, 2023. It is further stated that the work carried out by the student is a record of original work to the best of my knowledge for the partial fulfillment of the requirements for the award of the degree, B.Tech in Computer Science and Engineering.

Name of External Supervisor:	**Name of Internal Supervisor:**
Ganesh Kumar	Vikram
Designation of the External Supervisor	Designation of the Internal Supervisor
HR-HackersDaddy	
Dated: 14-11-2023	**Dated: 21-**12-2023

HACKERS DADDY

HackersDaddy Cyber Security Solutions

Date: December 21, 2023

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Vanum Jeevan Jyothi Kumar has worked with us as a Data Analyst-Intern from August 14, 2023 to Till Date. During his employment period with us, he has undertaken various projects with utmost satisfaction. He is sincere, hardworking and punctual in delivering his tasks on time. We wish him good luck for his future endeavours.

CYBER SECURITY ... LLP

Sandeep Reddy - Director

CONTINUOUS ASSESSMENT (CA) for INTERNSHIP/OJT

(By external Supervisor from organization)

Name of the student Vanum Jeevan Jyothi Kumar Registration Number 12018267

Internship Project Title (if/any): Data Analyst Intern

Name of Organization & Address: HackersDaddy Cyber Security Solutions Pvt Ltd, H.no:8-76 Sri Sai Nagar, Hyderabad, 500098.

Name of External Internship in-charge (with mobile number):

Ganesh Kumar Contact No: 8988888621

S.No.	Criteria	Marks Obtained	Maximum Marks
1	Student conduct during internship	9	10
2	Punctuality and Enthusiasm	19	20
3	Technical Skill & Knowledge	18	20
4	Performance	48	50
	Total	94	**100**

Date 21-12-2023 Authorized Signatory

Name Ganesh Kumar Designation HR Manager

HACKERSDADDY CYBER SECURITY SOLUTIONS LLP ★
HACKERS DADDY

Company Seal

Acknowledgment:

he elation and gratification of this internship will be incomplete without mentioning all the people who helped me to make it possible, and whose gratitude and encouragement was invaluable to me. I would like to thank all the trainers and my Manager **Mr. Ganesh Kumar** and my University Mentors for their support and guidance throughout my internship. I would like to express my gratitude towards my parents & members of the HackersDaddy Team for their kind cooperation and encouragement, which helped me throughout this internship. My thanks and appreciation go to my colleagues in developing the project and the people who have willingly helped me out with their abilities.

VANUM JEEVAN JYOTHI KUMAR(12018267)

INTRODUCTION TO COMPANY

HACKERS
DADDY

At Hackersdaddy, we are passionate about cyber security and helping businesses protect themselves from potential threats. Our team of experts provides comprehensive services and training to ensure your business is protected.

SERVICES PROVIDED BY THE HACKERSDADDY

IT Auditing and Risk Assessment :

Our Cyber Security Audit is a thorough examination of your company's IT infrastructure.
It detectsthreats and vulnerabilities, as well as flaws and high-risk behaviors.
Our Cyber Security Experts can advise you on the best course of action to greatly increase your cyber resilience, secure your data, and safeguard your organization.

Cyber Incident Response :

Our Cyber Incident Response service is available 24 hours a day, 7 days a week, and is designed to swiftly limit the detrimental consequences of a cyber-attack while also assisting organizations in their recovery.

Cyber Crime Investigation & Cyber Forensics :

HakersDaddy Cyber Security Solutions' Cyber Crime Investigation and Cyber Forensic Services are targeted at proactively managing security risks by detecting and resolving Cyber Crime situations that might negatively affect enterprises, such as cybercrime and online fraud, 24 hours a day, 365 days a year.

Fraud Detection System :

Fraud detection is a collection of procedures and analytics that help firms discover and prevent unlawful financial transactions. This can involve things like credit card fraud, identity theft, cyber hacking, and insurance fraud, among other things.

OUR PARTNERS

CyberHakz

CERTIPORT®

A PEARSON VUE BUSINESS

Enhancing SMS Spam Detection:
A Comparative Study of Machine Learning and Deep Learning Approaches

ABSTRACT

The number of people using mobile devices increasing day by day. SMS (short message service) is a text message service available in smartphones as well as basic phones. So, the traffic of SMS increased drastically. The spam messages also increased. The hackers try to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information, etc. So, spam classification has special attention. In this paper, we applied various machine learning and deep learning techniques for SMS spam detection. we used a dataset to train the machine learning and deep learning models like LSTM and NB. The SMS spam collection data set is used for testing the method. The dataset is split into two categories for training and testing the research. Our experimental results have shown that our NB model outperforms previous models in spam detection with an accuracy of good.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

1.2Jupyter

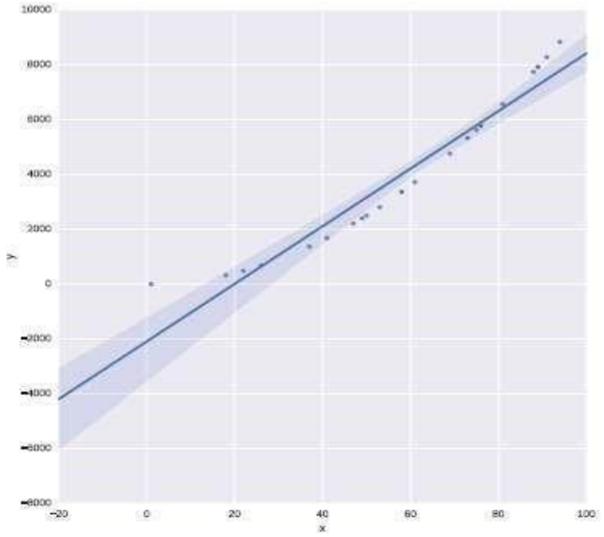
Jupyter, previously known as IPython Notebook, is a web-based, interactive development environment. Originally developed for Python, it has since expanded to support over 40 other programming languages including Julia and R.

Jupyter allows for *notebooks*to be written that contain text, live code, images, and equations. These notebooks can be shared, and can even be hosted on GitHubfor free. For each section of this tutorial, you can download a Juypter notebook that allows you to edit and experiment with the code and examples for each topic. Jupyter is part of the Anaconda distribution; it can be started from the command line using the jupyter command:

```
$ jupyter notebook
```

1.2 Machine Learning

We will now move on to the task of machine learning itself. In the following sections we will describe how to use some basic algorithms, and perform regression, classification, and clustering on some freely available medical datasets concerning breast cancer and diabetes, and we will also take a look at a DNA microarray dataset.



SciKit-Learn

SciKit-Learn provides a standardised interface to many of the most commonly used machine learning algorithms, and is the most popular and frequently used library for machine learning for Python. As well as providing many learning algorithms, SciKitLearn has a large number of convenience functions for common preprocessing tasks (for example, normalisation or *k*-fold cross validation).

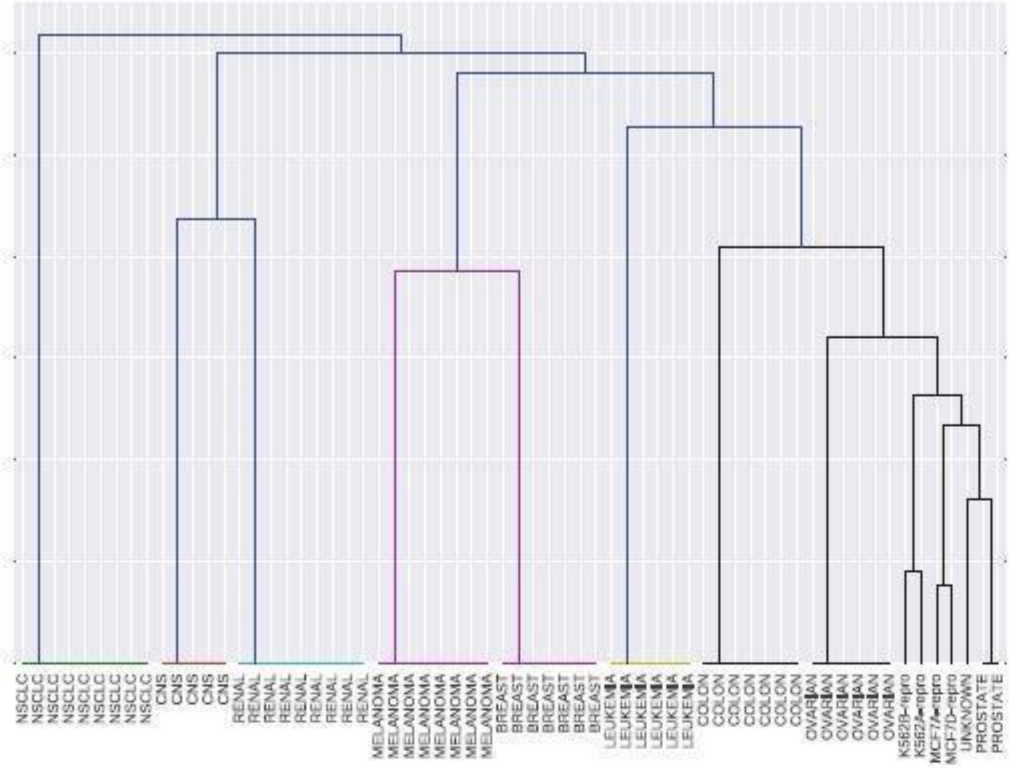
SciKit-Learn is a very large software library.

Clustering

Clustering algorithms focus on ordering data together into groups. In general clustering algorithms are unsupervised—they require no **y** response variable as input. That is to say, they attempt to find groups or clusters within data where you do not know the label for each sample. SciKit-Learn have many clusteringalgorithms, but in this section we will demonstrate hierarchical clustering on a DNA expression microarray dataset using an algorithm from the SciPy library.

We will plot a visualisation of the clustering using what is known as a dendrogram, also using the SciPy library.

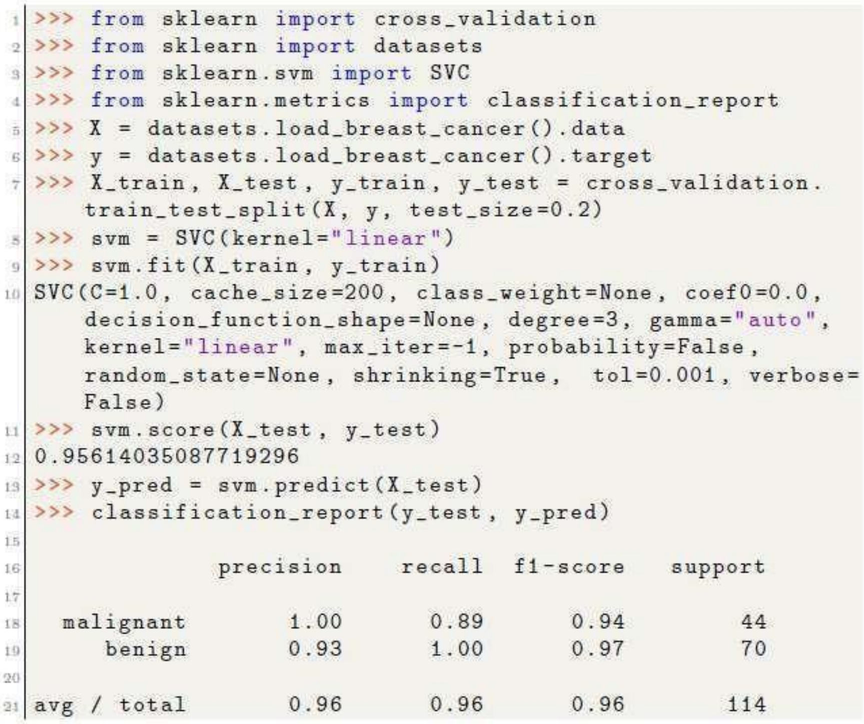
The goal is to cluster the data properly in logical groups, in this case into the cancer types represented by each sample‘s expression data. We do this using agglomerative hierarchical clustering, using Ward‘s linkage method:



1.4Classification

weanalysed data that was **unlabelled**—we did not know to what class a sample belonged (known as unsupervised learning). In contrast to this, a supervised problem deals with **labelled** data where are aware of the discrete classes to which each sample belongs. When we wish to predict which class a sample belongs to, we call this a classification problem. SciKit-Learn has a number of algorithms for classification, in this section we will look at the Support Vector Machine.

We will work on the Wisconsin breast cancer dataset, split it into a training set and a test set, train a Support Vector Machine with a linear kernel, and test the trained model on an unseen dataset. The Support Vector Machine model should be able to predict if a new sample is malignant or benign based on the features of a new, unseen sample:



```
>>> from sklearn import cross_validation
>>> from sklearn import datasets
>>> from sklearn.svm import SVC
>>> from sklearn.metrics import classification_report
>>> X = datasets.load_breast_cancer().data
>>> y = datasets.load_breast_cancer().target
>>> X_train, X_test, y_train, y_test = cross_validation.
    train_test_split(X, y, test_size=0.2)
>>> svm = SVC(kernel="linear")
>>> svm.fit(X_train, y_train)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape=None, degree=3, gamma="auto",
    kernel="linear", max_iter=-1, probability=False,
    random_state=None, shrinking=True,  tol=0.001, verbose=
    False)
>>> svm.score(X_test, y_test)
0.95614035087719296
>>> y_pred = svm.predict(X_test)
>>> classification_report(y_test, y_pred)

             precision    recall  f1-score   support

  malignant       1.00      0.89      0.94        44
     benign       0.93      1.00      0.97        70

avg / total       0.96      0.96      0.96       114
```

You will notice that the SVM model performed very well at predicting the malignancy of new, unseen samples from the test set—this can be quantified nicely by printing a number of metrics using the classification report function. Here, the precision, recall, and *F*1 score (*F*1 = 2· precision·recall/precision+recall) for each class is shown. The support column is a count of the number of samples for each class.

Support Vector Machines are a very powerful tool for classification. They work well in high dimensional spaces, even when the number of features is higher than the number of samples. However, their running time is quadratic to the number of samples so large datasets can become difficult to train. Quadratic means that if you increase a dataset in size by 10 times, it will take 100 times longer to train.

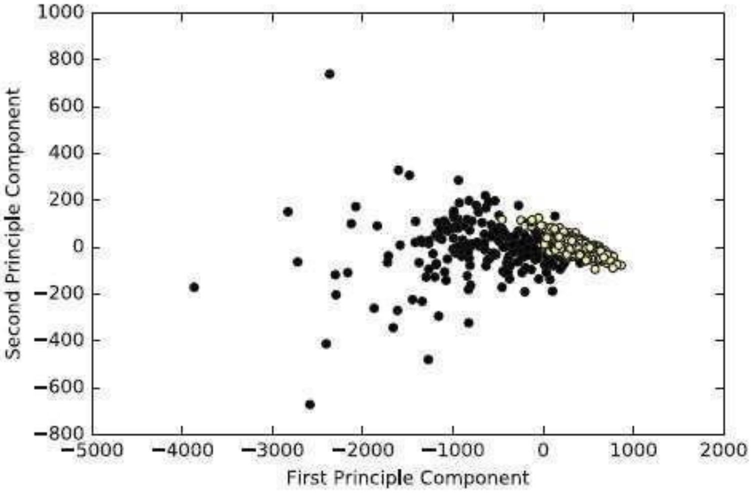
Last, you will notice that the breast cancer dataset consisted of 30 features. This makes it difficult to visualize or plot the data. To aid in visualization of highly dimensional data, we can apply a technique called dimensionality reduction.

Dimensionality Reduction

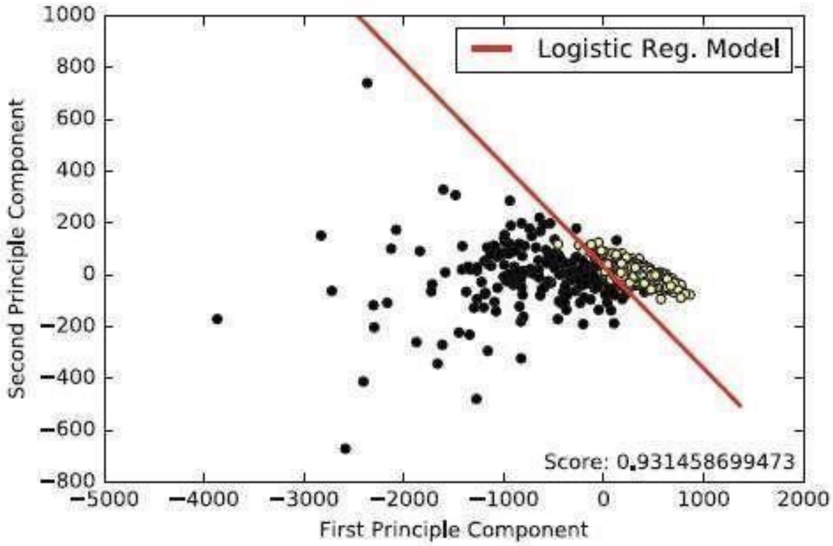
Another important method in machine learning, and data science in general, is dimensionality reduction. For this example, we will look at the Wisconsin breast cancer dataset once again. The dataset consists of over 500 samples, where each sample has 30 features. The features relate to images of a fine needle aspirate of breast tissue, and the features describe the characteristics of the cells present in the images. All features are real values. The target variable is a discrete value (either malignant or benign) and is therefore a classification dataset.

You will recall from the Iris example in Sect. 7.3 that we plotted a scatter matrix of the data, where each feature was plotted against every other feature in the dataset to look for potential correlations (Fig. 3). By examining this plot you could probably find features which would separate the dataset into groups. Because the dataset only had 4 features we were able to plot each feature against each other relatively easily. However, as the numbers of features grow, this becomes less and less feasible, especially if you consider the gene expression example in Sect. 9.4 which had over 6000 features. One method that is used to handle data that is highly dimensional is Principle Component Analysis, or PCA. PCA is an unsupervised algorithm for reducing the number of dimensions of a dataset. For example, for plotting purposes you might want to reduce your data down to 2 or 3 dimensions, and PCA allows you to do this by generating components, which are combinations of the original features, that you can then use to plot your data.

PCA is an unsupervised algorithm. You supply it with your data, **X**, and you specify the number of components you wish to reduce its dimensionality to. This is known as transforming the data:



Again, you would not use this model for new data—in a real world scenario, you would, for example, perform a 10-fold cross validation on the dataset, choosing the model parameters that perform best on the cross validation. This model would be much more likely to perform well on new data. At the very least, you would randomly select a subset, say 30% of the data, as a test set and train the model on the remaining 70% of the dataset. You would evaluate the model based on the score on the test set and not on the training set



1.4.1 NEURAL NETWORKS AND DEEP LEARNING

While a proper description of neural networks and deep learning is far beyond the scope of this chapter, we will however discuss an example use case of one of the most popular frameworks for deep learning: Keras4.

In this section we will use Keras to build a simple neural network to classify theWisconsin breast cancer dataset that was described earlier. Often, deep learning algorithms and neural networks are used to classify images—convolutional neural networks are especially used for image related classification. However, they can of course be used for text or tabular-based data as well. In this we will build a standard feed-forward, densely connected neural network and classify a text-based cancer dataset in order to demonstrate the framework‘susage.

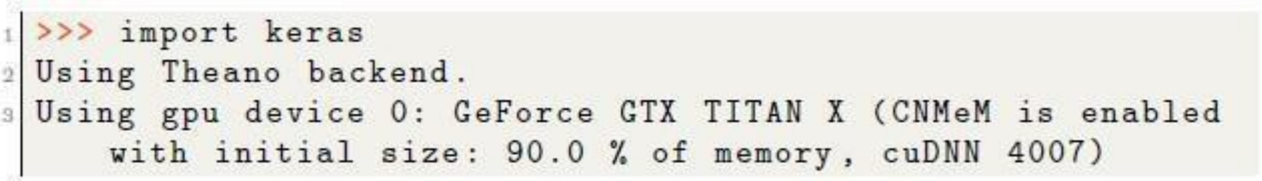
In this example we are once again using the Wisconsin breast cancer dataset, which consists of 30 features and 569 individual samples. To make it more challenging for the neural network, we will use a training set consisting of only 50% of the entire dataset, and test our neural network on the remaining 50% of the data.

Note,Keras is not installed as part of the Anaconda distribution, to install it use pip:

```
$sudo pip install keras
```

Keras additionally requires either Theano or TensorFlow to be installed. In the examples in this chapter we are using Theano as a backend, however the code will work identically for either backend. You can install Theano using pip, but it has a number of dependencies that must be installed first. Refer to the Theano and TensorFlow documentation for more information [12].

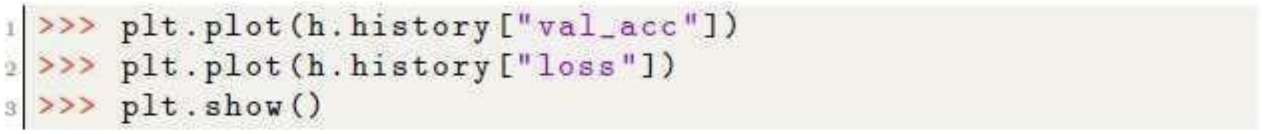
Keras is a modular API. It allows you to create neural networks by building a stack of modules, from the input of the neural network, to the output of the neural network, piece by piece until you have a complete network. Also, Keras can be configured to use your Graphics Processing Unit, or GPU. This makes training neural networks far faster than if we were to use a CPU. We begin by importing Keras:



```
>>> import keras
Using Theano backend.
Using gpu device 0: GeForce GTX TITAN X (CNMeM is enabled
    with initial size: 90.0 % of memory, cuDNN 4007)
```

We may want to view the network's accuracy on the test (or its loss on the training set) over time (measured at each epoch), to get a better idea how well it is learning. An epoch is one complete cycle through the training data.

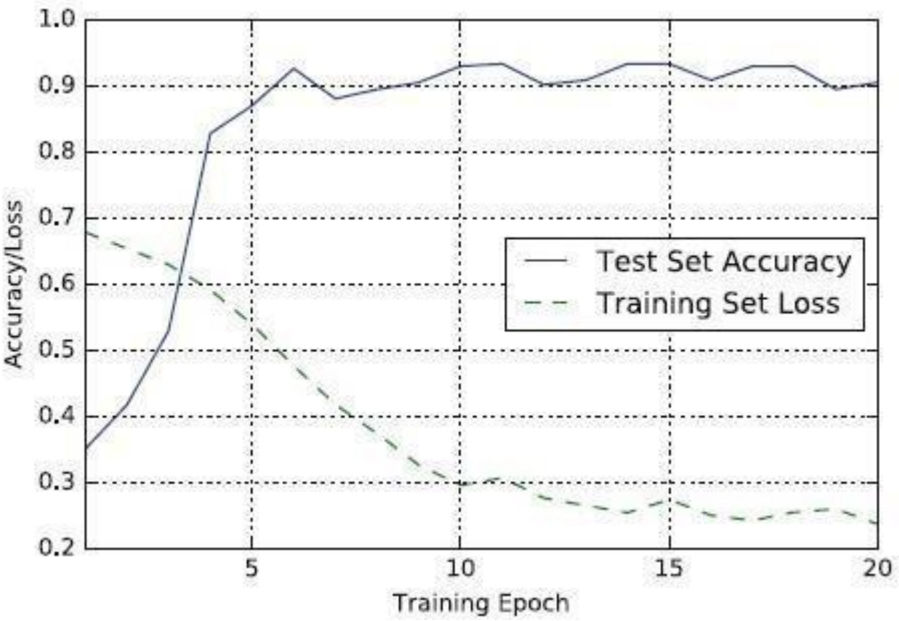
Fortunately, this is quite easy to plot as Keras' fit function returns a history object which we can use to do exactly this:



```
>>> plt.plot(h.history["val_acc"])
>>> plt.plot(h.history["loss"])
>>> plt.show()
```

This will result in a plot similar to that shown. Often you will also want to plot the loss on the test set and training set, and the accuracy on the test set and training set.

Plotting the loss and accuracy can be used to see if you are over fitting (you experience tiny loss on the training set, but large loss on the test set) and to see when your training has plateaued.



PROBLEM STATEMENT:

A number of major differences exist between spam-filtering in text messages and emails. Unlike emails, which have a variety of large datasets available, real databases for SMS spams are very limited. Additionally, due to the small length of text messages, the number of features that can be used for their classification is far smaller than the corresponding number in emails. Here, no header exists as well. Additionally, text messages are full of abbreviations and have much less formal language that what one would expect from emails. All of these factors may result in serious degradation in performance of major email spam filtering algorithms applied to short text messages.

1.5 OBJECTIVE STATEMENT:

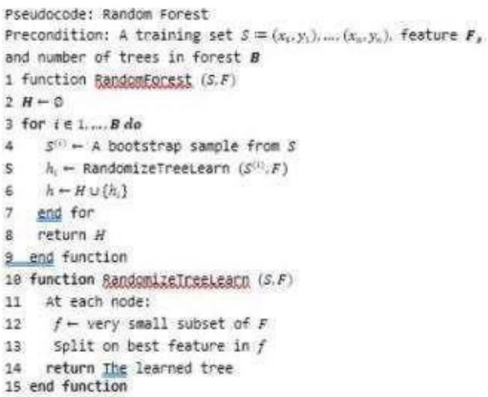
Prediction of SMS spam has been an important area of research for a long time. the goal is to apply different machine learning algorithms to SMS spam classification problem, compare their performance to gain insight and further explore the problem, and design an application based on one of these algorithms that can filter SMS spams with high accuracy. The current work proposes a gamut of machine learning and deep learningbased predictive models for accurately predicting the sms spam movement. The predictive power of the models is further enhanced by introducing the powerful deep learning-based long- and short-term memory (LSTM) network into the predictive framework.

SCOPE OF THE PROJECT:

1. The Proposed mode is based on the study of sms text data and technical indicators. Algorithm selects best free parameters combination for LSTM to avoid over-fitting and local minima problems and improve prediction accuracy.
2. Our dataset consists of one large text file in which each line corresponds to a text message. Therefore, preprocessing of the data, extraction of features, and tokenization of each message is required. After the feature extraction, an initial analysis on the data is done using label encoder and then the models like naive Bayes (NB) algorithm and LSTM are used on next steps are for prediction.
3. The two methods used to predict the spam messages that are Fundamental and technical analyses. .

1.6 EXISTING SYSTEM

Random Forest (RF) algorithm will used for classification of ham or spam during this phase. RF is averaging ensemble learning method that can be used for classification problem. This algorithm combines various decision tree models in order to eliminate the over fitting problem in decision trees. In RF algorithm, each tree is capable in providing its own prediction results, different from each other. As a result, each tree gives different performances, in which the average of their performances will be generalized and calculated. During the training phase, a set of decision trees will be constructed before they can operate on randomly selected features. Regardless, RF can work well with a large dataset with a variety of feature types, similar to binary, categorical and numerical.
The algorithm works as follows diagram: for each tree in the forest, a bootstrap sample is selected from S where S (i) represents the ith bootstrap. A decision-tree is then learn using a modified decision-tree learning algorithm.



```
Pseudocode: Random Forest
Precondition: A training set $S := (x_1, y_1), \dots, (x_n, y_n)$, feature $F$,
and number of trees in forest $B$
function RandomForest ($S, F$)
$H \leftarrow \emptyset$
for $i \in 1, \dots, B$ do
    $S^{(i)} \leftarrow$ A bootstrap sample from $S$
    $h_i \leftarrow$ RandomizeTreeLearn ($S^{(i)}, F$)
    $h \leftarrow H \cup \{h_i\}$
  end for
  return $H$
end function
function RandomizeTreeLearn ($S, F$)
  At each node:
    $f \leftarrow$ very small subset of $F$
    Split on best feature in $f$
  return The learned tree
end function
```

The algorithm is modified as follows: at each node of the tree, instead of examining all possible feature-splits, some subset of the features text f ⊆ F is selected randomly. where

F is the set of Spam features. The node then splits on the best feature in f rather than F. In practice f is much, much smaller than F. Deciding on which feature to split is oftentimes the most computationally expensive aspect of decision tree learning. By narrowing the set of features, the speed up the learning of the tree is increase drastically.

EXISTING TECHINIQUE:

KNN and Random forest.

TECHNIQUE DEFINITION:

K-Nearest Neighbors (KNN)

The KNN algorithm classifies new data based on the class of the k nearest neighbors. This paper uses the value of k as 6. The distance from neighbors can be calculated using various distance metrics, such as Euclidean distance, Manhattan distance (used in this paper), Minkowski distance, etc. The class of the new data may be decided by majority vote or by an inverse proportion to the distance computed. KNN is a nongeneralizing method, since the algorithm keeps all of its training data in memory, possibly transformed into a fast indexing structure such as a ball tree or a KD tree.

Random Forest (RF):

Random forest algorithm is a supervised learning algorithm that is developed to solve the problems of regression and classification. So, the main advantage of random forest is that they can handle both numerical and categorical data. Like other conventional algorithms decision tree algorithm creates a training model and that training model is used to predict the value or class of the target label/variable but here this is done by learning decision rules inferred from previous training dataset. This algorithm makes use of tree structure in which the internal nodes also known as decision node refers to an attribute and each internal node has two or more leaf nodes which corresponds to a class label. The topmost node known as root node corresponds to the best predictor i.e. best attribute of the dataset. This algorithm splits the whole data-frame into parts or subsets and simultaneously a random forest is developed and the end result of this is a tree with leaf nodes, internal nodes and a root node. As the tree becomes more deep and more complex, then the model becomes more and more fit.

1.6.1 DISADVANTAGES OF EXISTING SYSTEM

- Accuracy is low.
- Dataset selection is not correct
- Feature extraction is not accurate
- Complex structure, not easy to understand.

1.6.2 LITERATURE SURVEY

TITLE: SMS Spam Detection Based on Long Short-Term Memory and Gated Recurrent Unit

Author: Pumrapee Poomka, Wattana Pongsena, Nittaya Kerdprasop, and Kittisak Kerdprasop YEAR: - 2019

Abstract:

An SMS spam is the message that hackers develop and send to people via mobile devices targeting to get their important information. For people who are ignorant, if they follow the instruction in the message and fill their important information, such as internet banking account in a faked website or application, the hacker may get the information.

This may lead to loss their wealth. The efficient spam detection is an important tool in order to help people to classify whether it is a spam SMS or not. In this research, we propose a novel SMS spam detection based on the case study of the SMS spams in English language using Natural Language Process and Deep Learning techniques. To prepare the data for our model development process, we use word tokenization, padding data, truncating data and word embedding to make more dimension in data. Then, this data is used to develop the model based on Long Short-Term Memory and Gated Recurrent Unit algorithms. The performance of the proposed models is compared to the models based on machine learning algorithms including Support Vector Machine and Naïve Bayes. The experimental results show that the model built from the Long ShortTerm Memory technique provides the best overall accuracy as high as 98.18%. On accurately screening spam messages, this model shows the ability that it can detect spam messages with the 90.96% accuracy rate, while the error percentage that it misclassifies a normal message as a spam message is only 0.74%.

TITLE: SMS Spam Message Detection using Term Frequency-Inverse Document Frequency and Random Forest Algorithm

Author: Haslina Md Sarkan, Yazriwati Yahya, Suriani Mohd Sam YEAR:

- 2019 Abstract:

The daily traffic of Short Message Service (SMS) keeps increasing. As a result, it leads to dramatic increase in mobile attacks such as spammers who plague the service with spam messages sent to the groups of recipients. Mobile spams are a growing problem as the number of spams keep increasing day by day even with the filtering systems. Spams are defined as unsolicited bulk messages in various forms such as unwanted advertisements, credit opportunities or fake lottery winner notifications. Spam classification has become more challenging due to complexities of the messages imposed by spammers. Hence, various methods have been developed in order to filter spams. In this study, methods of term frequency-inverse document frequency (TF-IDF) and Random Forest Algorithm will be applied on SMS spam message data collection. Based on the experiment, Random Forest algorithm outperforms other algorithms with an accuracy of 97.50%

TITLE: Spam Detection Approach for Secure Mobile Message Communication Using Machine Learning Algorithms

Author: Shah Nazir,2 Habib Ullah Khan,3 and Amin Ul Haq
YEAR: - 2020 Abstract:

The spam detection is a big issue in mobile message communication due to which mobile message communication is insecure. In order to tackle this problem, an accurate and precise method is needed to detect the spam in mobile message communication. We proposed the applications of the machine learning-based spam detection method for accurate detection. In this technique, machine learning classifiers such as Logistic regression (LR), K-nearest neighbor (K-NN), and decision tree (DT) are used for classification of ham and spam messages in mobile device communication. The SMS spam collection data set is used for testing the method. The dataset is split into two categories for training and testing the research. The results of the experiments demonstrated that the classification performance of LR is high as compared with K-NN and DT, and the LR achieved a high accuracy of 99%. Additionally, the proposed method performance is good as compared with the existing state-of-the-art methods.

TITLE: SMS Spam Detection using Machine Learning and Deep Learning Techniques

Author: Sridevi Gadde

YEAR: - 2021

Abstract:

The number of people using mobile devices increasing day by day.SMS (short message service) is a text message service available in smartphones as well as basic phones. So, the traffic of SMS increased drastically. The spam messages also increased. The spammers try to send spam messages for their financial or business benefits like market growth, lottery ticket information, credit card information, etc. So, spam classification has special attention. In this paper, we applied various machine learning and deep learning techniques for SMS spam detection. we used a dataset from UCI and build a spam detection model. Our experimental results have shown that our LSTM model outperforms previous models in spam detection with an accuracy of 98.5%. We used python for all implementations.

TITLE: SMS Spam Detection using Machine Learning Approach

Author: Houshmand Shirani-Mehr, hshirani@stanford.edu

YEAR: - 2019

Abstract:

Over recent years, as the popularity of mobile phone devices has increased, Short Message Service (SMS) has grown into a multi-billion dollars industry. At the same time, reduction in the cost of messaging services has resulted in growth in unsolicited commercial advertisements (spams) being sent to mobile phones. In parts of Asia, up to 30% of text messages were spam in 2012. Lack of real databases for SMS spams, short length of messages and limited features, and their informal language are the factors that may cause the established email filtering algorithms to underperform in their classification. In this project, a database of real SMS Spams from UCI Machine Learning repository is used, and after preprocessing and feature extraction, different machine learning techniques are applied to the database. Finally, the results are compared and the best algorithm for spam filtering for text messaging is introduced. Final simulation results using 10-fold cross validation shows the best classifier in this work reduces the overall error rate of best model in original paper citing this dataset by more than half.

1.7 PROPOSED SYSTEM

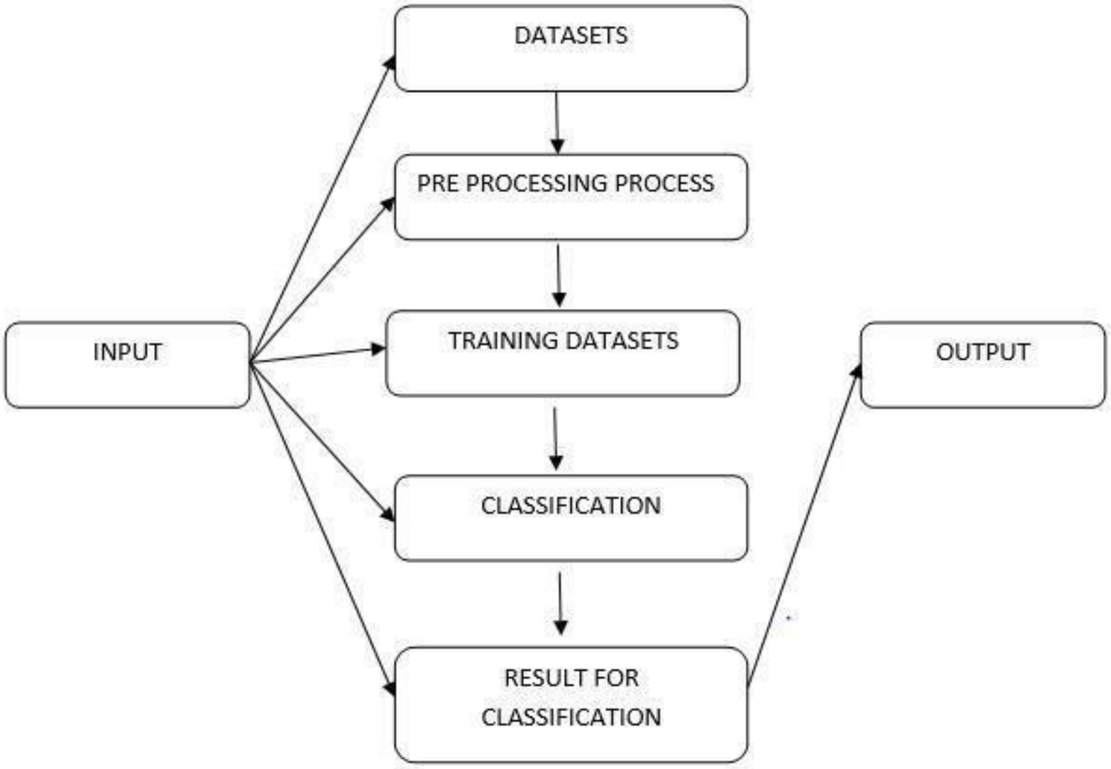
Applying NB algorithm to the dataset using extracted features with different training set sizes. The performance in learning curve is evaluated by splitting the dataset into 70% training set and 30% test set. The NB algorithm shows good overall accuracy.

We notice that the length of the text message (number of characters used) is a very good feature for the classification of spams. Sorting features based on their mutual information (MI) criteria shows that this feature has the highest MI with target labels. Additionally, going through the misclassified samples, we notice that text messages with length below a certain threshold are usually hams, yet because of the tokens corresponding to the alphabetic words or numeric strings in the message they might be classified as spams.

By looking at the learning curve, we see that once the NB is trained on features extracted, the training set error and test set error are close to each other. Therefore, we do not have a problem of high variance, and gathering more data may not result in much improvement in the performance of the learning algorithm. As the result, we should try reducing bias to improve this classifier. This means adding more meaningful features to the list of tokens can decrease the error rate, and is the option that is explored next.

1.7.1 PROPOSED SYSTEM ADVANTAGES

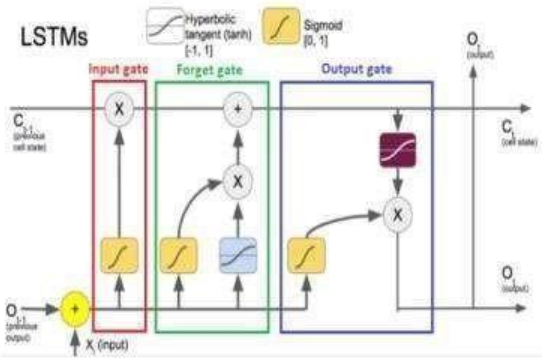
- Complexity is less compared to previous process □ Ability to learn and extract complex features.
- Accuracy is good
- With its simplicity and fast processing time, the proposed algorithm gives better execution time.
- Both machine learning and deep learning technique is performed to predict the value effectively.
- Prediction is accurate



Algorithms:

LSTM (Long Short Term Memory):

A special type of RNN, which can learn long-term dependence, is called Long-Short Term Memory (LSTM). LSTM enables RNN to remember long-term inputs. Contains information in memory, similar to computer memory. It is able to read, write and delete information in its memory. This memory can be seen as a closed cell, with a closed description, the cell decides to store or delete information. In LSTM, there are three gates: input, forget and exit gate. These gates determine whether new input (input gate) should be allowed, data deleted because it is not important (forget gate), or allow it to affect output at current timeline (output gate)



LSTM's are a special subset of RNN‘s that can capture context-specific temporal dependencies for long periods of time. Each LSTM neuron is a memory cell that can store other information i.e., it maintains its own cell state. While neurons in normal RNN‘s merely take in their previous hidden state and the current input to output a new hidden state, an LSTM neuron also takes in its old cell state and outputs its new cell state.

1. **Forget gate:** The forget gateway determines when certain parts of the cell will be inserted with information that is more recent. It subtracts almost 1 in parts of the cell state to be kept, and zero in values to be ignored.

2. **Input gate :** Based on the input (e.g., previous output o (t-1), input x (t), and the previous state of cell c (t-1)), this network category reads the conditions under which any information should be stored (or updated) in the state cell.

3. **Output gate:** Depending on the input mode and the cell, this component determines which information is forwarded in the next location in the network.

Thus, LSTM networks are ideal for exploring how variation in one unwanted sms can affect the users over a long period of time. They can also decide (in a dynamic fashion) for how long information about specific past trends in spam sms movement needs to be retained in order to more accurately predict future trends in the variation of spam sms.

Advantages of LSTM :

The main advantage of LSTM is its ability to read intermediate context. Each unit remembers details for a long or short period without explicitly utilizing the activation function within the recurring components. An important fact is that any cell state is repeated only with the release of the forget gate, which varies between 0 and 1. That is to say, the gateway for forgetting in the LSTM cell is responsible for both the hardware and the function of the cell state activation. Thus, the data from the previous cell can pass through the unchanged cell instead of explicitly increasing or decreasing in each step or layer, and the instruments can convert to their appropriate values over a limited time. This allows LSTM to solve a perishable gradient problem - because the amount stored in the memory cell is not converted in a recurring manner, the gradient does not end when trained to distribute backwards.

CHAPTER 2

2.1 INTRODUCTION

The increasing mobile phones become one of the attached companions for many individuals. With the explosive penetration of mobile devices and millions of people sending messages every day, Short Message Service (SMS) has become a multimilliondollar commercial industry with a value between 11.3 to 24.7 percent of the developing countries‘ Gross National Income (GNI) in the early year of 2013.

As the utilization of mobile phone devices has become commonplace, Short Message Service (SMS) has grown into a multi-billion dollars commercial industry [2]. SMS is a text communication platform that allows mobile phone users to exchange short text messages (usually less than 160 seven-bit characters). It is the most widely used data application with an estimated 3.5 billion active users, or about 80% of all mobile phone subscribers at the end of 2010 [3]. As the popularity of the platform has increased, we have seen a surge in the number of unsolicited commercial advertisements sent to mobile phones using text messaging. SMS spam is still not as common as email spam, where in 2010 around 90% of emails was spam, and in North America it is still not a major problem, contributing to less than 1% of text messages exchanged as of December 2012.

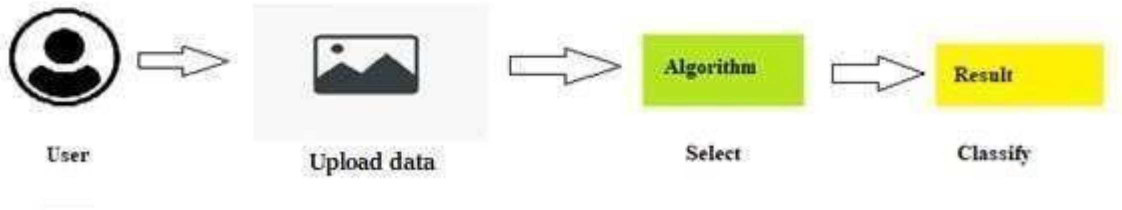
The spam increased in these days due more mobile devices deployed in environment for e-mail and message communication. Currently, 85% of mails and messages received by mobile users are spam. The cost of mails and messages are very low for senders but high for receipts of these messages. The cost paid some time by service providers and the cost of spam can be measured in the loss of human time and loss of important messages or mails. Due to these spam mails and messages, the values able e-mails and messages are affected because each user have limited Internet services, short time, and memory.

The dataset is a large text file, in which each line starts with the label of the message, followed by the text message string. After preprocessing of the data and extraction of features, machine learning techniques such as naive Bayes, SVM, and other methods are applied to the samples, and their performances are compared. Finally, the performance of best classifier from the project is compared against the performance of classifiers applied in the original paper citing this dataset.

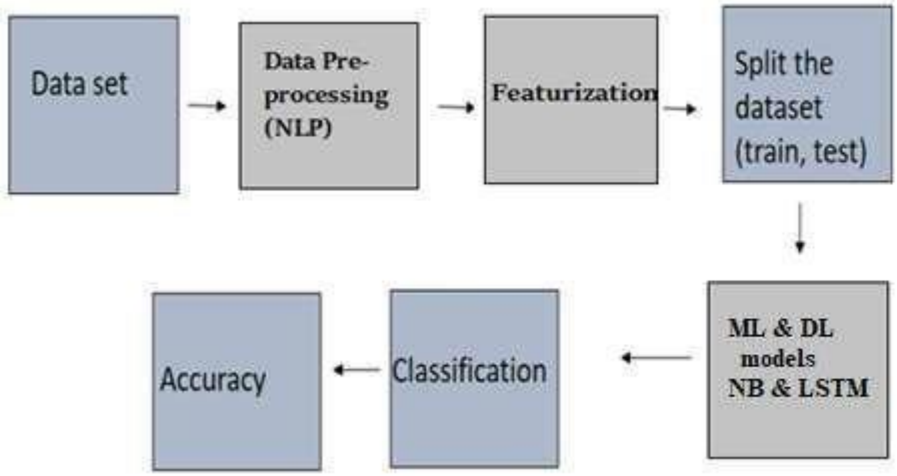
We proposed a spam detection method using machine learning algorithms such as NB (naïve Bayes) and LSTM for classification of ham and spam messages. The SMS spam collection dataset was considered for testing of the current research. The dataset was divided into two categories: 30% for testing and 70% for training purpose for the predictive models. The evaluation metrics for performance such as specificity, accuracy, and sensitivity were considered evaluating the proposed study. The results obtained from experiments confirmed that the proposed research achieved high accuracy.

2.2 DETAILED DIAGRAM

2.2.1Front End Module Diagrams:



2.2.2 Back End Module Diagrams:



2.3 SYSTEM SPECIFICATION:

2.3.1 HARDWARE REQUIREMENTS:

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and not how it should be implemented

PROCESSOR	:	Intel I5
RAM	:	4GB
HARD DISK	:	40 GB

2.3.2 SOFTWARE REQUIREMENTS:

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team‘s and tracking the team‘s progress throughout the development activity.

PYTHON IDE	**:**	Anaconda Jupyter Notebook
PROGRAMMING LANGUAGE	:	Python

2.4 MODULE DESCRIPTION

Data Collection:

The public dataset of SMS labelled messages is obtained from UCI Machine Learning Repository. The dataset considered in the current research is available on kaggle, a machine learning repository. This study finds that there are only 5,574 labelled messages in the dataset, with 4827 of messages belong to ham messages while the other 747 messages belong to spam messages. Nonetheless, this dataset consists of two named columns starting with the message labels (ham or spam) followed by strings of text messages and three unnamed columns.

It‘s time for a data analyst to pick up the baton and lead the way to machine learning implementation. The job of a data analyst is to find ways and sources of collecting relevant and comprehensive data, interpreting it, and analyzing results with the help of statistical techniques.

The type of data depends on what you want to predict.

There is no exact answer to the question ―How much data is needed?‖ because each machine learning problem is unique. In turn, the number of attributes data scientists will use when building a predictive model depends on the attributes' predictive value.

‗The more, the better' approach is reasonable for this phase. Some data scientists suggest considering that less than one-third of collected data may be useful. It's difficult to estimate which part of the data will provide the most accurate results until the model training begins. That's why it's important to collect and store all data — internal and open, structured and unstructured.

The purpose of preprocessing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling. ***Data formatting:***The importance of data formatting grows when data is acquired from various sources by different people. The first task for a data scientist is to standardize record formats. A specialist checks whether variables representing each attribute are recorded in the same way. Titles of products and services, prices, date formats, and addresses are examples of variables. The principle of data consistency also applies to attributes represented by numeric ranges.

Data cleaning: **-** This set of procedures allows for removing noise and fixing inconsistencies in data. A data scientist can fill in missing data using imputation techniques, e.g. substituting missing values with mean attributes. A specialist also detects outliers — observations that deviate significantly from the rest of distribution. If an outlier indicates erroneous data, a data scientist deletes or corrects them if possible. This stage also includes removing incomplete and useless data objects.

Data anonymization: - Sometimes a data scientist must anonymize or exclude attributes representing sensitive information (i.e. when working with healthcare and banking data).

Data sampling**: -** Big datasets require more time and computational power for analysis. If a dataset is too large, applying data sampling is the way to go. A data scientist uses this technique to select a smaller but representative data sample to build and run models much faster, and at the same time to produce accurate outcomes.

Pre-processing is the first stage in which the unstructured data is converted into more structured data. Since keywords in SMS text messages are prone to be replaced by symbols. In this study, the stop word list remover for English language have been applied to eliminate the stop words in the SMS text messages.

Natural Language Processing NLP

The input given by the user is processed through a number of stages to understand what the user is trying to say. Natural language processing (NLP) is the ability of a program to make use of the natural language spoken by a human and comprehend it‘s meaning. NLP is the study of the computational treatment of natural (human) language. The development of NLP is challenging because computers are used to of getting a highly structured input whereas natural language is highly complex and ambiguous with different linguistic structures and complex variables. NLP has various stages as follows:

- Tokenisation(lexical analysis), also referred as segmentation involves breaking up a sentence or paragraph into tokens or individual words, numbers or meaning full phrases. Tokens can be thought of as a small part like a word is a token in a sentence and a sentence is a token in a paragraph. The words are separated with the help of word boundaries. English is space delimited hence, word boundaries are the space between ending of one word and starting of the next one. Example: ―I am suffering with fever!‖ The output after tokenisation would be: [‗I‘ ,‘am‘ , ‗suffering‘ , ‗with‘ , ‗ fever‘] • Syntactic analysis involves analysis of words for grammar and putting the words together in a manner which can show their relationship. This can be done with a data structure such as a parse tree or syntax tree. The tree is constructed with the rules of grammar of the language. If the input can be produced using the syntax tree the input is found to have correct syntax. For example the string ―I pick that have to‖ will be considered incorrect syntax.

- Semantic analysis picks up the dictionary meaning of words and tries to understand the actual meaning of the sentence. It is the process of mapping syntactic structures with the actual or text independent meaning of the words. Strings like ―hot winter‖ will be disregarded.

- Pragmatic analysis: Pragmatic investigation manages outside word information, which implies learning the outer to the archives and additionally inquiries. Pragmatics analysis that centers around what was portrayed reinterpreted by what it really implied, inferring the different parts of language that require true learning.

Featuraization:-

Featuraization is a way to change some form of data (text data, graph data, time-series data…) into a numerical vector.

Featuraization is different from feature engineering. Feature engineering is just transforming the numerical features somehow so that the machine learning models work well. In feature engineering, features are already in the numerical form. Whereas in Featuraization data not need to be in the form of numerical vector.

The machine learning model cannot work with row text data directly. In the end, machine learning models work with numerical (categorical, real…) features. So it is import to change some type of data into numerical vector so that we can leverage the whole power of linear algebra (making the decision boundary between data points) and statistics tools with other types of data also.

Feature extraction and selection is important for the discrimination of ham and spam in SMS text messages. For this phases TFIDF will be used. TFIDF is the often-weighting method used to in the Vector Space Model, particularly in IR domain including text mining. It is a statistical method to measure the important of a word in the document to the whole corpus. The term frequency is simply calculated in proportion to the number of occurrences a word appears in the document and usually normalized in positive quadrant between 0 and 1 to eliminate bias towards lengthy documents. To construct the index of terms in TFIDF, punctuation is removed, and all text are lowercase during tokenization. The first two letter TF or term frequency refers to how important if it occurs more frequently in a document. Therefore, the higher TF reflects to the more estimated that the term is significant in respective documents. Additionally, IDF or Inverse Document Frequency calculated on how infrequent a word or term is in the documents. The weighted value is estimated using the whole training dataset. The idea of IDF is that a word is not considered to be good candidate to represent the document if it is occurring frequently in the whole dataset as it might be the stop words or common words that is generic. Hence only infrequent words in contrast of the entire dataset is relevant for that documents. TFIDF does not only assess the importance of words in the documents but it also evaluates the importance of words in document database or corpus. In this sense, the word frequency in the document will increase the weight of words proportionally but will then be offset by corpus‘s word frequency. This key characteristic of TF-IDF assumes that there are several words that appear more often compared to others in the document in general.

$$F - IDF = \frac{Ter \quad Frequency}{Document\ Frequency}$$

Splitting of data

After cleaning the data, data is normalized in training and testing the model. When data is spitted then we train algorithm on the training data set and keep test data set aside. This training process will produce the training model based on logic and algorithms and values of the feature in training data. Basically aim of feature extraction is to bring all the values under same scale.

A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

***Training set*: -**A data scientist uses a training set to train a model and define its optimal parameters — parameters it has to learn from data.

***Test set:* -** A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model‗s ability to identify patterns in new unseen data after having been trained over a training data. It‗s crucial to use different subsets for training and testing to avoid model over fitting, which is the incapacity for generalization we mentioned above.

Modeling:-

During this stage, a *data scientist* trains numerous models to define which one of them provides the most accurate predictions. **Model Testing:-**

The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning.

That‗s the optimization of model parameters to achieve an algorithm‗s best performance.

One of the more efficient methods for model evaluation and tuning is cross-validation

***Cross-validation:* -** Cross-validation is the most commonly used tuning method. It entails splitting a training dataset into ten equal parts (folds). A given model is trained on only nine folds and then tested on the tenth one (the one previously left out). Training continues until every fold is left aside and used for testing. As a result of model performance measure, a specialist calculates a cross-validated score for each set of hyper parameters. A data scientist trains models with different sets of hyper parameters to define which model has the highest prediction accuracy. The cross-validated score indicates average model performance across ten hold-out folds. Then a data science specialist tests models with a set of hyper parameter values that received the best cross-validated score. There are various error metrics for machine learning tasks.

NAIVE BAYES:

Naive Bayes is a simple probabilistic classifier based on applying Bayes' theorem (or Bayes‘s rule) with strong independence (naive) assumptions. Parameter estimation for Naïve Bayes models uses the maximum likelihood estimation. It takes only one pass over the training set and is computationally very fast.

- ***Bayes Rule***

A conditional probability is the likelihood of some conclusion, C, given some evidence/observation, D, where a dependence relationship exists between C and D.

This probability is denoted as

$(C|D)$ **where,** $(D/C) = [(D/C)(C)] / [P(D)]$

- ***NB Classifier***

Naïve Bayes classifier is one of the high detection approach for learning classification of text documents. Given a set of classified training samples, an application can learn from these samples, so as to predict the class of an unmet samples.

The features ($n1$, $n2$, $n3$, $n4$) which are present in sms are independent from each other. Every feature ($1 \leq i \leq 4$) text binary value showing whether the particular property comes in sms. The probability is calculated that the given web belongs to a class r ($r1$: spam and $r2$: Ham) as follows:

$(r1/N) = ((r1) * P(N/ri))/P(N)$

PERFORMANCE MATRICES:

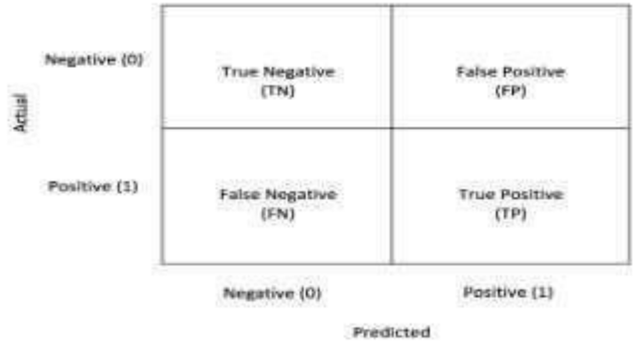
Data was divided into two portions, training data and testing data, both these portions consisting 70% and 30% data respectively. All these two algorithms were applied on same dataset using Enthought Canaopy and results were obtained.

$$\text{Accuracy} = (TP+TN) / (P + N)$$

Predicting accuracy is the main evaluation parameter that we used in this work. Accuracy can be defied using equation. Accuracy is the overall success rate of the algorithm.

CONFUSION MATRIX:

It is the most commonly used evaluation metrics in predictive analysis mainly because it is very easy to understand and it can be used to compute other essential metrics such as accuracy, recall, precision, etc. It is an NxN matrix that describes the overall performance of a model when used on some dataset, where N is the number of class labels in the classification problem.



Actual			
	Negative (0)	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)
		Negative (0)	Positive (1)
		Predicted	

All predicted true positive and true negative divided by all positive and negative. True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP) predicted by all algorithms are presented in table.

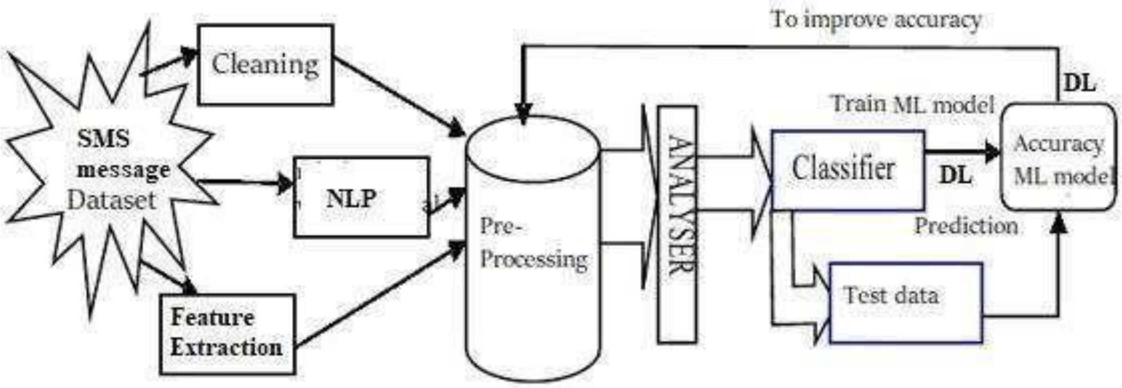
True positive (TP) indicates that the positive class is predicted as a positive class, and the number of sample positive classes was actually predicted by the model.

False negative indicates (FN) that the positive class is predicted as a negative class, and the number of negative classes in the sample was actually predicted by the model. False positive (FP) indicates that the negative class is predicted as a positive class, and the number of positive classes of samples was actually predicted by the model. True negative (TN) indicates that the negative class is predicted as a negative class, and the number of sample negative classes was actually predicted by the model.

2.5 SYSTEM DESIGN:

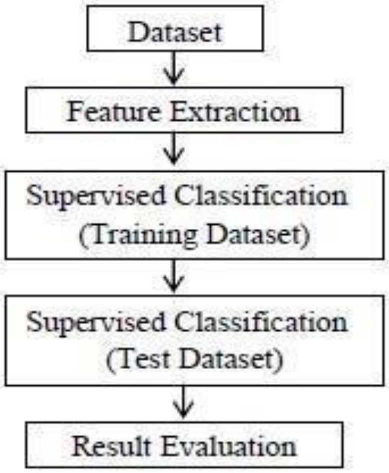
Designing of system is the process in which it is used to define the interface, modules and data for a system to specified the demand to satisfy. System design is seen as the application of the system theory. The main thing of the design a system is to develop the system architecture by giving the data and information that is necessary for the implementation of a system.

2.5.1 ARCHITECTURE DIAGRAM:



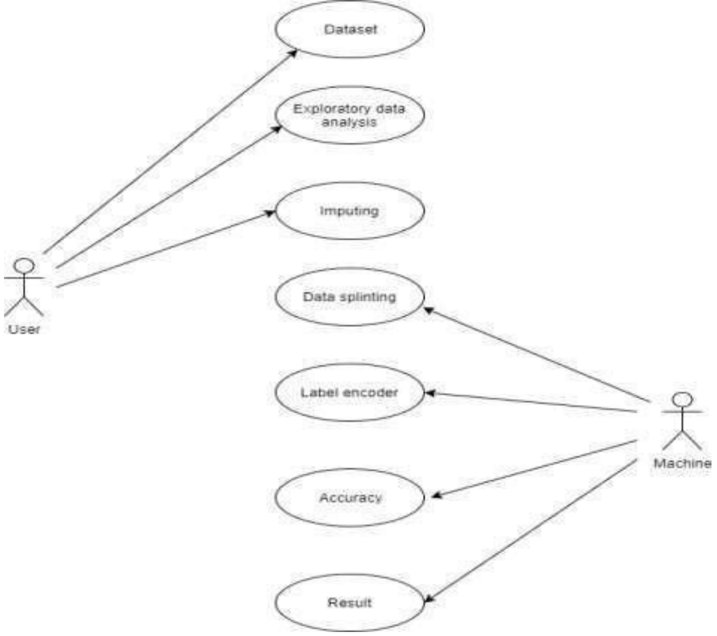
2.5.2 DATA FLOW DIAGRAM:

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow


Dataset
Feature Extraction
Supervised Classification (Training Dataset)
Supervised Classification (Test Dataset)
Result Evaluation

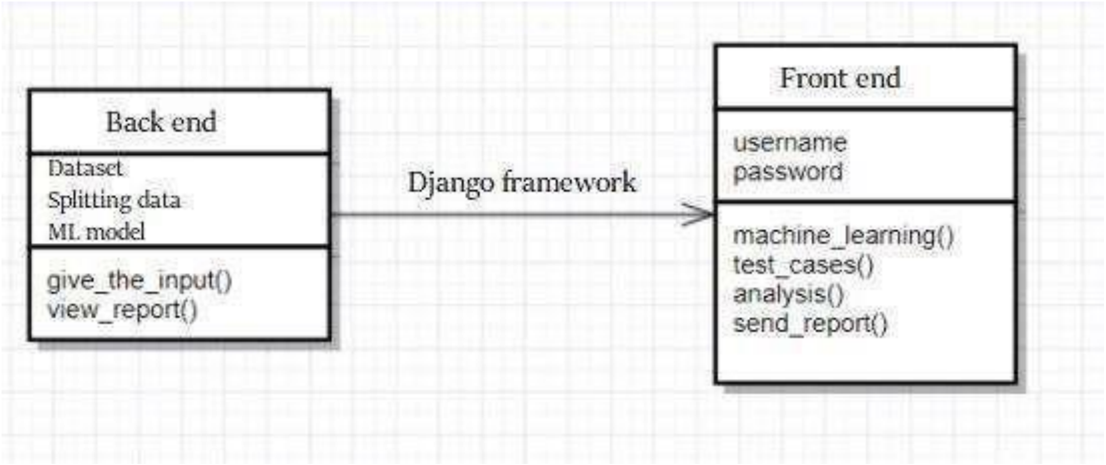
2.5.3 USECASE DIAGRAM:

Use case diagrams identify the functionalities provides by the use cases, the actors who interact with the system and the association between the actors and the functionalities.



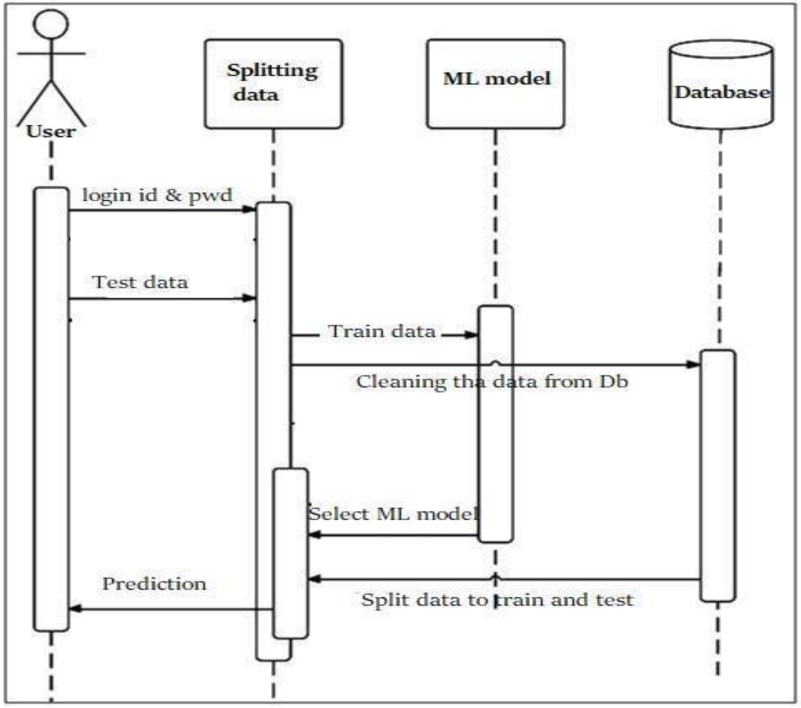
2.5.4 CLASS DIAGRAM:

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application



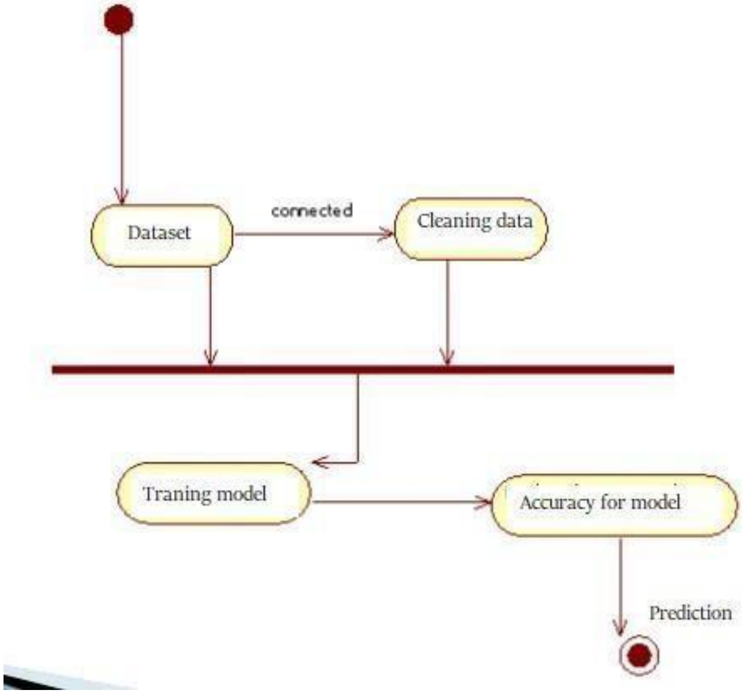
2.5.5 SEQUENCE DIAGRAM:

The sequence diagram of a system shows the entity interplay are ordered in the time order level. So, that it drafts the classes and object that are imply in the that plot and also the series of message exchange take place betwixt the body that need to be carried out by the purpose of that scenario.



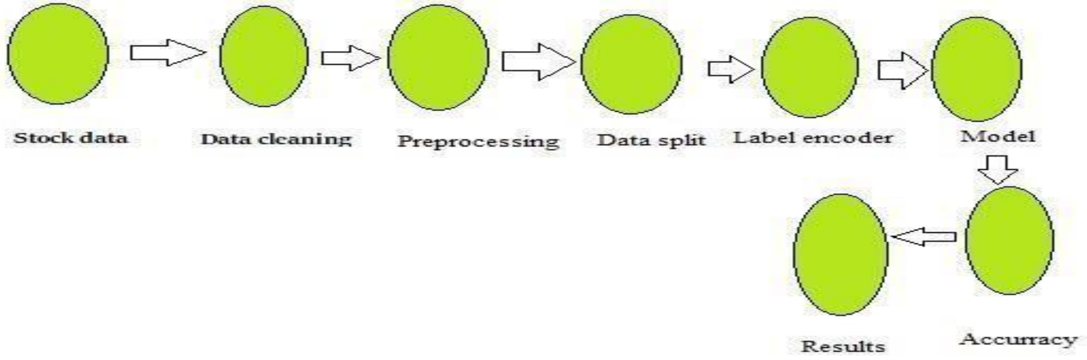
2.5.6 ACTIVITY DIAGRAM:

The Activity Diagram forms effective while modeling the functionality of the system. Hence this diagram reflects the activities, the types of flows between these activities and finally the response of objects to these activities.



2.5.7 STATE FLOW DIAGRAM:

The below state chart diagram describes the flow of control from one state to another state (event) in the flow of the events from the creation of an object to its termination.



CHAPTER 3

SOFTWARE SPECIFICATION 3.1 GENERAL

ANACONDA

It is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

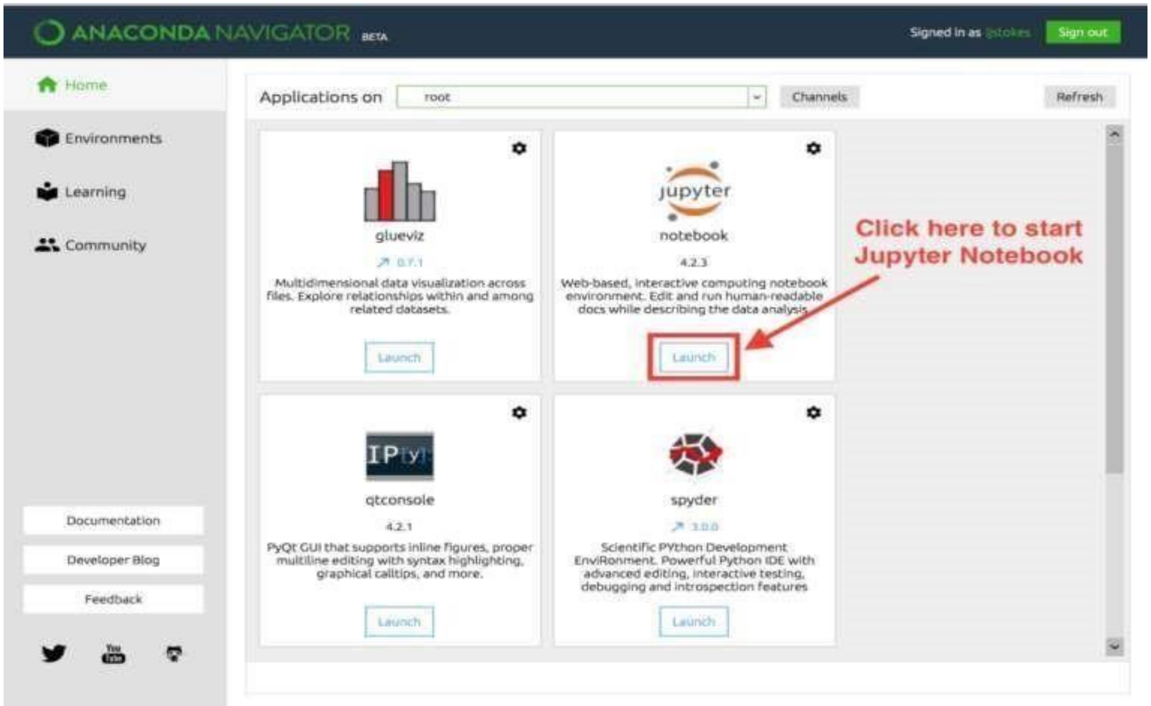
Anaconda distribution comes with more than 1,500 packages as well as the Conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the Command Line Interface (CLI).

The big difference between Conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason Conda exists. Pip installs all Python package dependencies required, whether or not those conflict with other packages you installed previously.

So your working installation of, for example, Google Tensorflow, can suddenly stop working when you pip install a different package that needs a different version of the

Numpy library. More insidiously, everything might still appear to work but now you get different results from your data science, or you are unable to reproduce the same results elsewhere because you didn't pip install in the same order.

Conda analyzes your current environment, everything you have installed, any version limitations you specify (e.g. you only want tensorflow>= 2.0) and figures out how to install compatible dependencies. Or it will tell you that what you want can't be done. Pip, by contrast, will just install the thing you wanted and any dependencies, even if that breaks other things.Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or your own private repository or mirror, using the conda install command. Anaconda Inc compiles and builds all the packages in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. You can also install anything on PyPI into a Conda environment using pip, and Conda knows what it has installed and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories.The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.



Multidimensional data visualization across files. Explore relationships within and among related datasets.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Click here to start Jupyter Notebook

Anaconda Navigator is a desktop Graphical User Interface (GUI) included in Anaconda distribution that allows users to launch applications and 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, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

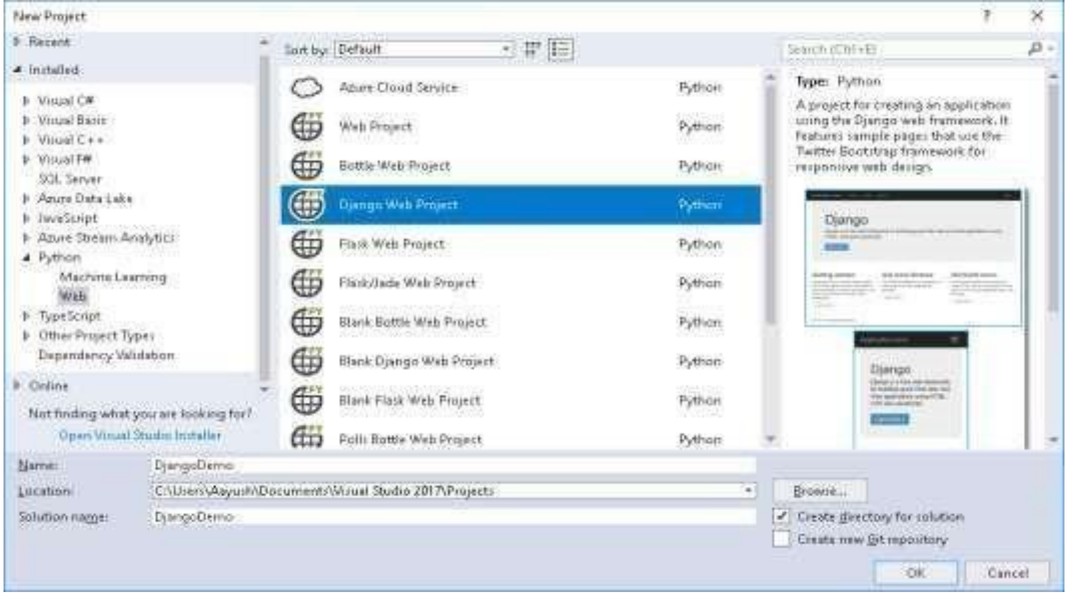
The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QtConsole
- Spyder
- Glueviz
- Orange
- Rstudio
- Visual Studio Code

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There‛s no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

—.NET‖ is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

Microsoft VISUAL STUDIO is an Integrated Development Environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps.



New Project

Sort by: Default

Type: Python

A project for creating an application using the Django web framework. It features sample pages that use the Twitter Bootstrap framework for responsive web design.

Name: DjangoDemo

Location: C:\Users\Aayush\Documents\Visual Studio 2017\Projects

Solution name: DjangoDemo

☑ Create directory for solution

☐ Create new Git repository

Recent
Installed
Visual C#
Visual Basic
Visual C++
Visual F#
SQL Server
Azure Data Lake
JavaScript
Azure Stream Analytics
Python
Machine Learning
Web
TypeScript
Other Project Types
Dependency Validation
Online
Not finding what you are looking for?
Open Visual Studio Installer
Search (Ctrl+E)
Azure Cloud Service Python
Web Project Python
Bottle Web Project Python
Django Web Project Python
Flask Web Project Python
Flask/Jade Web Project Python
Blank Bottle Web Project Python
Blank Django Web Project Python
Blank Flask Web Project Python
Polls Bottle Web Project Python
Django
Django
Browse...
OK
Cancel

Python is a powerful multi-purpose programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. Python features are:

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted
- Large Standard Library
- Dynamically Typed Language

3.3 PYTHON:

- Python is a powerful multi-purpose programming language created by Guido van Rossum.
- It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. Features Of Python :

1.Easy to code:

Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.

2. Free and Open Source:

Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword.

Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.

3.Object-Oriented Language:

One of the key features of python is Object-Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.

4. GUI Programming Support:

Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python.

PyQt5 is the most popular option for creating graphical apps with Python.

5. High-Level Language:

Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6.Extensible feature:

Python is a Extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language:

Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language:

Python is also an Integrated language because we can easily integrated python with other language like c, c++ etc.

9. Interpreted Language:

Python is an Interpreted Language. because python code is executed line by line at a time. like other language c, c++, java etc there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library:

Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing.There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language:

Python is dynamically-typed language. That means the type (for example- int, double, long etc) for a variable is decided at run time not in advance.because of this feature we don‘t need to specify the type of variable.

APPLICATIONS OF PYTHON :

WEB APPLICATIONS

- You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are:Django, Flask, Pyramid, Plone, Django CMS. • Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

3.2.1 SCIENTIFIC AND NUMERIC COMPUTING

- There are numerous libraries available in Python for scientific and numeric computing. There are libraries like:SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on.
- Also, the language is heavily used in machine learning, data mining and deep learning.

3.2.2 CREATING SOFTWARE PROTOTYPES

- Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must.
- However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

• 3.2.3 GOOD LANGUAGE TO TEACH PROGRAMMING

- Python is used by many companies to teach programming to kids It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use system.

CHAPTER 4

4.1 CONCLUSION AND REFERENCES

The SMS spam message problem is plaguing almost every country and keeps increasing without a sign of slowing down as the number of mobile users increase in addition to cheap rates of SMS services. Therefore, this paper presents the spam filtering technique using various machine learning algorithms. Based on the experiment, TF-IDF with Nave bayes classification algorithm outperforms good compare to other algorithm like LSTM in terms of accuracy percentage. However, it is not enough to evaluate the performance based on the accuracy alone since the dataset is imbalanced. After some examinations, NB algorithm still manages to provide good precision and fmeasure with 0.98 of precision while 0.97 for f-measure. Different algorithms will provide different performances and results based on the features used. For future works, adding more features such as message lengths might help the classifiers to train data better and give better performance.

FUTURE SCOPE:

Future scope of this project will involve adding more feature parameter. The more the parameters are taken into account more will be the accuracy. The algorithms can also be applied for analyzing the contents of public comments and thus determine patterns/relationships between the customer and the company. The use of traditional algorithms and data mining techniques can also help predict the corporation performance structure as a whole. In the future, we plan to integrate neural network with some other techniques such as genetic algorithm or fuzzy logic. Genetic algorithm can be used to identify optimal network architecture and training parameters. Fuzzy logic provides the ability to account for some uncertainty produced by the neural network predictions. Their uses in conjunction with neural network could provide an improvement for SMS spam prediction.

APPLICATION:

- It can used for company to prevent users using fake links.
- Hacking can be prevented. **REFERENCES:**

[1] Modupe, A., O. O. Olugbara, and S. O. Ojo. (2014) ―Filtering of Mobile Short Messaging Communication Using Latent Dirichlet Allocation with Social Network Analysis‖, in Transactions on Engineering Technologies: Special Volume of the World Congress on Engineering 2013, G.-C. Yang, S.-I. Ao, and L. Gelman, Eds. Springer Science & Business. pp. 671–686.

[2] Shirani-Mehr, H. (2013) ―SMS Spam Detection using Machine Learning Approach.‖ p. 4.

[3] Abdulhamid, S. M. et al., (2017) ―A Review on Mobile SMS Spam Filtering Techniques.‖ IEEE Access 5: 15650–15666.

[4] Aski, A. S., and N. K. Sourati. (2016) ―Proposed Efficient Algorithm to Filter Spam Using Machine Learning Techniques.‖ Pac. Sci. Rev. Nat. Sci. Eng. 18 (2):145–149.

[5] Narayan, A., and P. Saxena. (2013) ―The Curse of 140 Characters: Evaluating The Efficacy of SMS Spam Detection on Android.‖ p. 33– 42.

[6] Almeida, T. A., J. M. Gómez, and A. Yamakami. (2011) ―Contributions to the Study of SMS Spam Filtering: New Collection and Results.‖ p. 4.

[7] Mujtaba, D. G., and M. Yasin. (2014) ―SMS Spam Detection Using Simple Message Content Features.‖ J. Basic Appl. Sci. Res. 4 (4): 5.

[8] Gudkova, D., M. Vergelis, T. Shcherbakova, and N. Demidova. (2017) ―Spam and Phishing in Q3 2017.‖ Securelist - Kaspersky Lab‗s Cyberthreat Research and Reports. Available from: https://securelist.com/spam-and-phishing-in-q3-2017/82901/. [Accessed: 10th April 2018].

[9] Choudhary, N., and A. K. Jain. (2017) ―Towards Filtering of SMS Spam Messages Using Machine Learning Based Technique‖, in Advanced Informatics for Computing Research 712: 18-30.

[10] Safie, W., N.N.A. Sjarif, N.F.M. Azmi, S.S. Yuhaniz, R.C. Mohd, and S.Y. Yusof. (2018) ―SMS Spam Classification using Vector Space Model and Artificial Neural Network.‖ International Journal of Advances in Soft Computing & Its Applications 10 (3): 129-141.

[11] Fawagreh, Khaled, Mohamed Medhat Gaber, and Eyad Elyan. (2014) ―Random Forests: From Early Developments to Recent Advancements, Systems Science &

Control Engineering.‖ An Open Access Journal 2 (1): 602-609.

[12] Sajedi, H., G. Z. Parast, and F. Akbari. (2016) —SMS Spam Filtering Using Machine Learning Techniques: A Survey.‖ Machine Learning, 1 (1): 14.

[13] Q. Xu, E., W. Xiang, Q. Yang, J. Du, and J. Zhong. (2012) —SMS Spam Detection Using Noncontent Features.‖ IEEE Intell. Syst. 27(6): 44–51.

[14] Sethi, G., and V. Bhootna. (2014) SMS Spam Filtering Application Using Android.

[15] Nagwani, N. K. (2017) —A Bi-Level Text Classification Approach for SMS Spam Filtering and Identifying Priority Messages.‖ 14 (4): 8.

[16] Delany, S. J., M. Buckley, and D. Greene. (2012) —SMS Spam Filtering: Methods and Data,‖ Expert Syst. Appl. 39(10): 9899–9908.

[17] Chan, P. P. K., C. Yang, D. S. Yeung, and W. W. Y. Ng. (2015) —Spam Filtering for Short Messages in Adversarial Environment.‖ Neurocomputing 155: 167–176.

[18] Sethi, P., V. Bhandari, and B. Kohli. (2017) —SMS Spam Detection and Comparison of Various Machine Learning Algorithms‖, in 2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN). pp. 28–31.

[19] Warade, S. J., P. A. Tijare, and S. N. Sawalkar. (2014) —An Approach for SMS Spam Detection.‖ Int. J. Res. Advent Technol. 2 (2): 4.

[20] Almeida. T. A., and J. M. G. Hidalgo. (2018) —SMS Spam Collection.‖ Available from: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/. [Accessed: 11st April 2018].