**SMART**

**AI SPAM**

**CLASSIFIER**

**USING**

**AI**

**PHASE 5 SUBMISSION.**

# DESIGN THINKING:

# **PROBLEM DEFINITION:**

**Spam detection is a supervised machine learning problem. This means you must provide your machine learning model with a set of examples of spam and ham messages and let it find the relevant patterns that separate the two different categories.**

# **DEFINITION:**

**In order to more effectively analyze the content and not trash a real message, sophisticated spam filters use artificial intelligence (AI) techniques that look for key words and attempt to decipher their meaning in sentences (see Bayesian filtering). See spam trap, spam relay and spamdexing.**

# **MODEL SELECTION:**

**Naive Bayes:**

**Pros: Simple and computationally efficient, making it a good choice for small to medium-sized datasets. It often performs well when the assumption of feature independence holds reasonably.**

**Cons: May not capture complex relationships in data, as it assumes that features are independent.**

# **Support Vector Machines (SVM):**

**Pros: Effective for high-dimensional data and can capture complex decision boundaries. SVMs can perform well with proper tuning**

# **Random Forest:**

**Pros: Ensemble models like Random Forest can handle complex relationships in data, are robust to overfitting, and can provide feature importance scores.**

**Cons: Requires more data than simpler models to perform well, and hyperparameter tuning is necessary.**

# **EVALUATION:**

**F1-Score:**

**The F1-Score is the harmonic mean of precision and recall and provides a balance between the two metrics. It is calculated as 2 \* (Precision \* Recall) / (Precision + Recall).**

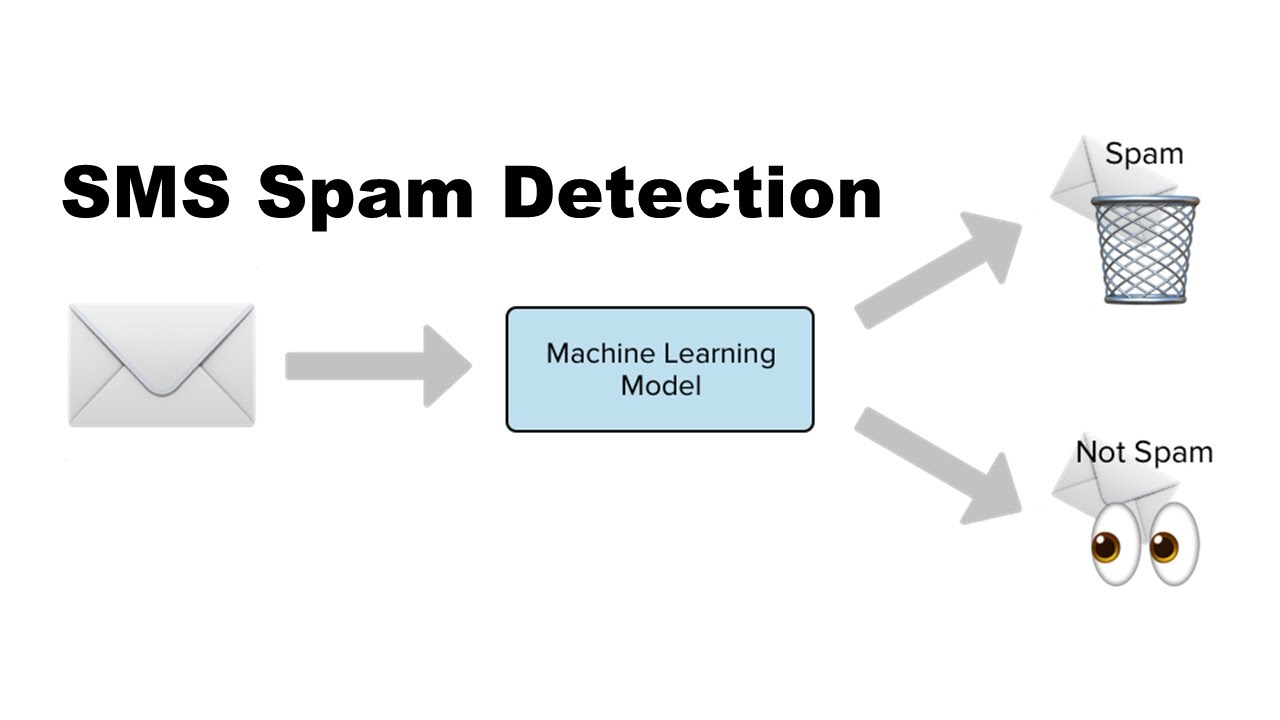
**The F1-Score is especially useful when you want to balance precision and recall in your spam detector.**

**INNOVATION:**

# **Problem Description:**

Understanding the problem is a crucial first step in solving any machine learning problem. In this article, we will explore and understand the process of classifying emails as spam or not spam. This is called Spam Detection, and it is a binary classification problem.

The reason to do this is simple: by detecting unsolicited and unwanted emails, we can prevent spam messages from creeping into the user’s inbox, thereby improving user experience.



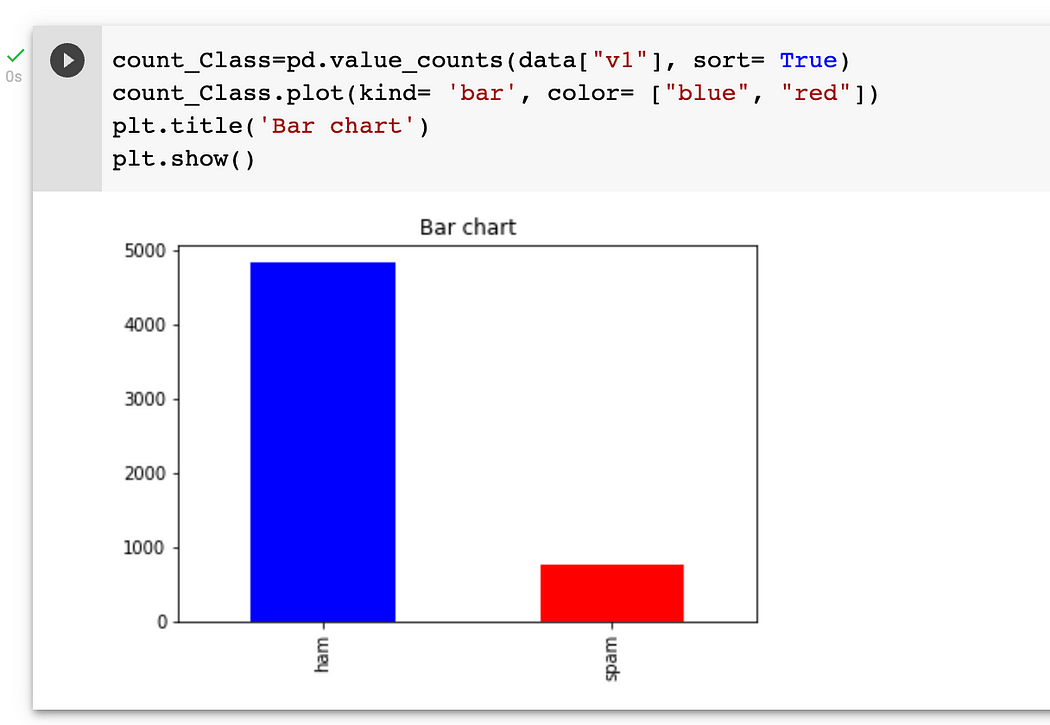
**MACHINE LEARNING ALGORITHM :**

**NAÏVE BAYES:**

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features. The probability of an event ‘A’ occurring, given the condition ‘B’, is calculated .

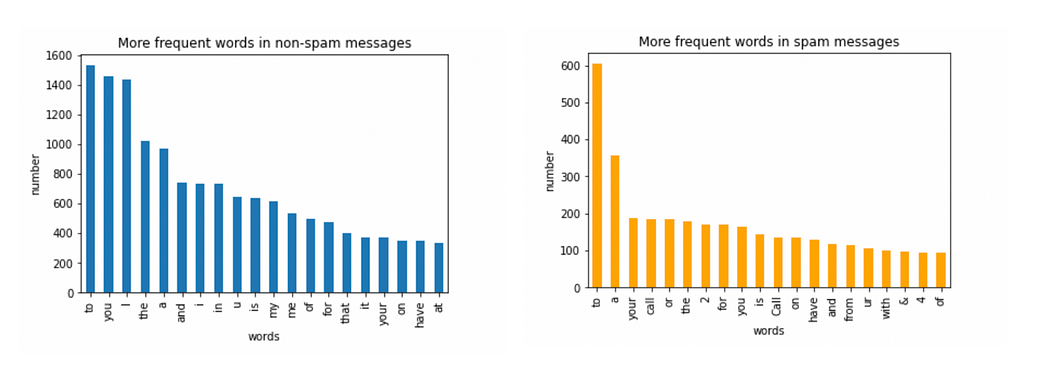


**Visualizing the number of spam and non-spam(ham) messages**



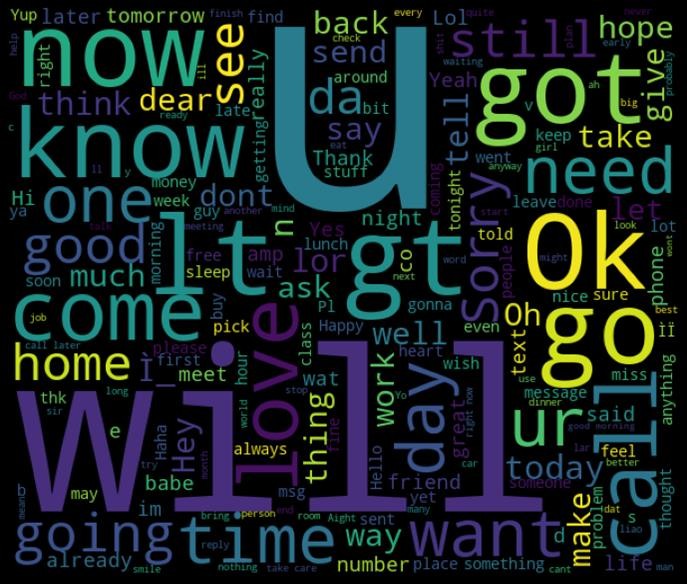
As we can see, there are around 4800 ham messages and around 800 messages spam messages. The classification of spam messages will have the spam classification status — 1, as it is the event of our interest and occurs less often than the ham messages.

**Frequent words in spam and non-spam messages.**



We can see that the majority of frequent words in both classes are stop words such as ‘to’, ‘a’, ‘or’ and so on. These common words won’t be significant while classifying whether a message is significant or not. We need to analyze the presence of more relevant words in a spam message. So, to remove the common words, we can use **stopwords** function in Natural Language Toolkit library (nltk).

**RANDOM FOREST:**



By implementing these innovative techniques and

strategies you can improve the effectiveness and adaptability of your AI spam detector in combating spam digital platforms .







INTRODUCTION:















:







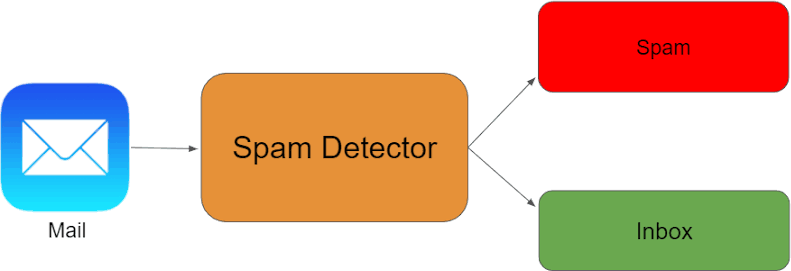




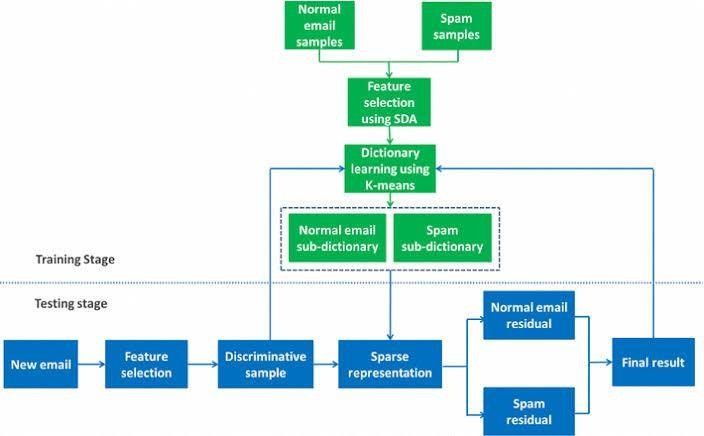
































**CLASSIFIERS:**

**from sklearn.naive\_bayes import \***

**from sklearn.dummy import \***

**from sklearn.ensemble import \***

**from sklearn.neighbors import \***

**from sklearn.tree import \***

**from sklearn.feature\_extraction.text import CountVectorizer**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.feature\_extraction.text import HashingVectorizer**

**from sklearn.calibration import \***

**from sklearn.linear\_model import \***

**from sklearn.multiclass import \***

**from sklearn.svm import \***

**import pandas**

**def perform(classifiers, vectorizers, train\_data, test\_data):**

**for classifier in classifiers:**

**for vectorizer in vectorizers:**

**string = ''**

**string += classifier.\_\_class\_\_.\_\_name\_\_ + ' with ' + vectorizer.\_\_class\_\_.\_\_name\_\_**

**# train**

**vectorize\_text = vectorizer.fit\_transform(train\_data.v2)**

**classifier.fit(vectorize\_text, train\_data.v1)**

**# score**

**vectorize\_text = vectorizer.transform(test\_data.v2)**

**score = classifier.score(vectorize\_text, test\_data.v1)**

**string += '. Has score: ' + str(score)**

**print(string)**

**data = pandas.read\_csv('spam.csv', encoding='latin-1')**

**learn = data[:4400] # 4400 items**

**test = data[4400:] # 1172 items**

**perform(**

**[**

**BernoulliNB(),**

**RandomForestClassifier(n\_estimators=100, n\_jobs=-1),**

**AdaBoostClassifier(),**

**BaggingClassifier(),**

**ExtraTreesClassifier(),**

**GradientBoostingClassifier(),**

**DecisionTreeClassifier(),**

**CalibratedClassifierCV(),**

**DummyClassifier(),**

**PassiveAggressiveClassifier(),**

**RidgeClassifier(),**

**RidgeClassifierCV(),**

**SGDClassifier(),**

**OneVsRestClassifier(SVC(kernel='linear')),**

**OneVsRestClassifier(LogisticRegression()),**

**KNeighborsClassifier()**

**],**

**[**

**CountVectorizer(),**

**TfidfVectorizer(),**

**HashingVectorizer()**

**],**

**learn,**

**test**

**)**

**SCIKIT:**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.calibration import \***

**from sklearn.linear\_model import \***

**from sklearn.multiclass import \***

**from sklearn.svm import \***

**import pandas**

**data = pandas.read\_csv('spam.csv', encoding='latin-1')**

**train\_data = data[:4400] # 4400 items**

**test\_data = data[4400:] # 1172 items**

**classifier = OneVsRestClassifier(SVC(kernel='linear'))**

**vectorizer = TfidfVectorizer()**

**# train**

**vectorize\_text = vectorizer.fit\_transform(train\_data.v2)**

**classifier.fit(vectorize\_text, train\_data.v1)**

**vectorize\_text = vectorizer.transform(test\_data.v2)**

**score = classifier.score(vectorize\_text, test\_data.v1)**

**print(score) # 98,8**

**APP.PY**

**import os**

**from flask import Flask, render\_template, request, redirect, url\_for, jsonify**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.multiclass import \***

**from sklearn.svm import \***

**import pandas**

**app = Flask(\_\_name\_\_)**

**global Classifier**

**global Vectorizer**

**# load data**

**data = pandas.read\_csv('spam.csv', encoding='latin-1')**

**train\_data = data[:4400] # 4400 items**

**test\_data = data[4400:] # 1172 items**

**# train model**

**Classifier = OneVsRestClassifier(SVC(kernel='linear', probability=True))**

**Vectorizer = TfidfVectorizer()**

**vectorize\_text = Vectorizer.fit\_transform(train\_data.v2)**

**Classifier.fit(vectorize\_text, train\_data.v1)**

**@app.route('/', methods=['GET'])**

**def index():**

**message = request.args.get('message', '')**

**error = ''**

**predict\_proba = ''**

**predict = ''**

**global Classifier**

**global Vectorizer**

**try:**

**if len(message) > 0:**

**vectorize\_message = Vectorizer.transform([message])**

**predict = Classifier.predict(vectorize\_message)[0]**

**predict\_proba = Classifier.predict\_proba(vectorize\_message).tolist()**

**except BaseException as inst:**

**error = str(type(inst).\_\_name\_\_) + ' ' + str(inst)**

**return jsonify(**

**message=message, predict\_proba=predict\_proba,**

**predict=predict, error=error)**

**if \_\_name\_\_ == '\_\_main\_\_':**

**port = int(os.environ.get('PORT', 5000))**

**app.run(host='0.0.0.0', port=port, debug=True, use\_reloader=True)**

**TEST SCORE.PY**

**from sklearn.naive\_bayes import \***

**from sklearn.dummy import \***

**from sklearn.ensemble import \***

**from sklearn.neighbors import \***

**from sklearn.tree import \***

**from sklearn.feature\_extraction.text import CountVectorizer**

**from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.feature\_extraction.text import HashingVectorizer**

**from sklearn.calibration import \***

**from sklearn.linear\_model import \***

**from sklearn.multiclass import \***

**from sklearn.svm import \***

**import pandas**

**import csv**

**data = pandas.read\_csv('spam.csv', encoding='latin-1')**

**train\_data = data[:4400] # 4400 items**

**test\_data = data[4400:] # 1172 items**

**classifier = OneVsRestClassifier(SVC(kernel='linear'))**

**vectorizer = TfidfVectorizer()**

**# train**

**vectorize\_text = vectorizer.fit\_transform(train\_data.v2)**

**classifier.fit(vectorize\_text, train\_data.v1)**

**# score**

**# vectorize\_text = vectorizer.transform(test\_data.v2)**

**# score = classifier.score(vectorize\_text, test\_data.v1)**

**# print(score) # 98,8**

**csv\_arr = []**

**for index, row in test\_data.iterrows():**

**answer = row[0]**

**text = row[1]**

**vectorize\_text = vectorizer.transform([text])**

**predict = classifier.predict(vectorize\_text)[0]**

**if predict == answer:**

**result = 'right'**

**else:**

**result = 'wrong'**

**csv\_arr.append([len(csv\_arr), text, answer, predict, result])**

**# write csv**

**with open('test\_score.csv', 'w', newline='') as csvfile:**

**spamwriter = csv.writer(csvfile, delimiter=';',**

**quotechar='"', quoting=csv.QUOTE\_MINIMAL)**

**spamwriter.writerow(['#', 'text', 'answer', 'predict', result])**

**for row in csv\_arr:**

**spamwriter.writerow(row)**



**SPAM.CSV**

| **v1** | **v2** |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | ham | Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat... |  |  |  |
| 3 | ham | Ok lar... Joking wif u oni... |  |  |  |
| 4 | spam | Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's |  |  |  |
| 5 | ham | U dun say so early hor... U c already then say... |  |  |  |
| 6 | ham | Nah I don't think he goes to usf, he lives around here though |  |  |  |
| 7 | spam | FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, å£1.50 to rcv |  |  |  |
| 8 | ham | Even my brother is not like to speak with me. They treat me like aids patent. |  |  |  |
| 9 | ham | As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press \*9 to copy your friends Callertune |  |  |  |
| 10 | spam | WINNER!! As a valued network customer you have been selected to receivea å£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only. |  |  |  |
| 11 | spam | Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030 |  |  |  |
| 12 | ham | I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today. |  |  |  |
| 13 | spam | SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info |  |  |  |
| 14 | spam | URGENT! You have won a 1 week FREE membership in our å£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18 |  |  |  |
| 15 | ham | I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times. |  |  |  |
| 16 | ham | I HAVE A DATE ON SUNDAY WITH WILL!! |  |  |  |
| 17 | spam | XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL |  |  |  |
| 18 | ham | Oh k...i'm watching here:) |  |  |  |
| 19 | ham | Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet. |  |  |  |
| 20 | ham | Fine if thatåÕs the way u feel. ThatåÕs the way its gota b |  |  |  |
| 21 | spam | England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/Ì¼1.20 POBOXox36504W45WQ 16+ |  |  |  |
| 22 | ham | Is that seriously how you spell his name? |  |  |  |
| 23 | ham | I‰Û÷m going to try for 2 months ha ha only joking |  |  |  |
| 24 | ham | So Ì\_ pay first lar... Then when is da stock comin... |  |  |  |
| 25 | ham | Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already? |  |  |  |
| 26 | ham | Ffffffffff. Alright no way I can meet up with you sooner? |  |  |  |
| 27 | ham | Just forced myself to eat a slice. I'm really not hungry tho. This sucks. Mark is getting worried. He knows I'm sick when I turn down pizza. Lol |  |  |  |
| 28 | ham | Lol your always so convincing. |  |  |  |
| 29 | ham | Did you catch the bus ? Are you frying an egg ? Did you make a tea? Are you eating your mom's left over dinner ? Do you feel my Love ? |  |  |  |
| 30 | ham | I'm back &amp; we're packing the car now, I'll let you know if there's room |  |  |  |
| 31 | ham | Ahhh. Work. I vaguely remember that! What does it feel like? Lol |  |  |  |

**OUTPUT:**

**THANK**

**YOU..**