PROJECT ON SMS SPAM DETECTION

A REPORT

Submitted by

PRATEEK DUBEY

200640116011

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

Information Technology

K. J. INSTITUTE OF ENGINEERING AND TECHNOLOGY

Savli

Gujarat Technological University, Ahmedabad

Team ID: 420772

K. J. INSTITUTE OF ENGINEERING AND TECHNOLOGY

Savli

CERTIFICATE

This is to certify that the Project report submitted along with the project entitled **SMS SPAM DETECTION** has been carried out by **Prateek Dubey (200640116011)** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in Information Technology, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2023-24.

Mrs. Pranavi Patel
Internal Guide

Mr. Ajay Baria
Head of the Department

Team ID: 420772

K. J. INSTITUTE OF ENGINEERING AND TECHNOLOGY

Savli

DECLARATION

We hereby declare that the Project Report submitted along with the Project entitled **SMS Spam Detection** submitted in partial fulfillment for the degree of Bachelor of Engineering in Computer Engineering to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me under and that no part of this report has been directly copied from any student's reports or taken from any other source, without providing due reference.

Name of the Student	Sign of Student
PRATEEK DUBEY	

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to all those who provided me with the possibility of completion of the internship. A special gratitude of thanks I give to our Internal Guide, Mrs. Pranavi Patel, whose contribution to stimulating suggestions and Encouragement, helped me to coordinate the internship especially in drafting this report.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the Head of the Department, Mr. Ajay Baria, who gave the permission to use all required equipment and the necessary material to fulfill the task. Last but not the least, many thanks go to the teachers and my friends and families who have invested their full effort in guiding us in achieving the goal.

Thank You

Prateek Dubey

Enrollment No.: - 200640116011

Sign of Student……………………………

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.

List of Figures

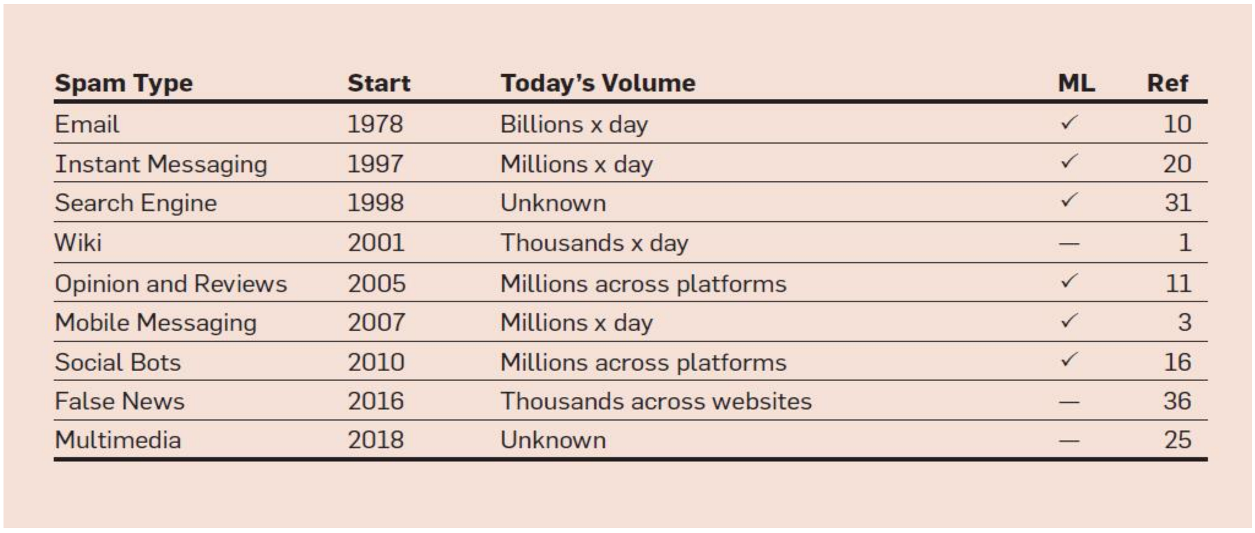
Table of Contents

CHAPTER 1: HISTORY OF DIGITAL SPAM

1.1 HISTORY

Key Insights

- *Throughout the Internet's history, digital spam has pervaded all techno-social platforms and it is constantly evolving. This article provides a taxonomy of digital spam, from its inception to current spam techniques.*
- *Since the email spam epidemic of the early 1990s, new forms of spam have emerged, including search engine spam, fake reviews, spam bots, and false news. In its latest incarnation, spam threats to pollute AI systems making them biased and ultimately dangerous for our society.*
- *By illustrating some of the risks posed by digital spam in all its forms, including AI spam, we provide policy recommendations and technical insights to tackle old and new forms of spam.*



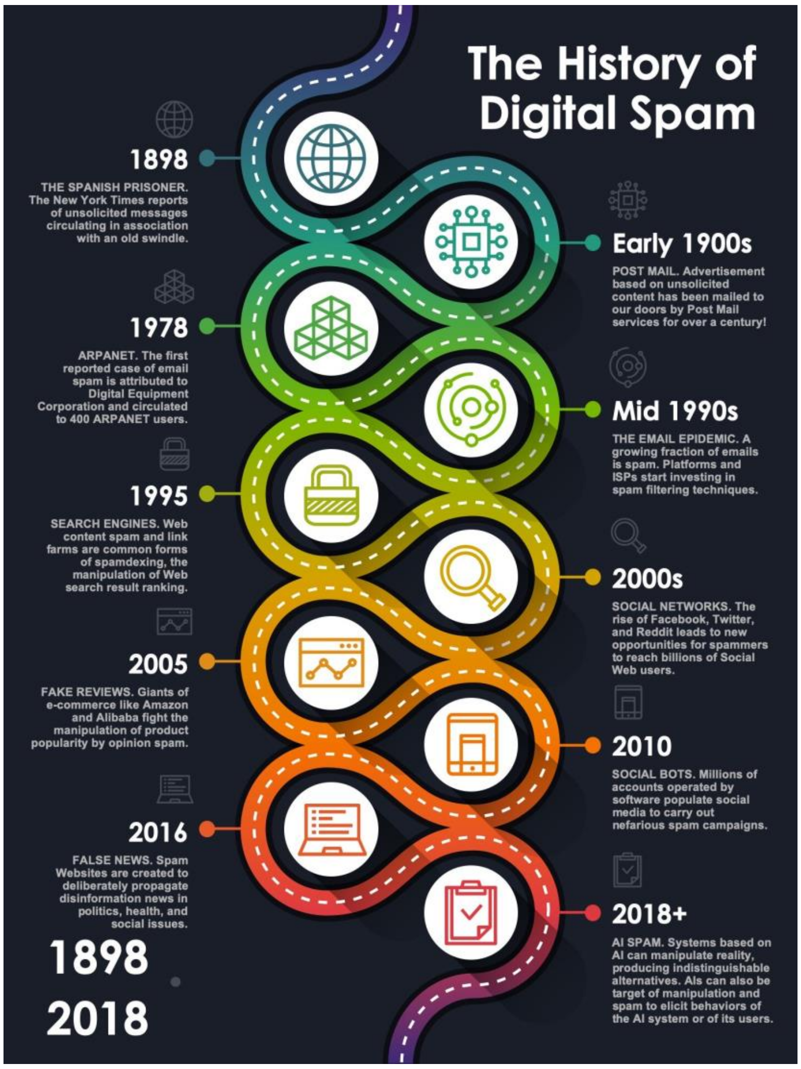
Spam Type	Start	Today's Volume	ML	Ref
Email	1978	Billions x day	✓	10
Instant Messaging	1997	Millions x day	✓	20
Search Engine	1998	Unknown	✓	31
Wiki	2001	Thousands x day	—	1
Opinion and Reviews	2005	Millions across platforms	✓	11
Mobile Messaging	2007	Millions x day	✓	3
Social Bots	2010	Millions across platforms	✓	16
False News	2016	Thousands across websites	—	36
Multimedia	2018	Unknown	—	25

History of Spam Type

Figure 1.1

Figure. Examples of types of spam and relative statistics.

Figure 1.2



1898 THE SPANISH PRISONER. The New York Times reports of unsolicited messages circulating in association with an old swindle.

Early 1900s POST MAIL. Advertisement based on unsolicited content has been mailed to our doors by Post Mail services for over a century!

1978 ARPANET. The first reported case of email spam is attributed to Digital Equipment Corporation and circulated to 400 ARPANET users.

Mid 1990s THE EMAIL EPIDEMIC. A growing fraction of emails is spam. Platforms and ISPs start investing in spam filtering techniques.

1995 SEARCH ENGINES. Web content spam and link farms are common forms of spamdexing, the manipulation of Web search result ranking.

2000s SOCIAL NETWORKS. The rise of Facebook, Twitter, and Reddit leads to new opportunities for spammers to reach billions of Social Web users.

2005 FAKE REVIEWS. Giants of e-commerce like Amazon and Alibaba fight the manipulation of product popularity by opinion spam.

2010 SOCIAL BOTS. Millions of accounts operated by software populate social media to carry out nefarious spam campaigns.

2016 FALSE NEWS. Spam Websites are created to deliberately propagate disinformation news in politics, health, and social issues.

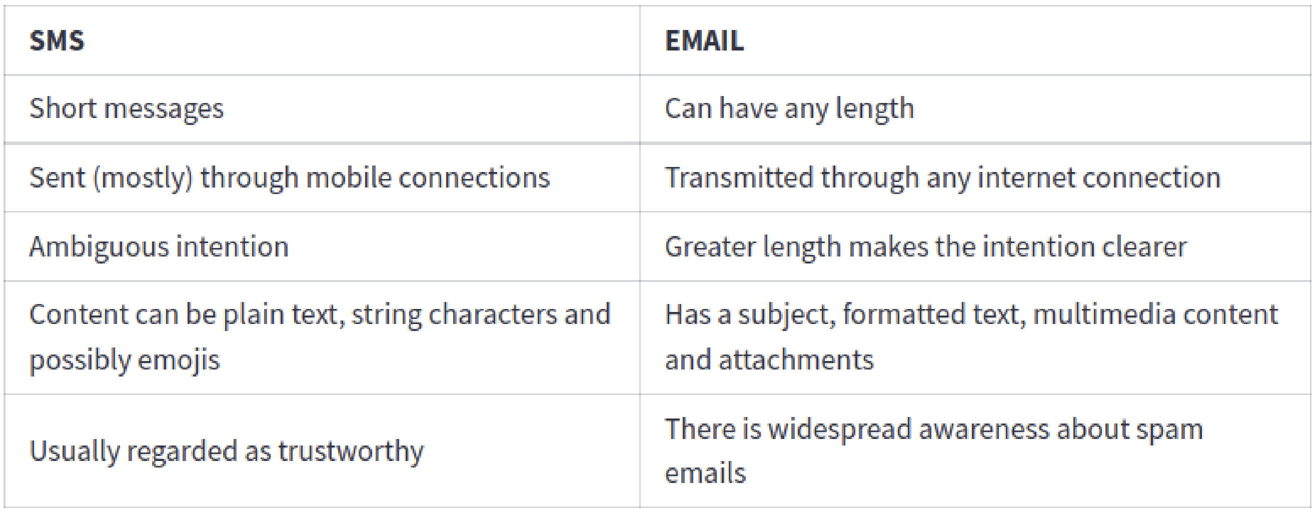
2018+ AI SPAM. Systems based on AI can manipulate reality, producing indistinguishable alternatives. AIs can also be target of manipulation and spam to elicit behaviors of the AI system or of its users.

Figure Timeline of the major milestones in the history of spam, from its inception to modern days.

Spam is unsolicited and unwanted messages sent electronically and whose content may be malicious. Email spam is sent/received over the Internet while SMS spam is typically transmitted over a mobile network. We'll refer to user that sent spam as 'spammers.

SMS messages are usually very cheap (if not free) for the user to send, making it appealing for unrightful exploitation. This is Furth aggravated by the fact that SMS is usually regarded by the user as a safer, more trustworthy form of communication than other sources, e. g., emails.

1.2 Difference of SMS Spam vs Email Spam



SMS	EMAIL
Short messages	Can have any length
Sent (mostly) through mobile connections	Transmitted through any internet connection
Ambiguous intention	Greater length makes the intention clearer
Content can be plain text, string characters and possibly emojis	Has a subject, formatted text, multimedia content and attachments
Usually regarded as trustworthy	There is widespread awareness about spam emails

Spammers' behavior: - *Spammers attempt to test the operator's anti-spam infrastructure by sending varying quantities of spam to determine if volume barriers are present. Is is very common for them to use multiple numbers to send messages, which rules out number blocking as an strategy to prevent spam. This situation requires some kind of content-based filtering that relies not only on volume but also on the content of the SMS itself.* ***State of spam filtering within the service: -*** *At the start of the project, actions against spammers consisted only in banning users that exceeded fixed daily and hourly thresholds with respect the number of sent SMS. There was at the time no content-based filtering nor any consideration based on user 0*

1.3 EXAMPLES OF SMS/EMAIL SPAM

Examples *These are examples based on real messages (not all were spam):*

- Your package is awaiting delivery 📥 bit.ly/xxxxxxx

How fast are your fingers? Test Now! -> https://play.google.com/store/apps/details?idxxxxxxx)

- Good evening sir, happy resumption. I had called your number today but you did not pick it. Please forward watchword money to my account.

- I use PayTree sent you up to € 25 Sign up with my link to claim, then get €300, 000 give away fund: https://palmpay.site/QXAfLgKsPhKA

- You have received USD150 from John Carpenter (+1)11111111 in your Mobile Money account on 2021–04–22 Message from sender: Transaction ID: 2901380912. For a successful cash out Ecobank will contact you for more inquiries.

- Follow this link to join my WhatsApp group: http://unnoficcial_whatsapp.com/download

- Please check and send to me, my result 100 level third year . Name: John-Paul Gillian Jambaya Email: johnpaulGillianJ@gmail.com Jamb reg no. 49851407SA

1.4 Explore and Analyze the Data

Let's take a look at the email message content and have a basic understanding of the data

Ham

This looks like a normal email reply to another person, which is not difficult to classified as a ham:

This is a bit of a messy solution but might be useful -

If you have an internal zip drive (not sure about external) and
you bios supports using a zip as floppy drive, you could
use a bootable zip disk with all the relevant dos utils.

Hard Ham (Ham email that is trickier)

Hard Ham is indeed more difficult to differentiate from the spam data, as they contain some key words such as limited time order, Special "Back to School" Offer, this make it very suspicious !

Hello Friends!

We hope you had a pleasant week. Last weeks trivia questions was:

What do these 3 films have in common: One Crazy Summer, Whispers in the =
Dark, Moby Dick?=20

Answer: Nantucket Island

Congratulations to our Winners:

Caitlin O. of New Bedford, Massachusetts

Brigid M. of Marblehead, Massachusetts

Special "Back to School" Offer!

For a limited time order our "Back to School" Snack Basket and receive = 20% Off & FREE SHIPPING!

Spam

One of the spam training data does look like one of those spam advertisement email in our junk folder:

IMPORTANT INFORMATION:

The new domain names are finally available to the general public at discount prices. Now you can register one of the exciting new .BIZ or .INFO domain names, as well as the original .COM and .NET names for just $14.95. These brand new domain extensions were recently approved by ICANN and have the same rights as the original .COM and .NET domain names. The biggest benefit is of-course that the .BIZ and .INFO domain names are currently more available. i.e. it will be much easier to register an attractive and easy-to-remember domain name for the same price. Visit: http://www.affordable-domains.com today for more info.

CHAPTER 2: -INTRODUCTION TO LEARNING ALGORITHM

2.1Jupyter

Jupyter, previously known as I Python 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 GitHub for free. For each section of this tutorial, you can download a Jupyter 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
```

2.2 Machine Learning

Machine learning (ML) is a discipline of artificial intelligence (AI) that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention.

Machine learning methods enable computers to operate autonomously without explicit programming. ML applications are fed with new data, and they can independently learn, grow, develop, and adapt.

Machine learning derives insightful information from large volumes of data by leveraging algorithms to identify patterns and learn in an iterative process. ML algorithms use computation methods to learn *directly from data instead of relying on any predetermined equation that may serve as a model.*

The performance of ML algorithms adaptively improves with an increase in the number of available samples during the 'learning' processes. For example, deep learning is a sub-domain of machine learning that trains computers to imitate natural human traits like learning from examples. It offers better performance parameters than conventional ML algorithms.

While machine learning is not a new concept – dating back to World War II when the Enigma Machine was used – the ability to apply complex mathematical calculations automatically to growing volumes and varieties of available data is a relatively recent development.

Today, with the rise of big data, IoT, and ubiquitous computing, machine learning has become essential for solving problems across numerous areas, such as

- *Computational finance (credit scoring, algorithmic trading)*
- *Computer vision (facial recognition, motion tracking, object detection)*
- *Computational biology (DNA sequencing, brain tumor detection, drug discovery)*
- *Automotive, aerospace, and manufacturing (predictive maintenance)*
- *Natural language processing (voice recognition)*

2.3 SciKit-Learn

Scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface. It is an open-source machine-learning library that provides a plethora of tools for various machine-

learning tasks such as Classification, Regression, Clustering, and many more.

Installation of Scikit- learn

The latest version of Scikit-learn is 1.1 and it requires Python 3.8 or newer.

Scikit-learn requires:

- *NumPy*
- *SciPy as its dependencies.*

Before installing scikit-learn, ensure that you have NumPy and SciPy installed. Once you have a working installation of NumPy and SciPy, the easiest way to install scikit-learn is using pip:

```
! pip install -U scikit-learn
```

- *Clustering or cluster analysis is a machine learning technique, which groups the unlabeled dataset. It can be defined as* **"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."**

It does it by finding some similar patterns in the unlabeled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.

After applying this clustering technique, each cluster or group is provided with a cluster-ID. ML system can use this id to simplify the processing of large and complex datasets.

The clustering technique is commonly used for ***statistical data analysis.***

Note: Clustering is somewhere similar to the classification algorithm, but the difference is the type of dataset that we are using. In classification, we work with the labeled data set, whereas in clustering, we work with the unlabeled dataset.

Example:

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

- *Market Segmentation*
- *Statistical data analysis*
- *Social network analysis*
- *Image segmentation*
- *Anomaly detection, etc.*

2.4 Classification: - *The world has become data-driven, and artificial intelligence and machine learning are using this data to understand society, predict business outcomes, and drive decision-making and growth. Classification in machine learning is one of the most common and widely used supervised machine learning processes. It helps in categorizing data into different classes and has a broad array of applications, such as email spam detection, medical diagnostic test, fraud detection, image classification, and speech recognition among others. This guide is a deep dive into classification in machine learning, types of classification tasks, classification algorithms, and learners in classification problems. But before we dig deep, let's understand some related concepts.*

Classification : A classification problem is used to identify specific categories of new observations based on one or more independent variables. Also, in this article, we will focus on classification.

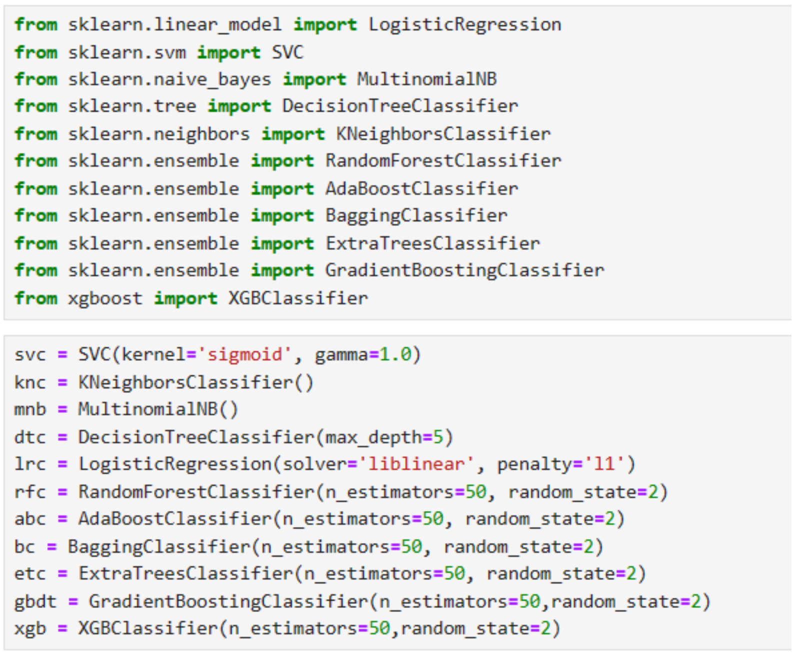
Overview of Classification

Classification is a supervised machine learning process of categorizing a given set of input data into classes based on one or more variables. Additionally, a classification problem can be performed on structured and unstructured data to accurately predict whether or not the data will fall into predetermined categories.

Classification in machine learning can require two or more categories of a given data set. Therefore, it generates a probability score to assign the data into a specific category, such as spam or not spam, yes or no, disease or no disease, red or green, male or female, etc.

Some Applications of Machine Learning Classification Problems

- *Image classification*
- *Fraud detection*
- *Document classification*
- *Spam filtering*
- *Facial recognition*
- *Voice recognition*
- *Medical diagnostic test*
- *Customer behavior prediction*
- *Product categorization*
- *Malware classification*



```python
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import GradientBoostingClassifier
from xgboost import XGBClassifier
```

```python
svc = SVC(kernel='sigmoid', gamma=1.0)
knc = KNeighborsClassifier()
mnb = MultinomialNB()
dtc = DecisionTreeClassifier(max_depth=5)
lrc = LogisticRegression(solver='liblinear', penalty='l1')
rfc = RandomForestClassifier(n_estimators=50, random_state=2)
abc = AdaBoostClassifier(n_estimators=50, random_state=2)
bc = BaggingClassifier(n_estimators=50, random_state=2)
etc = ExtraTreesClassifier(n_estimators=50, random_state=2)
gbdt = GradientBoostingClassifier(n_estimators=50,random_state=2)
xgb = XGBClassifier(n_estimators=50,random_state=2)
```

Fig. 2 *Ensemble Model*

- *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 F1 score (F1 = 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*

2.5 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 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 Principal 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*

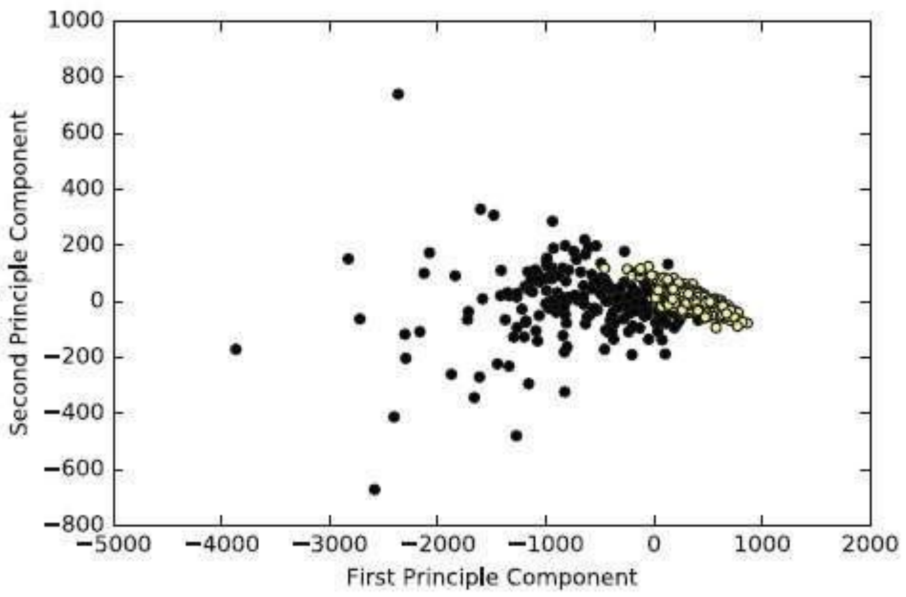


Fig. 2.1 PCA

Chapter 3: -Project Model building (ensemble modeling)

1. ***Logistic Regression*** *:- It is used for binary classification where we use sigmoid function, that takes input as independent variables and produces a probability value between 0 and 1*
   .
   For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 otherwise it belongs to Class 0. It's referred to as regression because it is the extension of linear regression but is mainly used for classification problems.

   Key Points:
   - *Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value.*
   - *It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.*
   - *In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).*

2. ***MuiltinomialNB: -*** *Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other. To start with, let us consider a dataset. One of the most simple and effective classification algorithms, the Naïve Bayes classifier aids in the rapid development of machine learning models with rapid prediction capabilities. Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. In text classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, rating classification etc. The advantage of using naïve Bayes is its speed. It is fast and*

the probability of an instance belongs to a class with a given set of feature value. It is a probabilistic

classifier. It is because it assumes that one feature in the model is independent of existence of another feature. In other words, each feature contributes to the predictions with no relation between each other. In real world, this condition satisfies rarely. It uses Bayes theorem in the algorithm for training and prediction

3. ***DECISION TREE*** *is a flowchart-like tree structure where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile supervised machine-learning algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in machine learning.*

   Decision Tree Terminologies

   Some of the common Terminologies used in Decision Trees are as follows:

   - ***Root Node:*** *It is the topmost node in the tree, which represents the complete dataset. It is the starting point of the decision-making process.*
   - *Decision/Internal Node: A node that symbolizes a choice regarding an input feature. Branching off of internal nodes connects them to leaf nodes or other internal nodes.*
   - ***Leaf/Terminal Node:*** *A node without any child nodes that indicates a class label or a numerical value.*
   - ***Splitting:*** *The process of splitting a node into two or more sub-nodes using a split criterion and a selected feature.*
   - ***Branch/Sub-Tree:*** *A subsection of the decision tree starts at an internal node and ends at the leaf nodes.*
   - ***Parent Node:*** *The node that divides into one or more child nodes.*
   - ***Child Node:*** *The nodes that emerge when a parent node is split.* ***Impurity****: A measurement of the target variable's homogeneity in a subset of data. It refers to the degree of randomness or uncertainty in a set of examples. The* ***Gini index*** *and* ***entropy*** *are two*

- *commonly used impurity measurements in decision trees for classifications task*
- ***Variance**: Variance measures how much the predicted and the target variables vary in different samples of a dataset. It is used for regression problems in decision trees. **Mean squared error,***

  ***Mean Absolute Error, friedman_mse, or Half Poisson deviance** are used to measure the variance for the regression tasks in the decision tree.*

- ***Information Gain:** Information gain is a measure of the reduction in impurity achieved by splitting a dataset on a particular feature in a decision tree. The splitting criterion is determined by the feature that offers the greatest information gain, It is used to determine the most informative feature to split on at each node of the tree, with the goal of creating pure subsets*
- ***Pruning**: The process of removing branches from the tree that do not provide any additional information or lead to overfitting*

4. ***ADABOOST** – AdaBoost is a boosting algorithm that also works on the principle of the stagewise addition method where multiple weak learners are used for getting strong learners. The value of the alpha parameter, in this case, will be indirectly proportional to the error of the weak learner, Unlike Gradient Boosting in XGBoost, the alpha parameter calculated is related to the errors of the weak learner, here the value of the alpha parameter will be indirectly proportional to the error of the weak learner.*

5. ***BAGGING** (or Bootstrap aggregating) is a type of ensemble learning in which multiple base models are trained independently and in parallel on different subsets of the training data. Each subset is generated using bootstrap sampling, in which data points are picked at random with replacement. In the case of the bagging classifier, the final prediction is made by aggregating the*
6. Team ID: 289947

models of regression, the final prediction is made by averaging the predictions of the all-base model, and that is known as bagging regression.

7. ***Extremely Randomized Trees Classifier (Extra Trees Classifier)*** *is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output its classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees. To perform feature selection using the above forest structure, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in the decision of feature of split (Gini Index if the Gini Index is used in the construction of the forest) is computed. This value is called the Gini Importance of the feature. To perform feature selection, each feature is ordered in descending order according to the Gini Importance of each feature and the user selects the top k features according to his/her choice.*

8. **GRADIENT *BOOSTING:*** - *It* is a powerful boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current and then trains a new weak model to minimize this gradient. The predictions of the new model are then added to the ensemble, and the process is repeated until a stopping criterion is met. *In contrast to AdaBoost, the weights of the training instances*

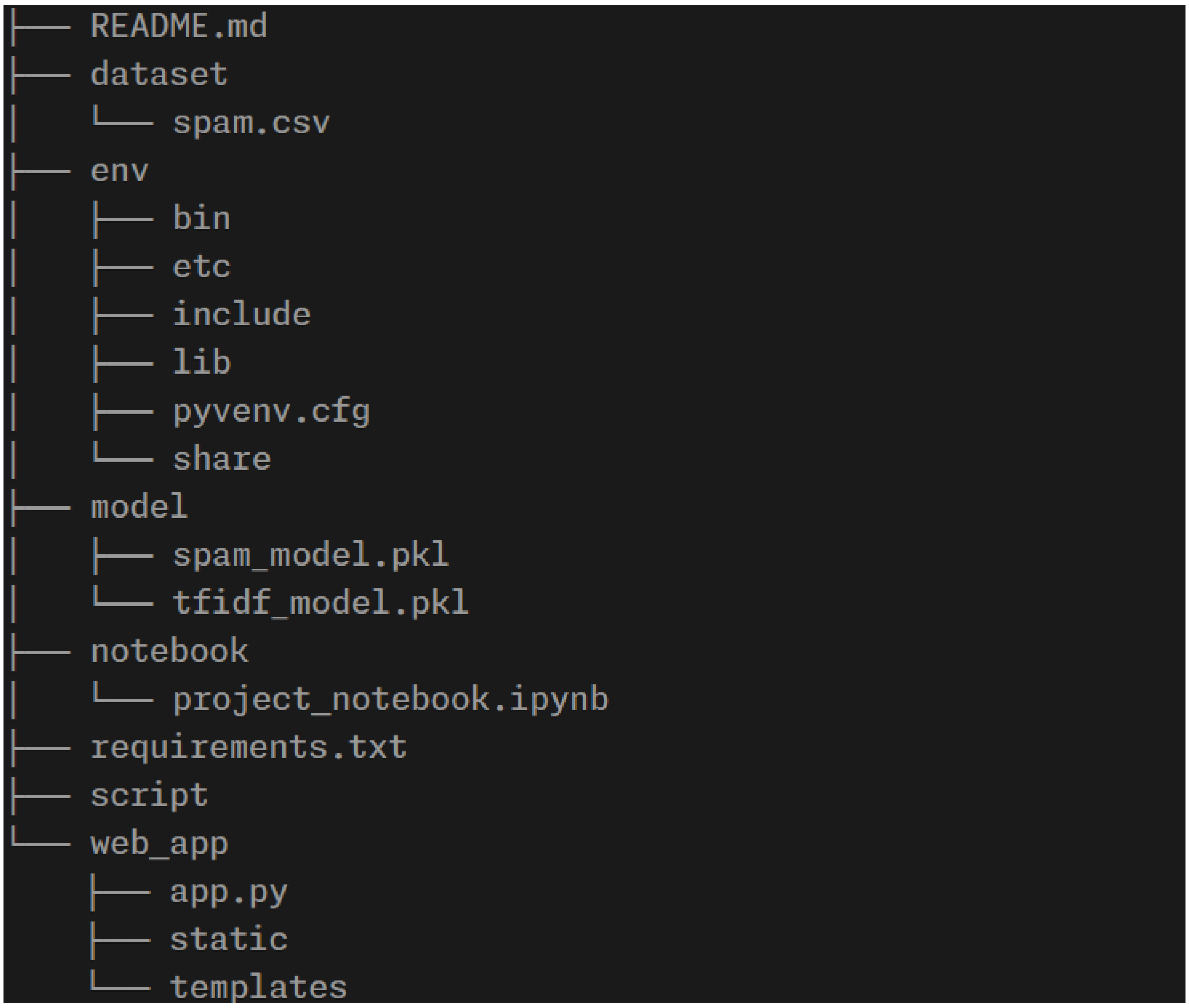
are not tweaked, instead, each predictor is trained using the residual errors of the predecessor as labels. There is a technique called the ***Gradient Boosted Trees*** *whose base learner is CART (Classification and Regression Trees). The below diagram explains how gradient-boosted trees are trained for regression problems.*

9. ***XGBOOST,*** *or Extreme Gradient Boosting, is a state-of-the-art machine learning algorithm renowned for its exceptional predictive performance. It is the gold standard in ensemble learning, especially when it comes to gradient-boosting algorithms. It develops a series of weak learners one after the other to produce a reliable and accurate predictive model. Fundamentally, XGBoost builds a strong predictive model by aggregating the predictions of several weak learners, usually decision trees. It uses a boosting technique to create an extremely accurate ensemble model by having each weak learner after it correct the mistakes of its predecessors. The optimization method (gradient) minimizes a cost function by repeatedly changing the model's parameters in response to the gradients of the errors. The algorithm also presents the idea of "gradient boosting with decision trees," in which the objective function is reduced by calculating the importance of each decision tree that is added to the ensemble in turn. By adding a regularization term and utilizing a more advanced optimization algorithm, XGBoost goes one step further and improves accuracy and efficiency. It has gained popularity and widespread usage because it can handle large datasets in a variety of machine-learning tasks, including regression and classification.*

CHAPTER 4: PROJECT DESCRIPTION

4.1 File Structure

An overview of the file directory for this project is shown below:



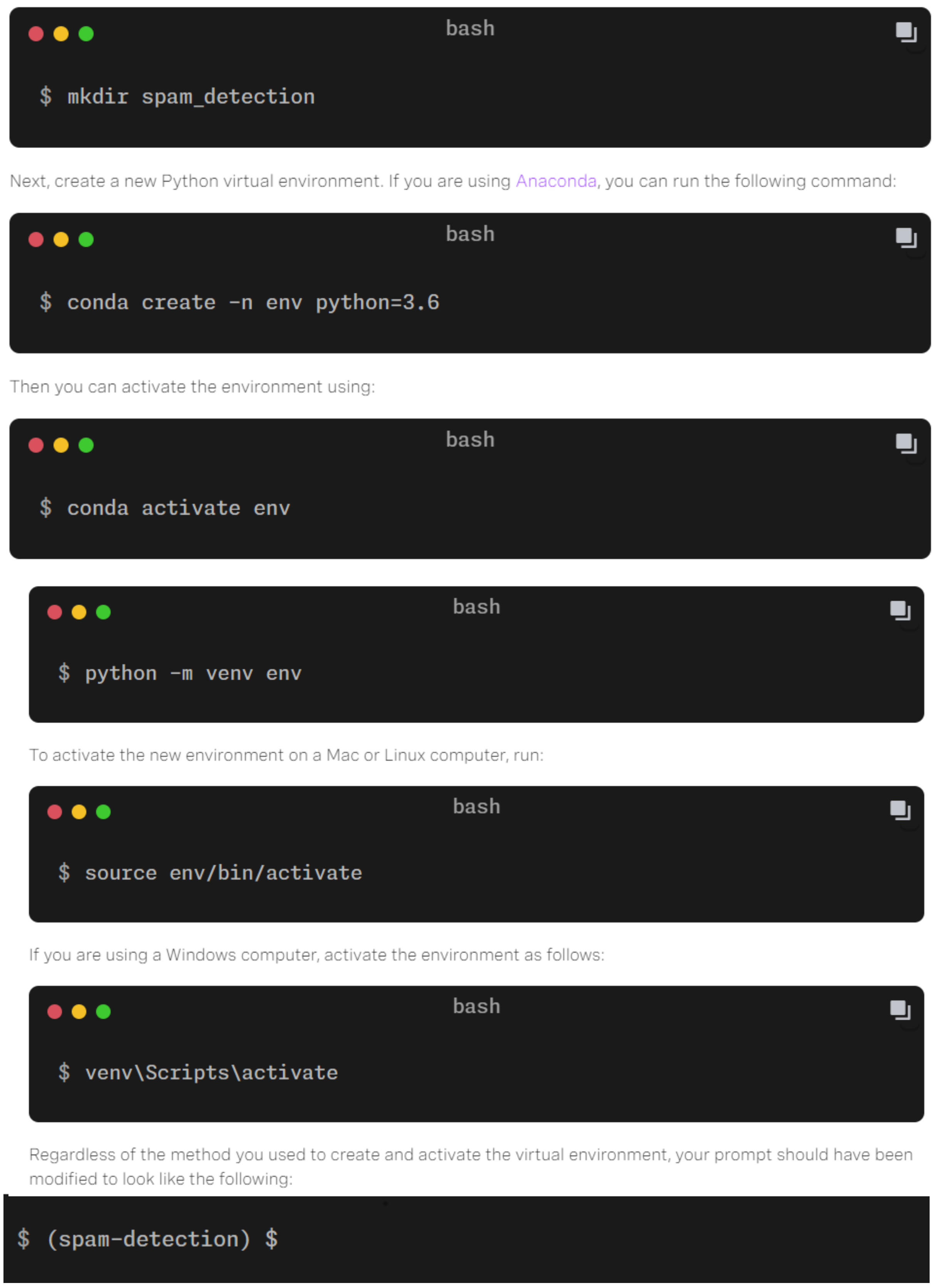
```
├── README.md
├── dataset
│   └── spam.csv
├── env
│   ├── bin
│   ├── etc
│   ├── include
│   ├── lib
│   ├── pyvenv.cfg
│   └── share
├── model
│   ├── spam_model.pkl
│   └── tfidf_model.pkl
├── notebook
│   └── project_notebook.ipynb
├── requirements.txt
├── script
└── web_app
    ├── app.py
    ├── static
    └── templates
```

FIG. 4.1 Files Structure

4.2 Set Up a Python Virtual Environment

We need to create an isolated environment for various Python dependencies unique to this project.

First, create a new development folder. In your terminal, run:



```
$ mkdir spam_detection
```

Next, create a new Python virtual environment. If you are using Anaconda, you can run the following command:

```
$ conda create -n env python=3.6
```

Then you can activate the environment using:

```
$ conda activate env
```

```
$ python -m venv env
```

To activate the new environment on a Mac or Linux computer, run:

```
$ source env/bin/activate
```

If you are using a Windows computer, activate the environment as follows:

```
$ venv\Scripts\activate
```

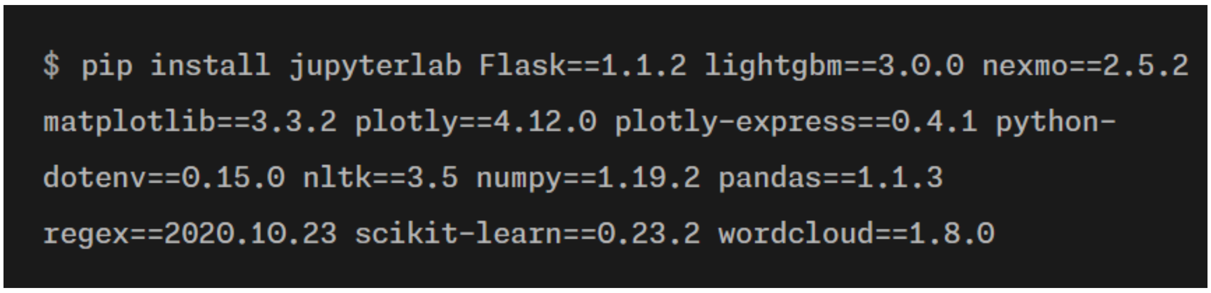
Regardless of the method you used to create and activate the virtual environment, your prompt should have been modified to look like the following:

```
$ (spam-detection) $
```

Fig. 4.2

4.4 Install Required Packages

Next, you'll install all the packages needed for this tutorial. In your new environment, install the following packages (which includes libraries and dependencies):



```
$ pip install jupyterlab Flask==1.1.2 lightgbm==3.0.0 nexmo==2.5.2
matplotlib==3.3.2 plotly==4.12.0 plotly-express==0.4.1 python-
dotenv==0.15.0 nltk==3.5 numpy==1.19.2 pandas==1.1.3
regex==2020.10.23 scikit-learn==0.23.2 wordcloud==1.8.0
```

Fig.4.4

Here are some details about these packages:

- *jupyterlab is for model building and data exploration.*
- *flask is for creating the application server and pages.*
- *lightgbm is the machine learning algorithm for building our model*
- *nexmo is a Python library for interacting with your Vonage account*
- *matplotlib, plotly, plotly-express are for data visualization*
- *python-dotenv is a package for managing environment variables such as API keys and other configuration values.*
- *nltk is for natural language operations*
- *numpy is for arrays computation*
- *pandas is for manipulating and wrangling structured data.*
- *regex is for regular expression operations*
- *scikit-learn is a machine learning toolkit*
- *wordcloud is used to create word cloud images from text*

After installation, start your Jupyter lab by running:

- *The spam increased in these days due more mobile devices deployed in environment for e-mail and message communication.*

```
$ jupyter lab
```

This opens the popular Jupyter lab interface in your web browser, where you are going to carry out some interactive data exploration and model building.

Jupyter lab interface is shown here Jupyterlab

4.5 Build and Train the SMS Detection Model

Now that your environment is ready, you’re going to download the SMS training data and build a simple machine learning model to classify the SMS messages. The spam dataset for this project can be downloaded here. The datasets contain 5574 messages with respective labels of spam and ham (legitimate). More about the dataset can be found here. With this data, we will train a machine learning model that can correctly classify SMS as ham or spam. These procedures will be carried out in a Jupyter notebook, which from our file directory is named 'project_notebook'

4.5.1 Exploratory Data Analysis (EDA)

Here, we will apply a variety of techniques to analyze the data and get a better understanding of it.

Import Libraries and Data

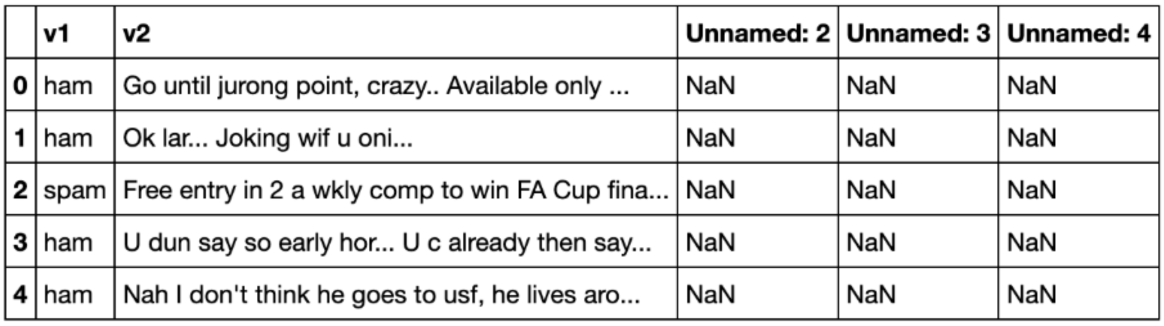
The necessary libraries for this project can be imported into `project_notebook.ipynb` as follows:

```
import pandas as pd
import numpy as np
import matplotlib. pyplot as plt
import plotly_express as px
import wordcloud
import nltk
import warnings
warnings.filterwarnings('ignore')
```

The spam dataset located in the dataset directory named spam.csv can be imported as follows:

```
df = pd.read_csv("../dataset/spam.csv", encoding='latin-1')
```

```
df.head()
```



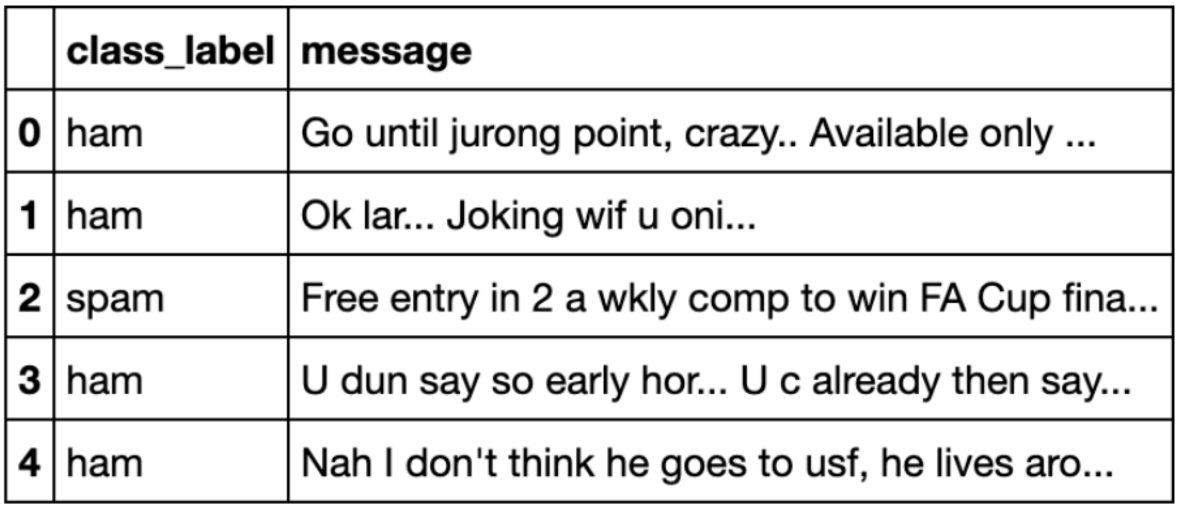
	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

Dataset overview

The dataset contains 5 columns. Column v1 is the dataset label ("ham" or "spam") and column v2 contains the text of the SMS message. Columns "Unnamed: 2", "Unnamed: 3", and "Unnamed: 4" contain "NaN" (not a number) signifying missing values. They are not needed, so they can be dropped as they are not going to be useful in building the model. The following code snippet will drop and rename the columns to improve understandability of the dataset:

```
df.drop(columns=['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'],
inplace=True)
df.rename(columns = {'v1':'class_label','v2':'message'},inplace=True)
df.head()
```

Team ID: 289947



	class_label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

4.6 Introduction

- *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.*

1. *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.*

KNN and Random Forest.

4.7 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), Makowski 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 non-generalizing 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 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.

CHAPTER 5: - LITERATURE SURVEY

1. 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 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 Short-Term 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%.

2. *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%*

3. 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 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.

4. *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

Team ID: 420772

5. TITLE: SMS Spam Detection using Machine Learning Approach

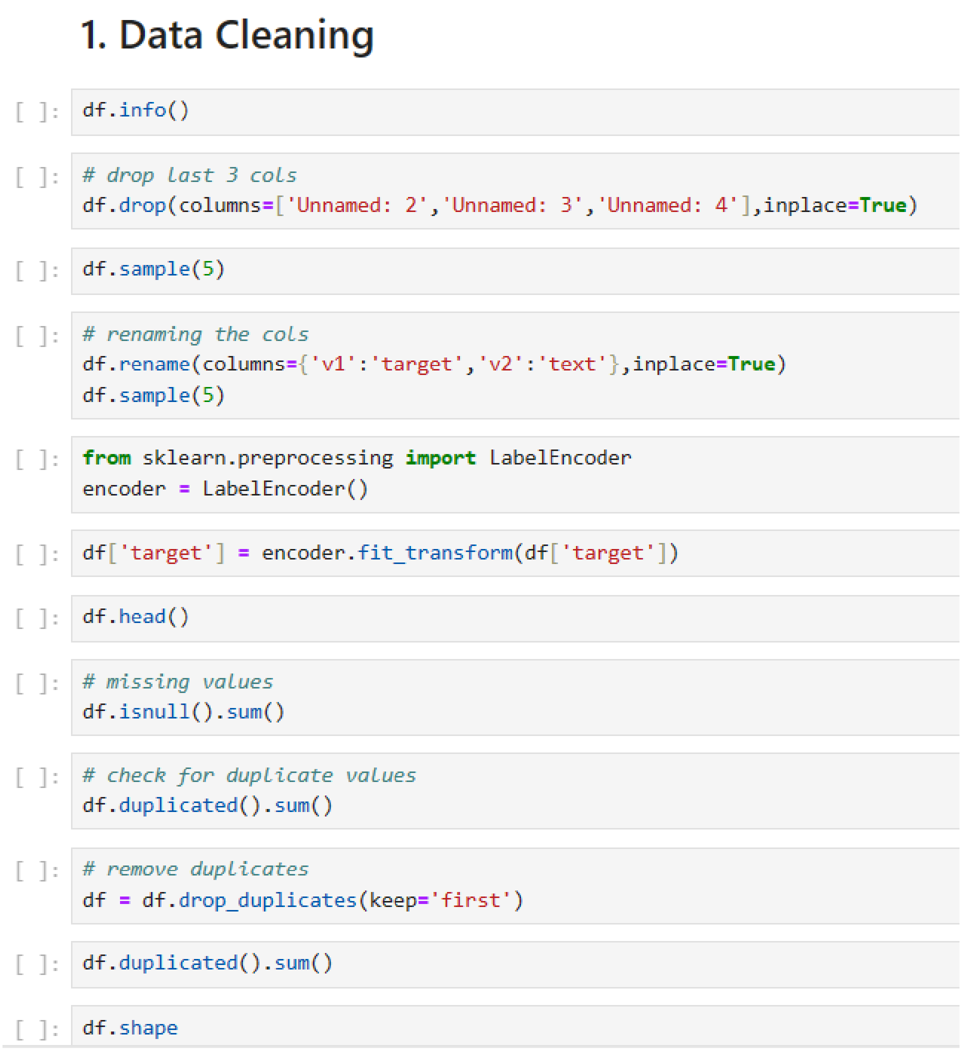
Author: *Houshmand Shirani-Mehr, hshirani@stanford.edu*

YEAR*: - 2019*

Abstracts:

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.

Chapter 6 Model Building



1. Data Cleaning

```
[ ]: df.info()

[ ]: # drop last 3 cols
     df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)

[ ]: df.sample(5)

[ ]: # renaming the cols
     df.rename(columns={'v1':'target','v2':'text'},inplace=True)
     df.sample(5)

[ ]: from sklearn.preprocessing import LabelEncoder
     encoder = LabelEncoder()

[ ]: df['target'] = encoder.fit_transform(df['target'])

[ ]: df.head()

[ ]: # missing values
     df.isnull().sum()

[ ]: # check for duplicate values
     df.duplicated().sum()

[ ]: # remove duplicates
     df = df.drop_duplicates(keep='first')

[ ]: df.duplicated().sum()

[ ]: df.shape
```

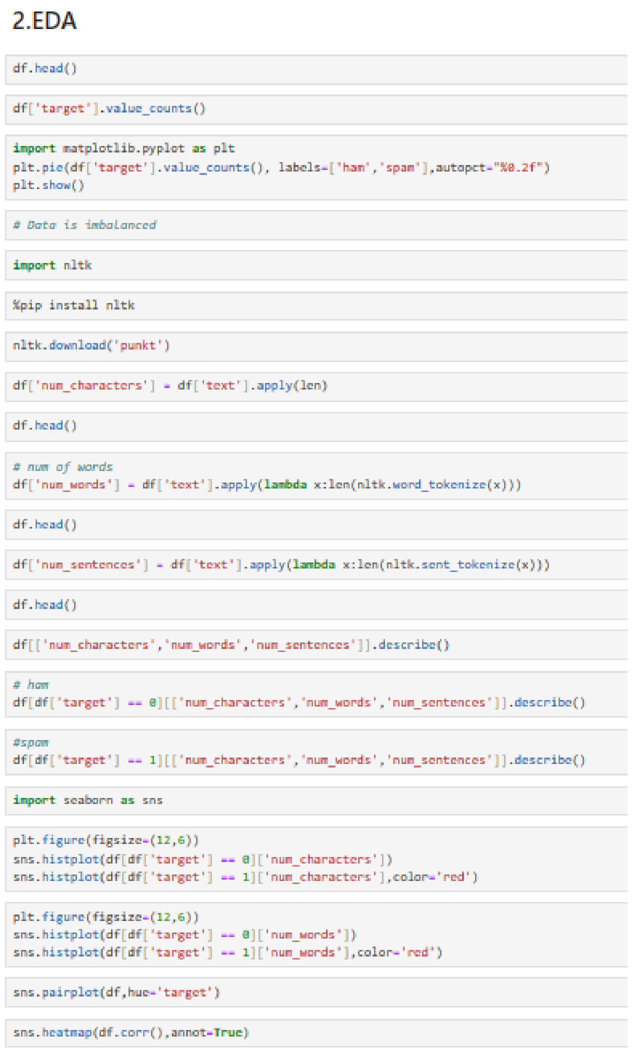
Data scientists spend a large amount of their time cleaning datasets and getting them down to a form with which they can work. In fact, a lot of data scientists argue that the initial steps of obtaining and cleaning data constitute 80% of the job.

Therefore, if you are just stepping into this field or planning to step into this field, it is important to be able to deal with messy data, whether that *means missing values, inconsistent formatting, malformed records, or nonsensical outliers.In this tutorial, we'll leverage Python's pandas and NumPy libraries to clean data.*

We'll cover the following:

- *Dropping unnecessary columns in a DataFrame*
- *Changing the index of a DataFrame*
- *Using .str () methods to clean columns*
- *Using the DataFrame.applymap() function to clean the entire dataset, element-wise*
- *Renaming columns to a more recognizable set of labels*
- *Skipping unnecessary rows in a CSV file*

2. EDA (Exploratory Data Analysis)



2.EDA

```
df.head()
```

```
df['target'].value_counts()
```

```
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
```

```
# Data is imbalanced
```

```
import nltk
```

```
%pip install nltk
```

```
nltk.download('punkt')
```

```
df['num_characters'] = df['text'].apply(len)
```

```
df.head()
```

```
# num of words
df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
```

```
df.head()
```

```
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
```

```
df.head()
```

```
df[['num_characters','num_words','num_sentences']].describe()
```

```
# ham
df[df['target'] == 0][['num_characters','num_words','num_sentences']].describe()
```

```
#spam
df[df['target'] == 1][['num_characters','num_words','num_sentences']].describe()
```

```
import seaborn as sns
```

```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
```

```
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_words'])
sns.histplot(df[df['target'] == 1]['num_words'],color='red')
```

```
sns.pairplot(df,hue='target')
```

```
sns.heatmap(df.corr(),annot=True)
```

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables. EDA is normally carried out as a preliminary step before undertaking extra formal statistical analyses or modeling.

The Foremost Goals of EDA

1. Data Cleaning: EDA involves examining the information for errors, lacking values, and inconsistencies. It includes techniques including records imputation, managing missing statistics, and figuring out and getting rid of outliers.

2. Descriptive Statistics: EDA utilizes precise records to recognize the important tendency, variability, and distribution of variables. Measures like suggest, median, mode, preferred deviation, range, and percentiles are usually used.

3. Data Visualization: EDA employs visual techniques to represent the statistics graphically. Visualizations consisting of histograms, box plots, scatter plots, line plots, heatmaps, and bar charts assist in identifying styles, trends, and relationships within the facts.

4. Feature Engineering: EDA allows for the exploration of various variables and their adjustments to create new functions or derive meaningful insights. Feature engineering can contain scaling, normalization, binning, encoding express variables, and creating interplay or derived variables.

5. Correlation and Relationships: EDA allows discover relationships and dependencies between variables. Techniques such as correlation analysis, scatter plots, and pass-tabulations offer insights into the power and direction of relationships between variables.

6. Data Segmentation: EDA can contain dividing the information into significant segments based totally on sure standards or traits. This segmentation allows advantage insights into unique subgroups inside the information and might cause extra focused analysis.

7. Hypothesis Generation: EDA aids in generating hypotheses or studies questions based totally on the preliminary exploration of the data. It facilitates form the inspiration for in addition evaluation and model building.

Team ID: 289947

8. Data Quality Assessment: EDA permits for assessing the nice and reliability of the information. It involves checking for records integrity, consistency, and accuracy to make certain the information is suitable for analysis.

Types of EDA

Depending on the number of columns we are analyzing we can divide EDA into two types.

EDA, or Exploratory Data Analysis, refers back to the method of analyzing and analyzing information units to uncover styles, pick out relationships, and gain insights. There are various sorts of EDA strategies that can be hired relying on the nature of the records and the desires of the evaluation. Here are some not unusual kinds of EDA:

1. Univariate Analysis: This sort of evaluation makes a specialty of analyzing character variables inside the records set. It involves summarizing and visualizing a unmarried variable at a time to understand its distribution, relevant tendency, unfold, and different applicable records. Techniques like histograms, field plots, bar charts, and precis information are generally used in univariate analysis.

2. Bivariate Analysis: Bivariate evaluation involves exploring the connection between variables. It enables find associations, correlations, and dependencies between pairs of variables. Scatter plots, line plots, correlation matrices, and move-tabulation are generally used strategies in bivariate analysis.

3. Multivariate Analysis: Multivariate analysis extends bivariate evaluation to encompass greater than variables. It ambitions to apprehend the complex interactions and dependencies among more than one variable in a records set. Techniques inclusive of heatmaps, parallel coordinates, aspect analysis, and primary component analysis (PCA) are used for multivariate analysis.

4. Time Series Analysis: This type of analysis is mainly applied to statistics sets that have a temporal component. Time collection evaluation entails inspecting and modeling styles, traits, and seasonality inside the statistics through the years. Techniques like line plots, autocorrelation analysis, transferring averages, and ARIMA (Autoregressive Integrated Moving Average) fashions are generally utilized in time series analysis.

5. Missing Data Analysis: Missing information is a not unusual issue in datasets, and it may impact the reliability and validity of the evaluation. Missing statistics analysis includes figuring out missing values, know-how the patterns of missingness, and using suitable techniques to deal with missing data. Techniques along with lacking facts styles, imputation strategies, and sensitivity evaluation are employed in lacking facts evaluation.

6. Outlier Analysis: Outliers are statistics factors that drastically deviate from the general sample of the facts. Outlier analysis includes identifying and knowledge the presence of outliers, their capability reasons, and their impact at the analysis. Techniques along with box plots, scatter plots, z-rankings, and clustering algorithms are used for outlier evaluation.

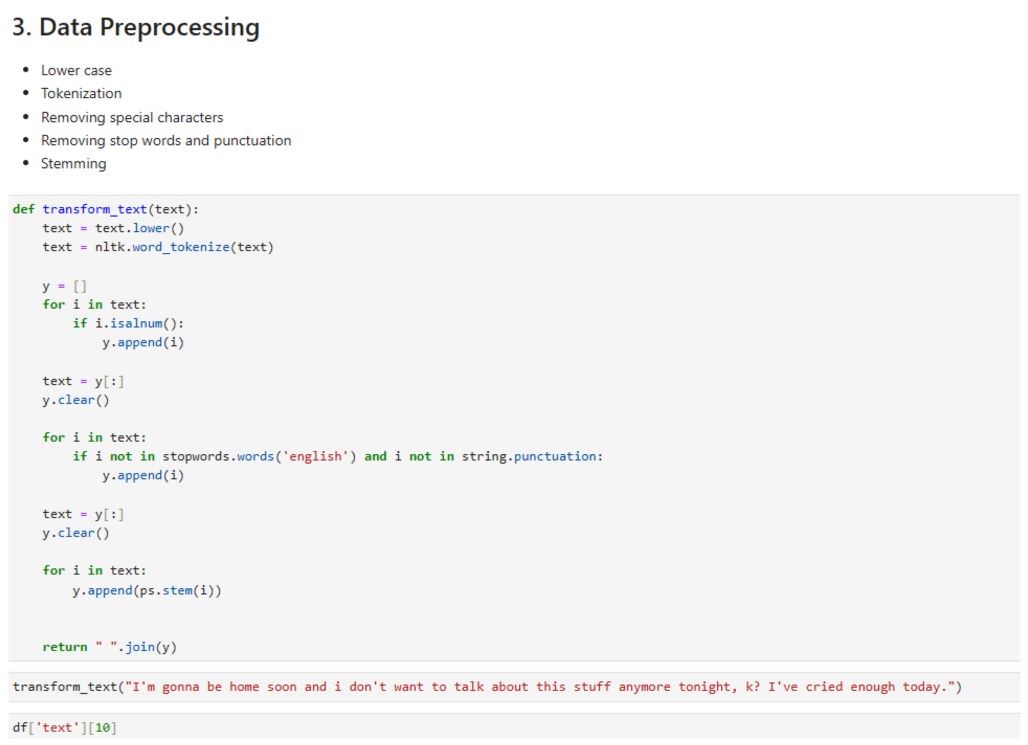
7. Data Visualization: Data visualization is a critical factor of EDA that entails creating visible representations of the statistics to facilitate understanding and exploration. Various visualization techniques, inclusive of bar charts, histograms, scatter plots, line plots, heatmaps, and interactive dashboards, are used to represent exclusive kinds of statistics.

These are just a few examples of the types of EDA techniques that can be employed at some stage in information evaluation. The choice of strategies relies upon on the information traits, research questions, and the insights sought from the analysis.

3. Data Preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So for this, we use data preprocessing task.



3. Data Preprocessing

- Lower case
- Tokenization
- Removing special characters
- Removing stop words and punctuation
- Stemming

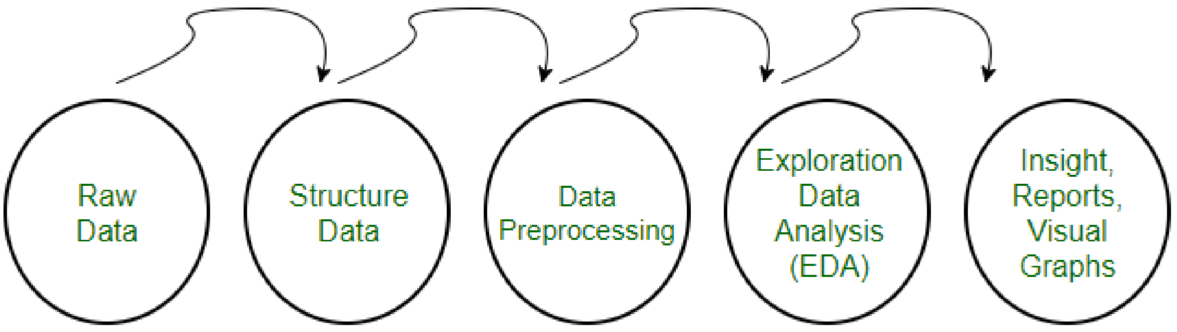
```
def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))


    return " ".join(y)
```

```
transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.")
```

```
df['text'][10]
```

Data Preprocessing

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis.


Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

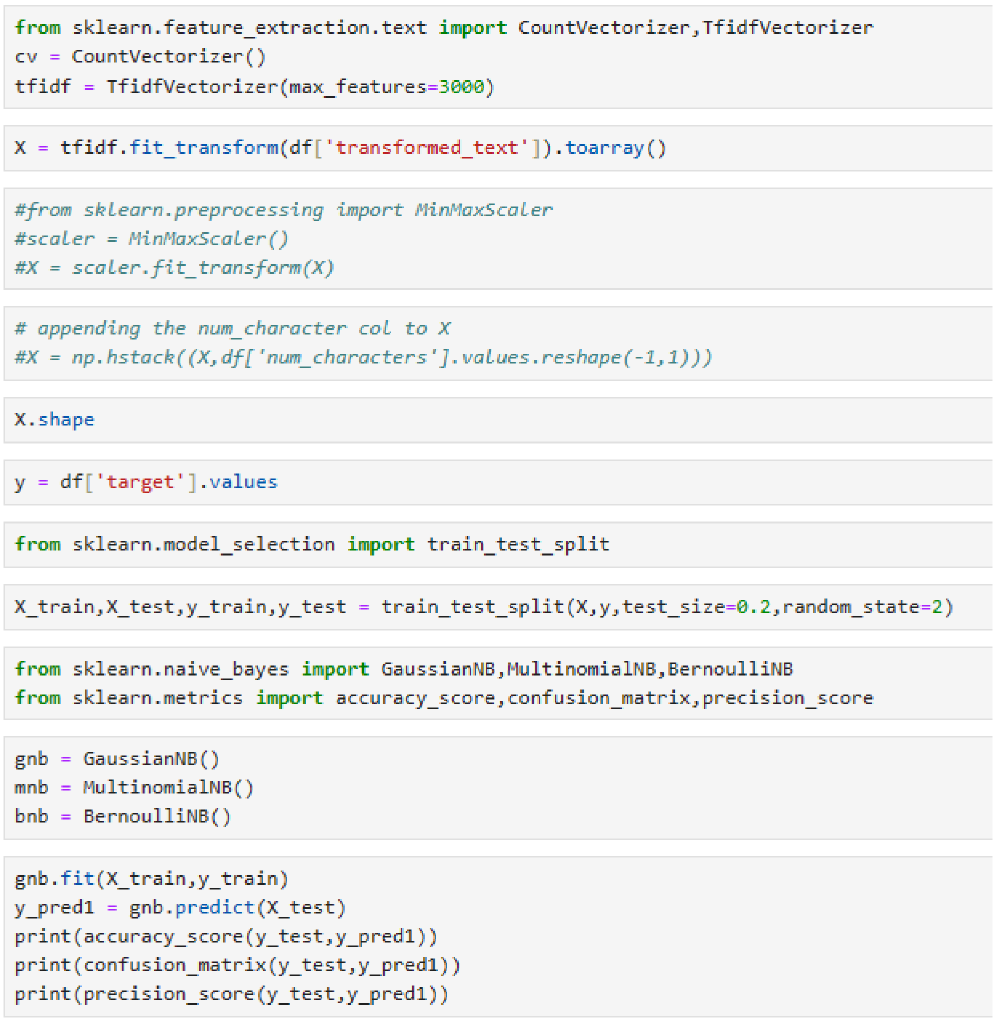
Data Preprocessing

Need of Data Preprocessing

- For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values, therefore to execute random forest algorithm null values have to be managed from the original raw data set.

- Another aspect is that the data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithm are executed in one data set, and best out of them is chosen.

4. *Model Building*



```
from sklearn.feature_extraction.text import CountVectorizer,TfidfVectorizer
cv = CountVectorizer()
tfidf = TfidfVectorizer(max_features=3000)
```

```
X = tfidf.fit_transform(df['transformed_text']).toarray()
```

```
#from sklearn.preprocessing import MinMaxScaler
#scaler = MinMaxScaler()
#X = scaler.fit_transform(X)
```

```
# appending the num_character col to X
#X = np.hstack((X,df['num_characters'].values.reshape(-1,1)))
```

```
X.shape
```

```
y = df['target'].values
```

```
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

```
from sklearn.naive_bayes import GaussianNB,MultinomialNB,BernoulliNB
from sklearn.metrics import accuracy_score,confusion_matrix,precision_score
```

```
gnb = GaussianNB()
mnb = MultinomialNB()
bnb = BernoulliNB()
```

```
gnb.fit(X_train,y_train)
y_pred1 = gnb.predict(X_test)
print(accuracy_score(y_test,y_pred1))
print(confusion_matrix(y_test,y_pred1))
print(precision_score(y_test,y_pred1))
```

Certainly! Building a machine learning model involves several key steps. Let's walk through them:

1. ***Data Collection for Machine Learning**:*
   - *Start by gathering relevant data for your problem. This data will be used to train and evaluate your model. Ensure that the data is representative and covers a wide range of scenarios.*
   - *Collect data from various sources, such as databases, APIs, or user-generated content.*

2. ***Preprocessing and Preparing Your Data***:
   - *Clean and preprocess the data to make it suitable for training. This includes handling missing values, removing outliers, and normalizing features.*
   - *Split the data into training and testing sets. The training set is used to train the model, while the testing set evaluates its performance.*
3. ***Selecting the Right Machine Learning Model***:
   - *Choose an appropriate algorithm based on your problem type (e.g., regression, classification, clustering).*
   - *Consider factors like interpretability, scalability, and computational efficiency when selecting a model.*
4. ***Training Your Machine Learning Model***:
   - *Feed the training data into the chosen model.*
   - *The model learns from the data and adjusts its internal parameters to minimize the prediction error.*
5. ***Evaluating Model Performance***:
   - *Use the testing data to assess how well the model generalizes to unseen examples.*
   - *Common evaluation metrics include accuracy, precision, recall, F1-score, and area under the receiver operating characteristic curve (AUC-ROC).*
6. ***Tuning and Optimizing Your Model***:
   - *Fine-tune the model by adjusting hyperparameters (e.g., learning rate, regularization strength).*
   - *Use techniques like cross-validation to find the best hyperparameter values.*
7. ***Deploying the Model and Making Predictions***:
   - *Once satisfied with the model's performance, deploy it in a production environment.*
   - *Use the deployed model to make predictions on new, unseen data.*

5. Machine Learning Model Evaluation

Model evaluation is the process that uses some metrics which help us to analyze the performance of the model. As we all know that model development is a multi-step process and a check should be kept on how well the model generalizes future predictions. Therefore, evaluating a model plays a vital role so that we can judge the performance of our model. The evaluation also helps to analyze a model's key weaknesses. There are many metrics like Accuracy, Precision, Recall, F1 score, Area under Curve, Confusion Matrix, and Mean Square Error. Cross Validation is one technique that is followed during the training phase and it is a model evaluation technique as well.

Cross Validation and Holdout

Cross Validation is a method in which we do not use the whole dataset for training. In this technique, some part of the dataset is reserved for testing the model. There are many types of Cross-Validation out of which K Fold Cross Validation is mostly used. In K Fold Cross Validation, the original dataset is divided into k subsets. The subsets are known as folds. This is repeated k times where 1-fold is used for testing purposes. Rest k-1 folds are used for training the model. So, each data point acts as a test subject for the model as well as acts as the training subject. It is seen that this technique generalizes the model well and reduces the error rate

Holdout is the simplest approach. It is used in neural networks as well as in many classifiers. In this technique, the dataset is divided into train and test datasets. The dataset is usually divided into ratios like 70:30 or 80:20. Normally a large percentage of data is used for training the model and a small portion of the dataset is used for testing the model.

6. *Improvement*

Certainly! Improving machine learning performance is a fascinating challenge. Here are some strategies you can explore:

1. ***Improve Performance with Data****:*
   - ***Get More Data****: Increasing the amount of training data often leads to better model performance. Modern nonlinear techniques like deep learning thrive with more data.*
   - ***Invent More Data****: If obtaining more data is challenging, consider generating new data. You can augment or permute existing data or use probabilistic models to create synthetic data.*
   - ***Clean Your Data****: Ensure your data is of high quality. Fix missing or corrupt observations, and remove outlier values.*
   - ***Resample Data****: Adjust the size or distribution of your dataset. You can use smaller samples for faster experiments or oversample/under sample specific observations.*
   - ***Reframe Your Problem****: Sometimes changing the type of prediction problem (e.g., regression, classification, time series) can lead to better results*[1].
2. ***Treat Missing and Outlier Values****:*
   - *Address missing data and outliers in your training set. These can negatively impact model accuracy. Impute missing values or remove outliers to improve data quality.*
3. ***Feature Engineering****:*
   - *Create new features from existing ones. Feature engineering can significantly impact model performance. Consider domain-specific transformations or interactions between features.*
4. ***Feature Selection****:*
   - *Choose relevant features for your model. Removing irrelevant or redundant features can simplify the problem and enhance performance.*
5. ***Multiple Algorithms****:*
   - *Experiment with different algorithms. Some algorithms may perform better on specific types of data or problems. Try a variety of models to find the best fit.*

6. ***Algorithm Tuning***:
    - *Optimize hyperparameters for your chosen algorithm. Hyperparameters control the behavior of the*
    - *model. Techniques like grid search or random search can help find optimal values*[2].
7. ***Ensemble Methods***:
    - *Combine predictions from multiple models. Techniques like bagging (Bootstrap Aggregating), boosting, and stacking can improve overall performance.*

Remember that there's no one-size-fits-all solution. Experiment, iterate, and adapt based on your specific problem and dataset. Happy machine learning!

8. Website and deployment

Certainly! Deploying a machine learning model as a web application is a great way to showcase your work and allow others to interact with it. Here are some steps to get you started:

1. ***Build Your Machine Learning Model**: First, you need a trained machine learning model. If you've already built one, great! If not, consider creating an image recognition model using Convolutional Neural Networks (CNNs) or any other model that interests you*[1]*.*
2. ***Create a Web Application**:*
   - *Choose a web framework like Flask or Django. Flask is lightweight and easy to use for small projects.*
   - *Set up your project structure. Create HTML templates for your web pages.*
   - *Define routes in your Flask app to handle different URLs.*
3. ***Deploy on a Web Server**:*
   - *Use a cloud service like Digital Ocean to set up a virtual server (Droplet) running Ubuntu.*
   - *Install necessary software (e.g., Python, Flask, and any other dependencies).*
   - *Upload your model file (e.g., a .h5 file) to the server.*
   - *Deploy your Flask app on the server.*
4. ***Test Your Web App**:*
   - *Access your web app using the server's IP address or domain name.*
   - *Allow users to upload images and make predictions using your model.*
   - *Display the results on the web page.*
5. ***Refine and Optimize**:*
   - *As you gain more experience, improve your web app by following best practices.*
   - *Consider adding user authentication, handling larger files, and optimizing performance.*

Remember, this is just a high-level overview. You can find detailed tutorials and examples online to guide you through each step. Good luck with your machine learning deployment project!

CHAPTER 7: - PROJECT IMPLEMENTATION

7.1 GENERAL

Python is a program that was originally designed to simplify the implementation of numerical linear algebra routines. It has since grown into something much bigger, and it is used to implement numerical algorithms for a wide range of applications

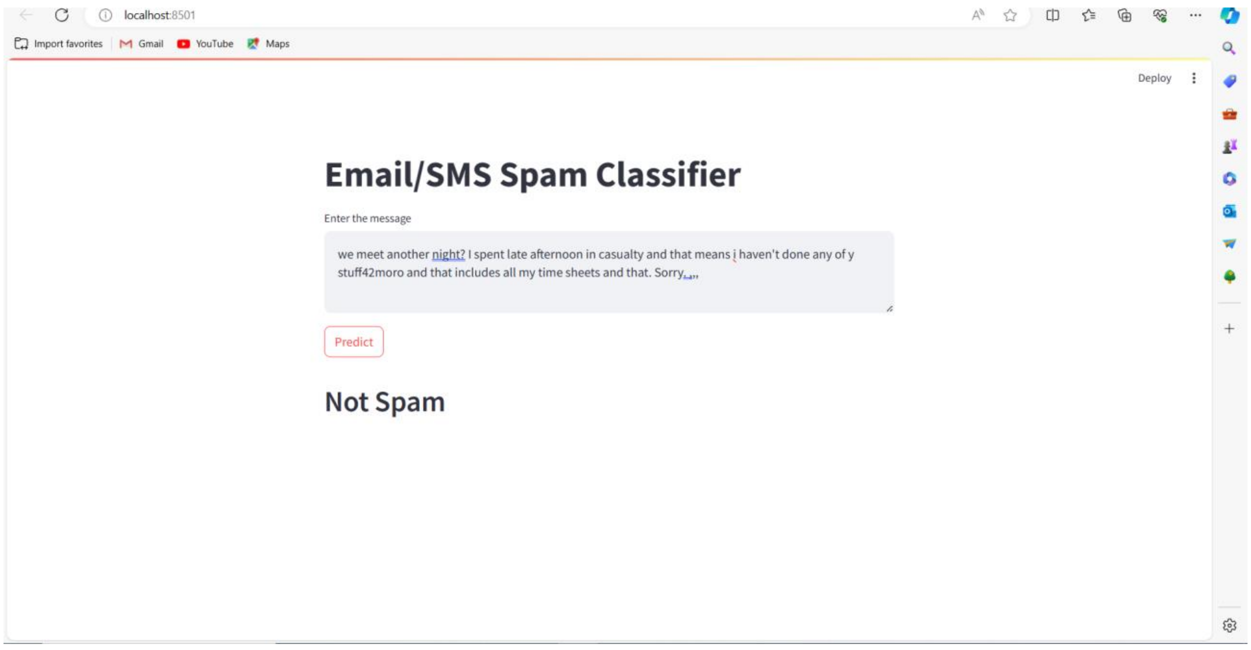
7.2 CODE IMPLEMENTATION

```
import streamlit as st
import pickle
import string
from nltk. corpus import stopwords
import nltk
nltk.download('punkt')

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)

    y = []
    for i in text:
        if i.isalnum():
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        if i not in stopwords.words('english') and i not in
```

```
string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))


  text = text.lower()
    text = nltk.word_tokenize(text)

  y = []
    for i in text:
        if i.isalnum():

            y. append(i)

    text = y[:]
    y. clear()


for i in text:
        if i not in stopwords.words('english') and i not in
string.punctuation:
            y.append(i)

    text = y[:]
    y.clear()

    for i in text:
        y.append(ps.stem(i))

    return " ".join(y)

tfidf = pickle.load(open('vectorizer.pkl','rb'))
```

```
model = pickle.load(open('model.pkl','rb'))

st.title("Email/SMS Spam Classifier")

input_sms = st.text_area("Enter the message")

if st.button('Predict'):

    # 1. preprocess
    transformed_sms = transform_text(input_sms)
    # 2. vectorize
    vector_input = tfidf.transform([transformed_sms])
    # 3. predict
    result = model.predict(vector_input)[0]
    # 4. Display
    if result == 1:
        st.header("Spam")
    else:
        st.header("Not Spam")
```

SNAPSHOOTS OF SPAM AND NOT SPAM



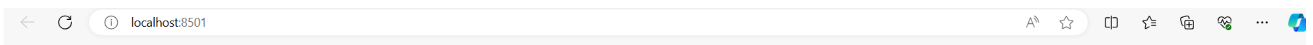
Email/SMS Spam Classifier

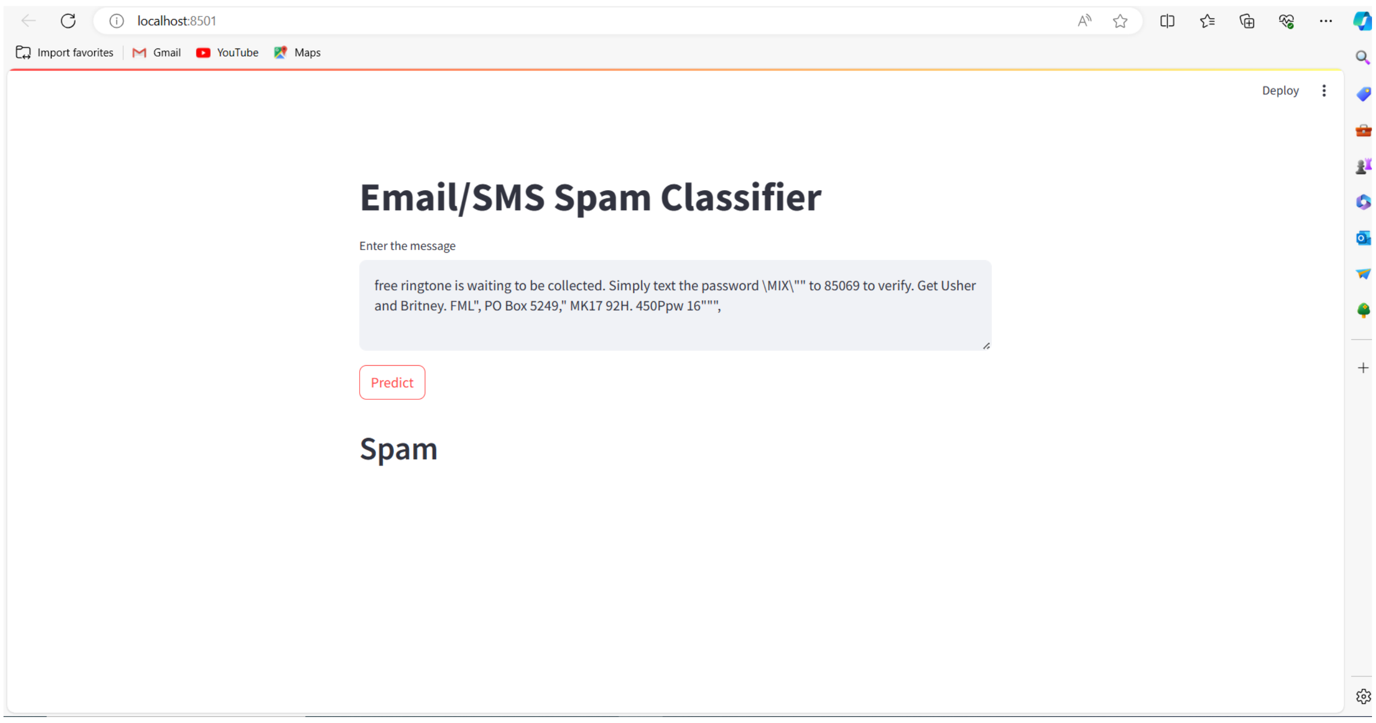
Enter the message

we meet another night? I spent late afternoon in casualty and that means i haven't done any of y stuff42moro and that includes all my time sheets and that. Sorry...

Predict

Not Spam





Email/SMS Spam Classifier

Enter the message

free ringtone is waiting to be collected. Simply text the password \MIX\"" to 85069 to verify. Get Usher and Britney. FML", PO Box 5249," MK17 92H. 450Ppw 16""",

Spam

Streamlit :-

Streamlit is an open-source Python library that makes it easy to create and share custom web apps for machine learning and data science. By using Streamlit you can quickly build and deploy powerful data applications. For more information about the open-source library, see the Streamlit Library documentation.

Streamlit in Snowflake helps developers securely build, deploy, and share Streamlit apps on Snowflake's data cloud. Using Streamlit in Snowflake, you can build applications that process and use data in Snowflake without moving data or application code to an external system.

CHAPTER 8: - CONCLUSION AND REFERENCES

8.1 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 f-measure 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. Team ID: 420772
Introduction

APPLICATION:

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

8.2 REFERENCES:

- *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.*

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

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

- *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.*

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

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

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

- *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.*