NAAN MUDHALVAN PROJECT REPORT

ON

"OPTIMIZING SPAM FILTERING WITH MACHINE LEARNING"

Submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

SUBMITTED BY

TEAM LEADER

LHONA PADMAYUKY S (Reg no. 20201371506113)

TEAM MEMBERS

SUBHADHAARANI B (Reg no 20201371506140)

SUREKA V (Reg no. 20201371506143)

VIVEKA PRIYA J (Reg no.20201371506148)

Department of Computer Science

Government Arts and Science College for Women – Alangulam

Tenkasi – 627851

April 2023

1. Introduction

1.1 – Overview

Optimizing spam filtering is a machine learning project that involves algorithms to automatically classify incoming SMS spam or legitimate messages, from reaching the user's inbox. The goal is to accurately filter out unwanted messages and minimize false positives, while ensuring that important messages are not mistakenly labeled as spam.

1.2 - Purpose

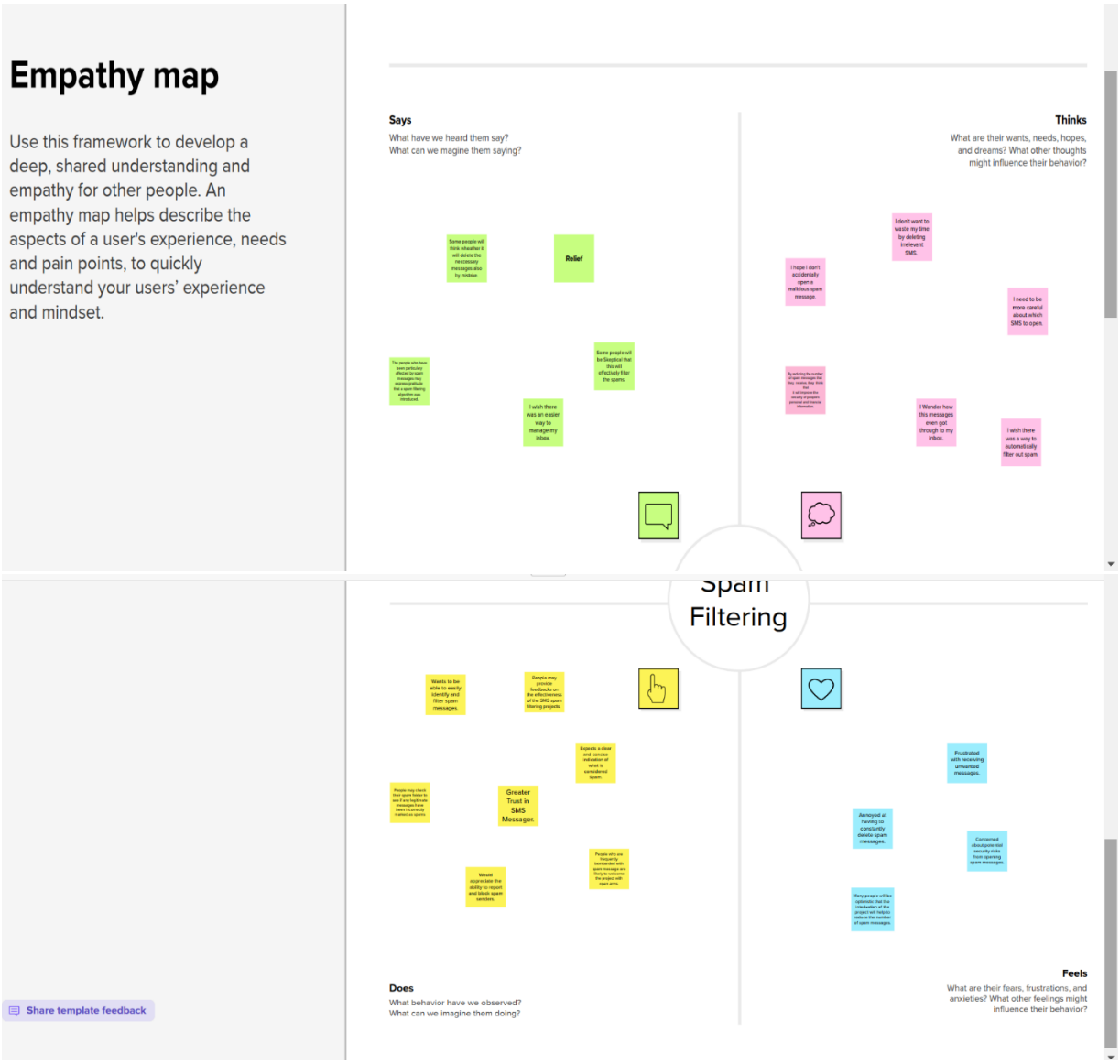
To achieve this, the project typically involves gathering a large dataset of messages that have already been classified as either spam or not-spam. This dataset is used to train a machine learning model, which can then classify new incoming messages based on their content and other features such as the sender's address, subject line, and attachments.

The machine learning model is typically trained using a variety of techniques, such as natural language processing (NLP), statistical modeling, and deep learning algorithms. The model is continuously updated and optimized based on feedback from users, which helps to improve its accuracy over time.

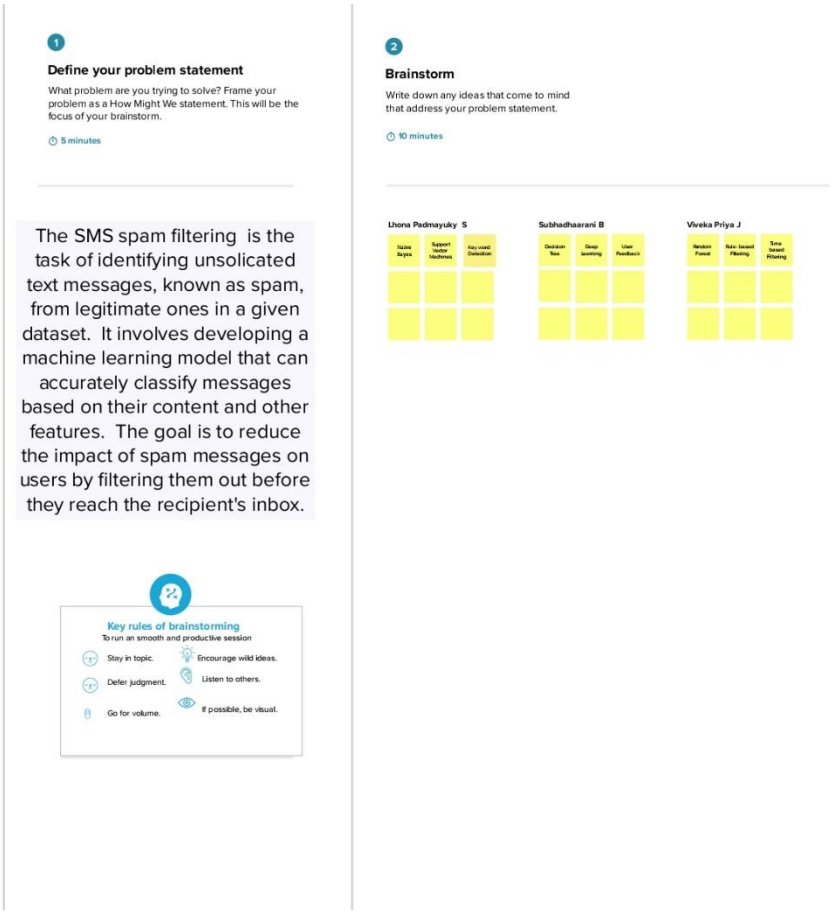
The ultimate goal of the project is to create a spam filter that is highly effective at identifying and blocking unwanted messages, while minimizing false positives and ensuring that important messages are not mistakenly flagged as spam. This can help to improve productivity and reduce the risk of security threats, such as phishing scams and malware infections.

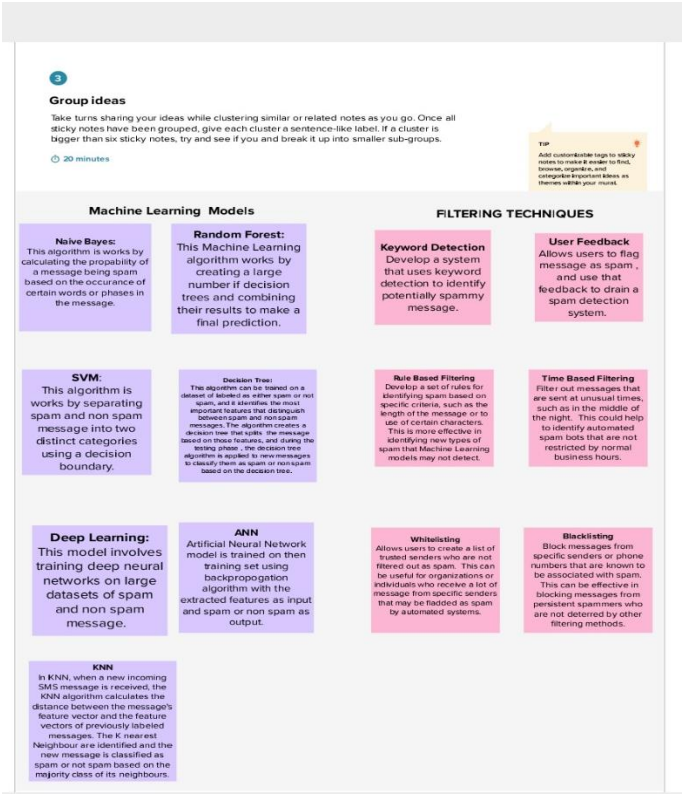
2. PROBLEM DEFINITION & DESIGN THINKING

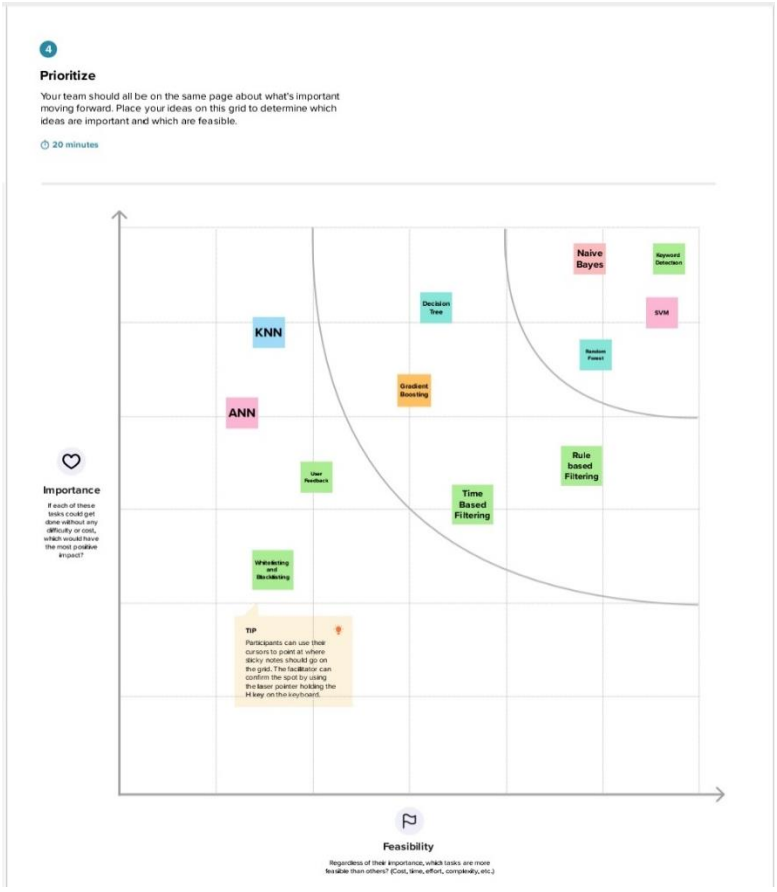
2.1. Empathy Map



2.2. Ideation & Brainstorming Map




3
Group ideas
Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
Machine Learning Models
FILTERING TECHNIQUES
Naive Bayes: This algorithm is works by calculating the propability of a message being spam based on the occurance of certain words or phases in the message.
Random Forest: This Machine Learning algorithm works by creating a large number if decision trees and combining their results to make a final prediction.
Keyword Detection Develop a system that uses keyword detection to identify potentially spammy message.
User Feedback Allows users to flag message as spam , and use that feedback to drain a spam detection system.
SVM: This algorithm is works by separating spam and non spam message into two distinct categories using a decision boundary.
Decision Tree: The algorithm can be trained on a dataset of labelled as either spam or not spam, and it identifies the most important features that distinguish between spam and non spam messages. The algorithm creates a decision tree that splits the message based on those features, and during the testing phase , the decision tree algorithm is applied to new messages to classify them as spam or non spam based on the decision tree.
Rule Based Filtering Develop a set of rules for identifying spam based on specific criteria, such as the length of the message or to use of certain characters. This is more effective in identifying new types of spam that Machine Learning models may not detect.
Time Based Filtering Filter out messages that are sent at unusual times, such as in the middle of the night. This could help to identify automated spam bots that are not restricted by normal business hours.
Deep Learning: This model involves training deep neural networks on large datasets of spam and non spam message.
ANN Artificial Neural Network model is trained on then training set using backpropogation algorithm with the extracted features as input and spam or non spam as output.
Whitelisting Allows users to create a list of trusted senders who are not filtered out as spam. This can be useful for organizations or individuals who receive a lot of message from specific senders that may be fladded as spam by automated systems.
Blacklisting Block messages from specific senders or phone numbers that are known to be associated with spam. This can be effective in blocking messages from persistent spammers who are not deterred by other filtering methods.
KNN In KNN, when a new incoming SMS message is received, the KNN algorithm calculates the distance between the message's feature vector and the feature vectors of previously labeled messages. The K nearest Neighbour are identified and the new message is classified as spam or not spam based on the majority class of its neighbours.


4
Prioritize
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.
20 minutes
Naive Bayes
Keyword Detection
Decision Tree
SVM
KNN
Random Forest
Gradient Boosting
ANN
Rule based Filtering
User Feedback
Time Based Filtering
Whitelisting and Blacklisting
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3. Result

4. Advantage and Disadvantages

Advantages:

- Saves time and resources by automating the process of identifying and filtering unwanted messages which can be time-consuming and tedious for humans to do manually.

- Helps to protect against phishing scams, malware, and other security threats that can be delivered through spam messages and emails.

- Improves the user experience by reducing the amount of unwanted messages that the users have to sort through.

Disadvantages:

- There is always a risk of false positives or false negatives, which can result in legitimate messages being incorrectly filtered as spam messages .
- Over-aggressive spam filtering can lead to legitimate messages being incorrectly flagged as spam, resulting in important messages being missed.
- Some users may find spam filtering to be intrusive or may be concerned about privacy issues related to the automated analysis of their messages .

5. APPLICATIONS

Spam filtering algorithms are used in a variety of applications to automatically identify and remove unwanted or unsolicited messages, emails, or other types of communication.

1. **SMS:** Spam filtering algorithms are used in SMS messaging services to identify and block unsolicited or fraudulent text messages.

2. **Email:** Spam filtering algorithms are commonly used in email clients and servers to automatically filter out unwanted or unsolicited emails, such as promotional emails or phishing scams.

3. **Social media:** Social media platforms use spam filtering algorithms to detect and remove spammy comments or messages that violate community guidelines.

4. **Network security:** Spam filtering algorithms are used in network security applications to identify and block emails and other communication that may contain viruses or malware.

5. **Online advertising:** Spam filtering algorithms are used in online advertising platforms to detect and remove fraudulent or low-quality ads that may be misleading or harmful.

6. **Mobile apps:** Spam filtering algorithms are used in mobile apps to identify and remove spammy or low-quality user-generated content.

6.CONCLUSION

In this project, I have used the Kaggle Spam dataset . I have used four models to find the accuracy. The models are Decision tree, Random forest, Naïve Bayes and ANN. Among these models ANN gives the best accuracy rate of 99% . So, I have used ANN to predict my Message as spam or non spam. And then I have build the python code and imported flask to direct the webpages. Home and Result are the two html file that I have Created. And atlast, my model has predicted its correct output.

7.FEATURE SCOPE :

- **Machine learning algorithms**: One effective approach to spam filtering is to use machine learning algorithms to detect patterns in spam messages. By analyzing large amounts of spam data, these algorithms can learn to recognize common characteristics of spam messages and filter them out.

- **Natural Language Processing (NLP):** NLP techniques can be used to identify the semantic structure of messages and detect any indications of spam messages. It can also be used to recognize patterns in the text that may indicate whether or not a message is spam.

- **Email header analysis**: Analyzing the email header information can provide important clues as to whether a message is spam or not. By examining the sender's IP address, email domain, and other header information, it is possible to identify and block known spam sources.

- **Real-time blacklists**: A real-time blacklist (RBL) is a list of known spamming sources that is constantly updated. By incorporating an RBL into a spam filter, the system can quickly identify and block messages from known spammers.

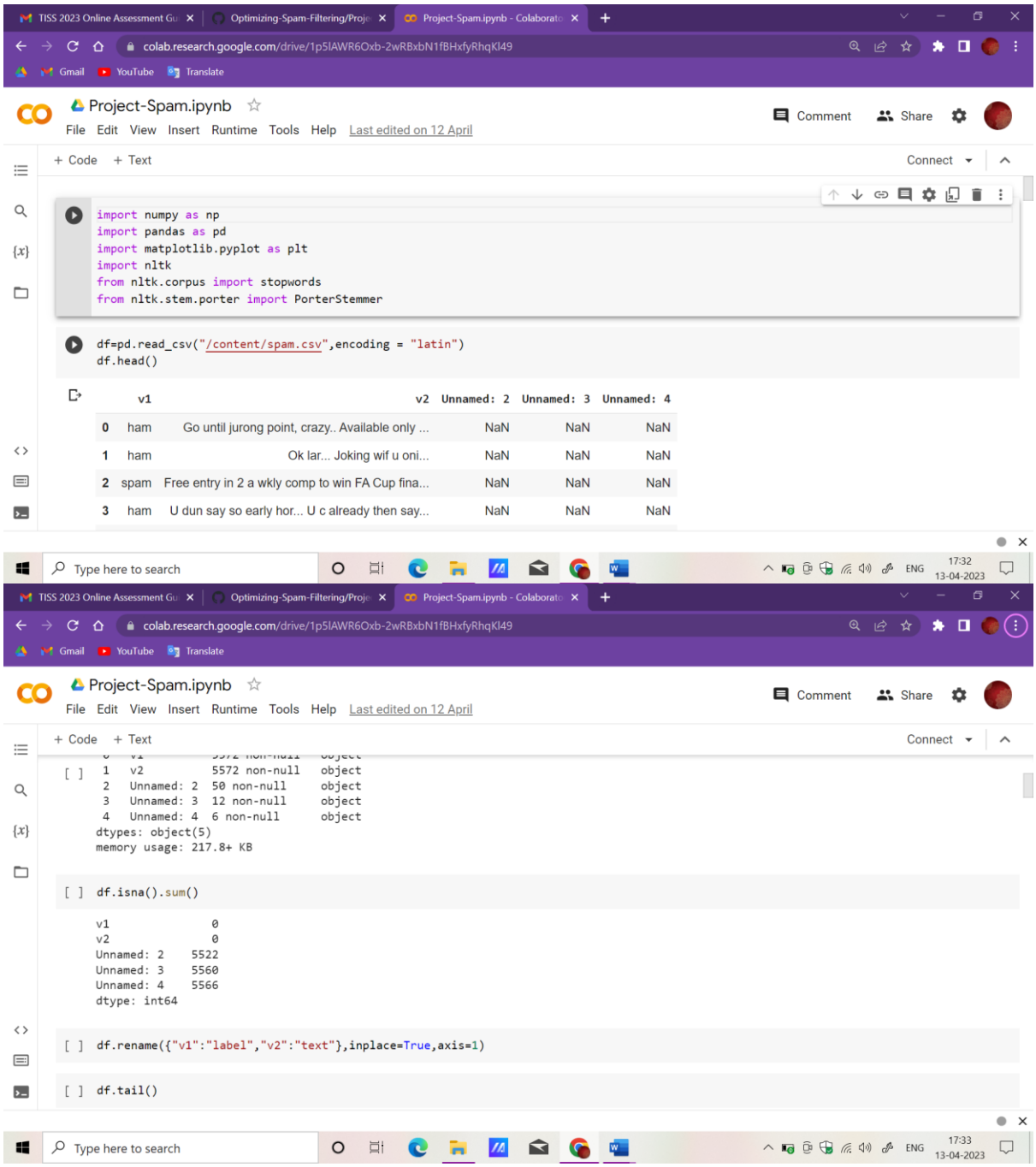
- **Bayesian filtering**: Bayesian filtering is a statistical technique that can be used to filter spam based on the probability that a message is spam. By analyzing the text and other characteristics of a message, Bayesian filtering can calculate the probability that it is spam and block it accordingly.

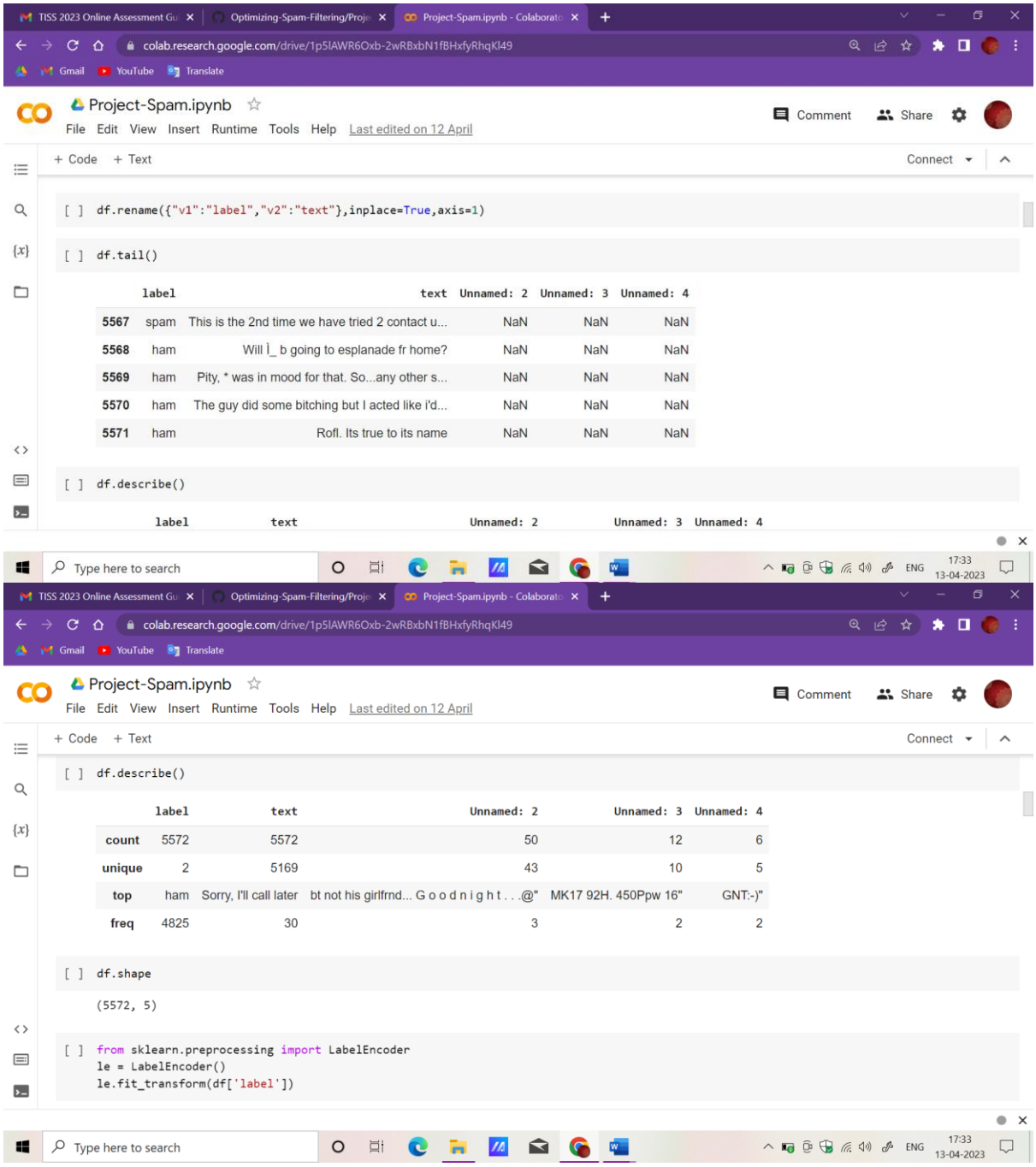
- **Sender reputation:** Sender reputation is a measure of the trustworthiness of an email sender. By analyzing the history of emails sent by a particular sender, it is possible to identify and block messages from senders with a history of sending spam.

- **User feedback:** User feedback can be used to improve the effectiveness of a spam filter over time. By allowing users

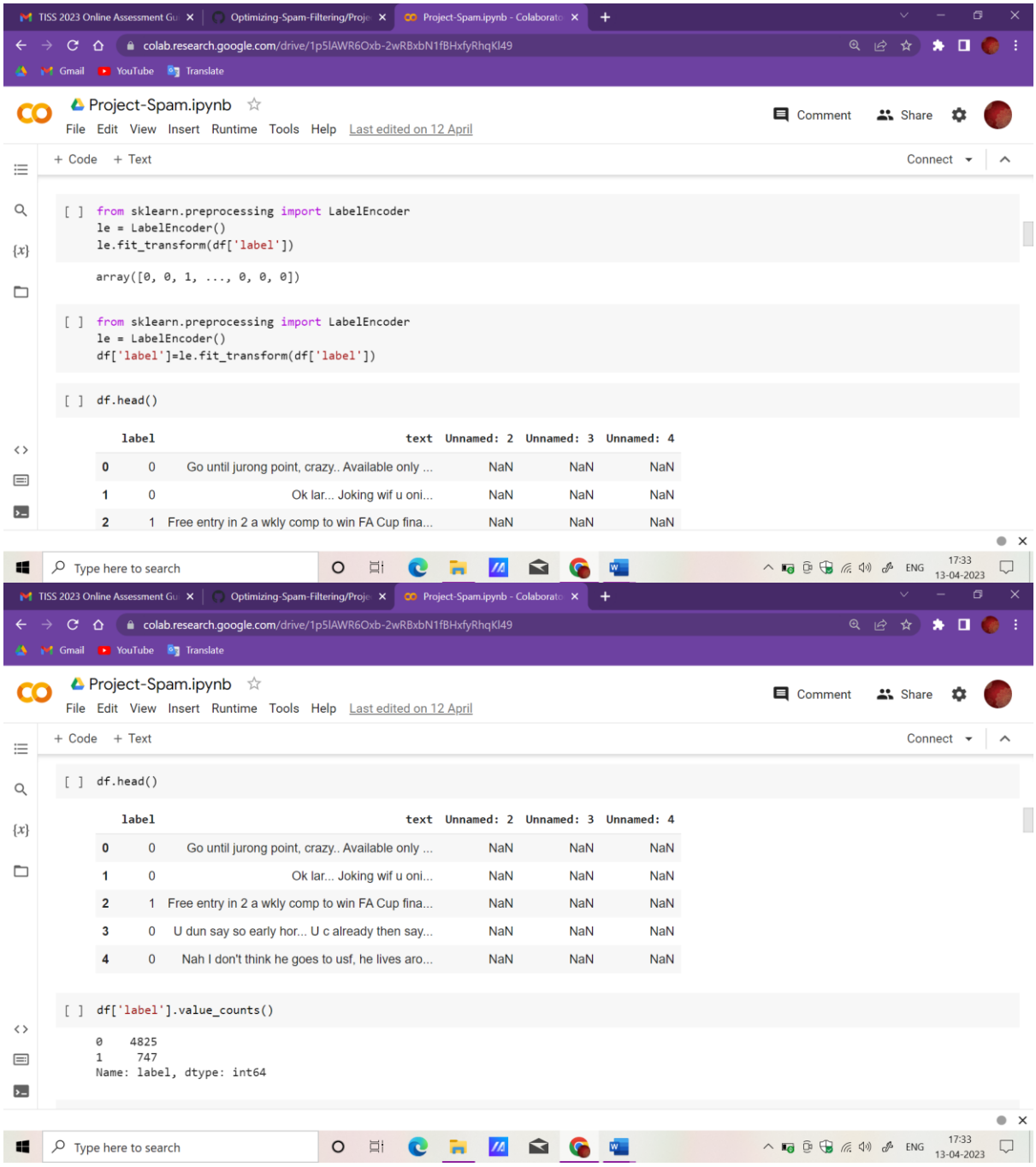
to mark messages as spam or not spam, the system can learn from these ratings and improve its accuracy..

8.APPENDIX




TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] df.rename({"v1":"label","v2":"text"},inplace=True,axis=1)
[ ] df.tail()
label text Unnamed: 2 Unnamed: 3 Unnamed: 4
5567 spam This is the 2nd time we have tried 2 contact u... NaN NaN NaN
5568 ham Will ì_ b going to esplanade fr home? NaN NaN NaN
5569 ham Pity, * was in mood for that. So...any other s... NaN NaN NaN
5570 ham The guy did some bitching but I acted like i'd... NaN NaN NaN
5571 ham Rofl. Its true to its name NaN NaN NaN
[ ] df.describe()
label text Unnamed: 2 Unnamed: 3 Unnamed: 4
Type here to search
ENG
17:33
13-04-2023

TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] df.describe()
label text Unnamed: 2 Unnamed: 3 Unnamed: 4
count 5572 5572 50 12 6
unique 2 5169 43 10 5
top ham Sorry, I'll call later bt not his girlfrnd... G o o d n i g h t . . .@" MK17 92H. 450Ppw 16" GNT:-)"
freq 4825 30 3 2 2
[ ] df.shape
(5572, 5)
[ ] from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit_transform(df['label'])
Type here to search
ENG
17:33
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit_transform(df['label'])
array([0, 0, 1, ..., 0, 0, 0])
[ ] from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['label']=le.fit_transform(df['label'])
[ ] df.head()
label text Unnamed: 2 Unnamed: 3 Unnamed: 4
0 0 Go until jurong point, crazy.. Available only ... NaN NaN NaN
1 0 Ok lar... Joking wif u oni... NaN NaN NaN
2 1 Free entry in 2 a wkly comp to win FA Cup fina... NaN NaN NaN
Type here to search
ENG
17:33
13-04-2023

TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] df.head()
label text Unnamed: 2 Unnamed: 3 Unnamed: 4
0 0 Go until jurong point, crazy.. Available only ... NaN NaN NaN
1 0 Ok lar... Joking wif u oni... NaN NaN NaN
2 1 Free entry in 2 a wkly comp to win FA Cup fina... NaN NaN NaN
3 0 U dun say so early hor... U c already then say... NaN NaN NaN
4 0 Nah I don't think he goes to usf, he lives aro... NaN NaN NaN
[ ] df['label'].value_counts()
0 4825
1 747
Name: label, dtype: int64
Type here to search
ENG
17:33
13-04-2023

TISS 2023 Online Assessment Gu | Optimizing-Spam-Filtering/Proje | Project-Spam.ipynb - Colaborato

colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49

Gmail YouTube Translate

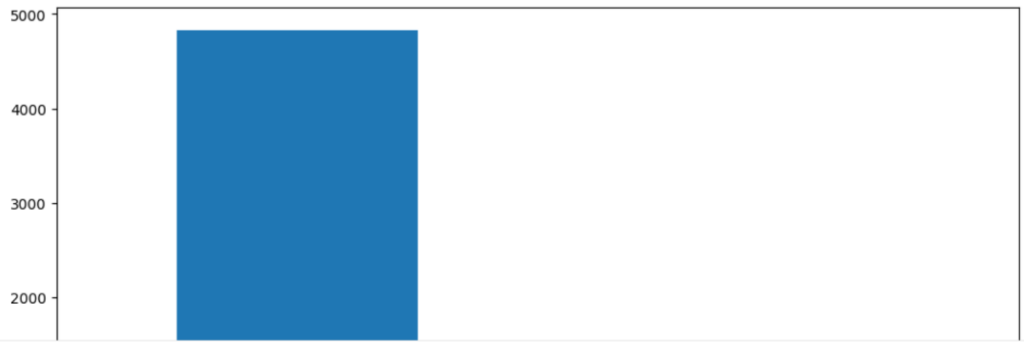
Project-Spam.ipynb

File Edit View Insert Runtime Tools Help Last edited on 12 April

Comment Share Connect

+ Code + Text

```
[ ] df["label"].value_counts().plot(kind="bar",figsize=(12,6))
    plt.xticks(np.arange(2),('Non spam','spam'),rotation=0);
```



Type here to search ENG 17:33 13-04-2023

TISS 2023 Online Assessment Gu | Optimizing-Spam-Filtering/Proje | Project-Spam.ipynb - Colaborato

colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49

Gmail YouTube Translate

Project-Spam.ipynb

File Edit View Insert Runtime Tools Help Last edited on 12 April

Comment Share Connect

+ Code + Text

+ Code + Text

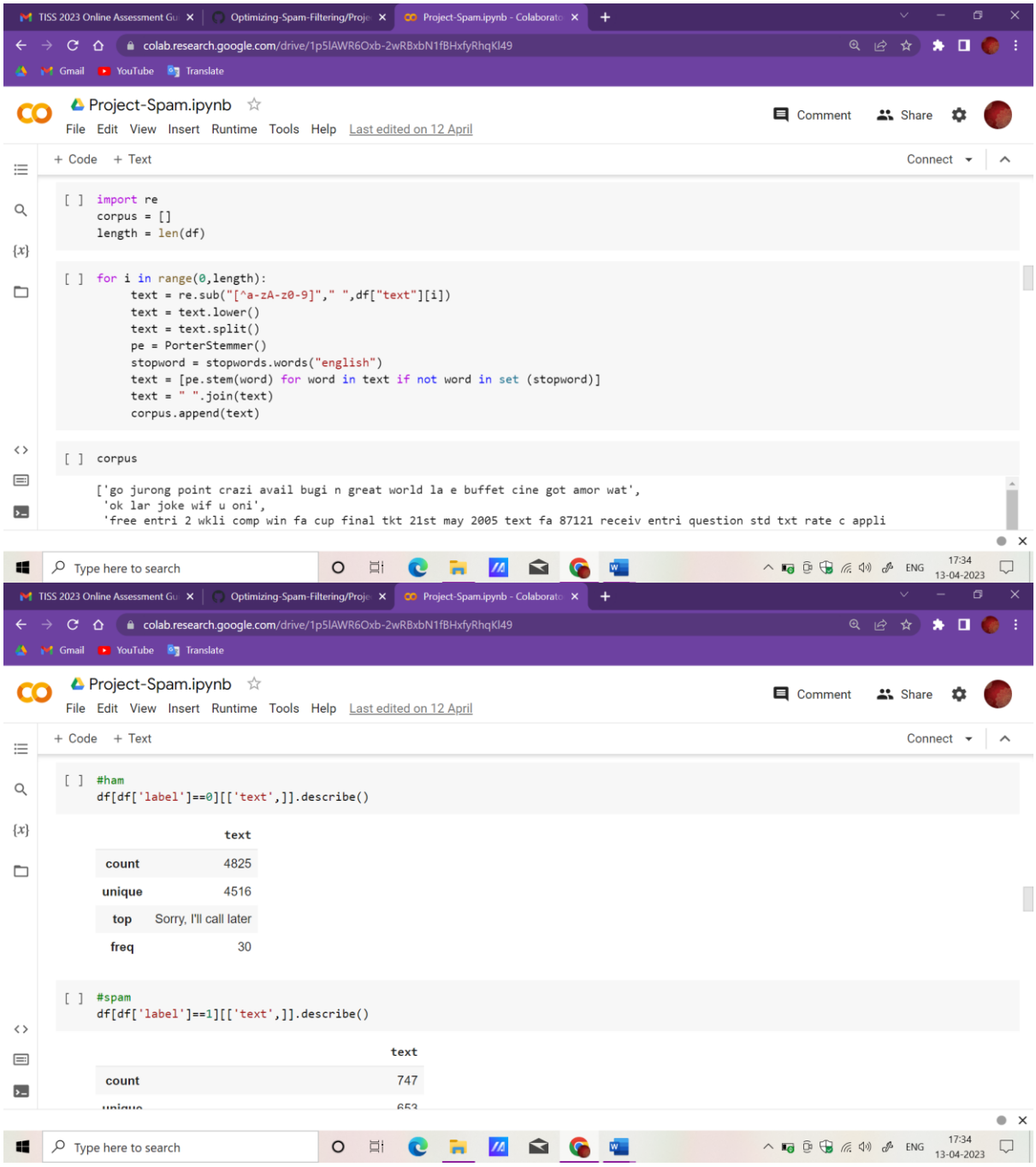
```
[ ] import nltk
    nltk.download("stopwords")
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

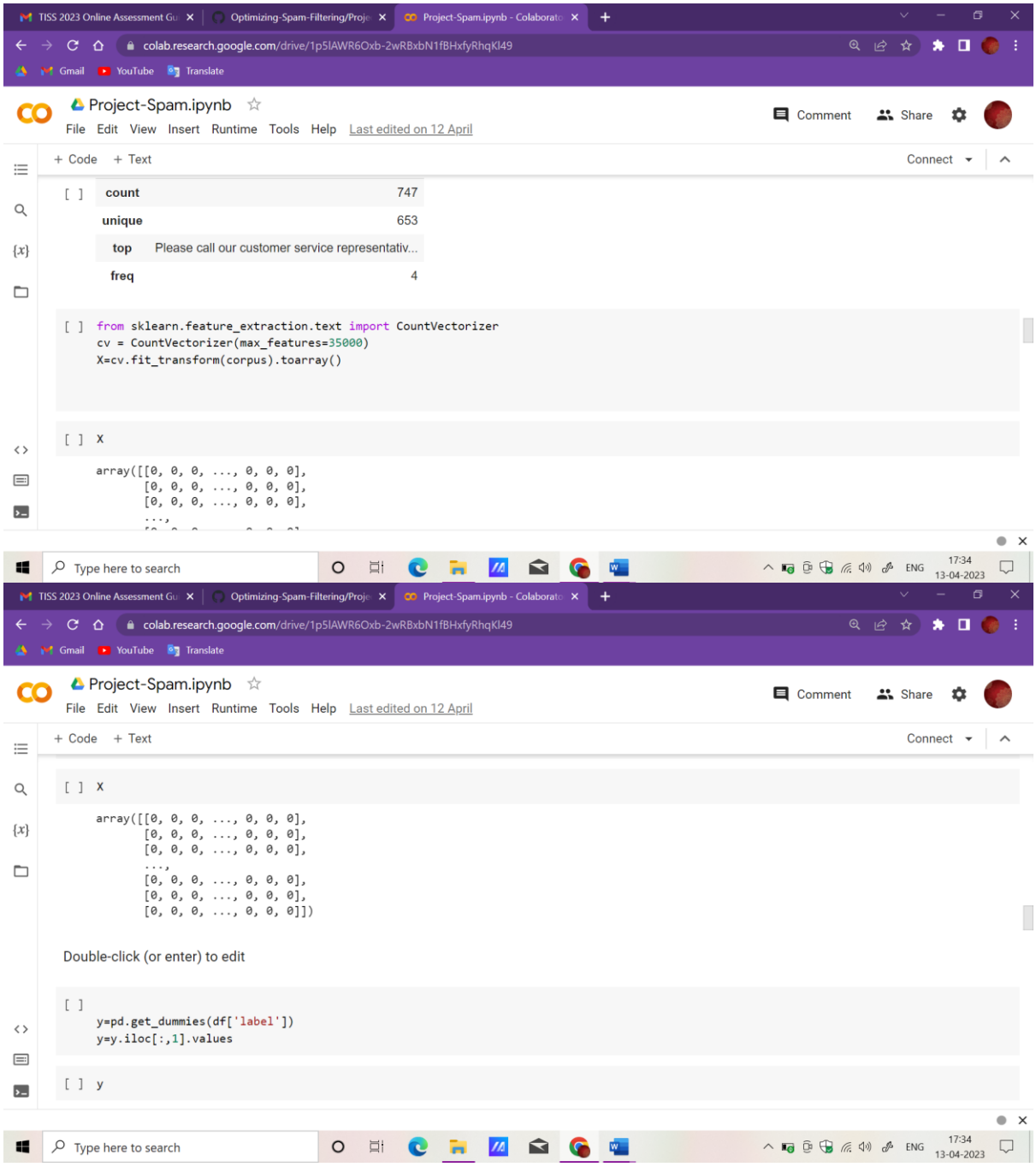
```
[ ] import nltk
    from nltk.corpus import stopwords
    from nltk.stem import PorterStemmer
```

```
[ ] import re
    corpus = []
```

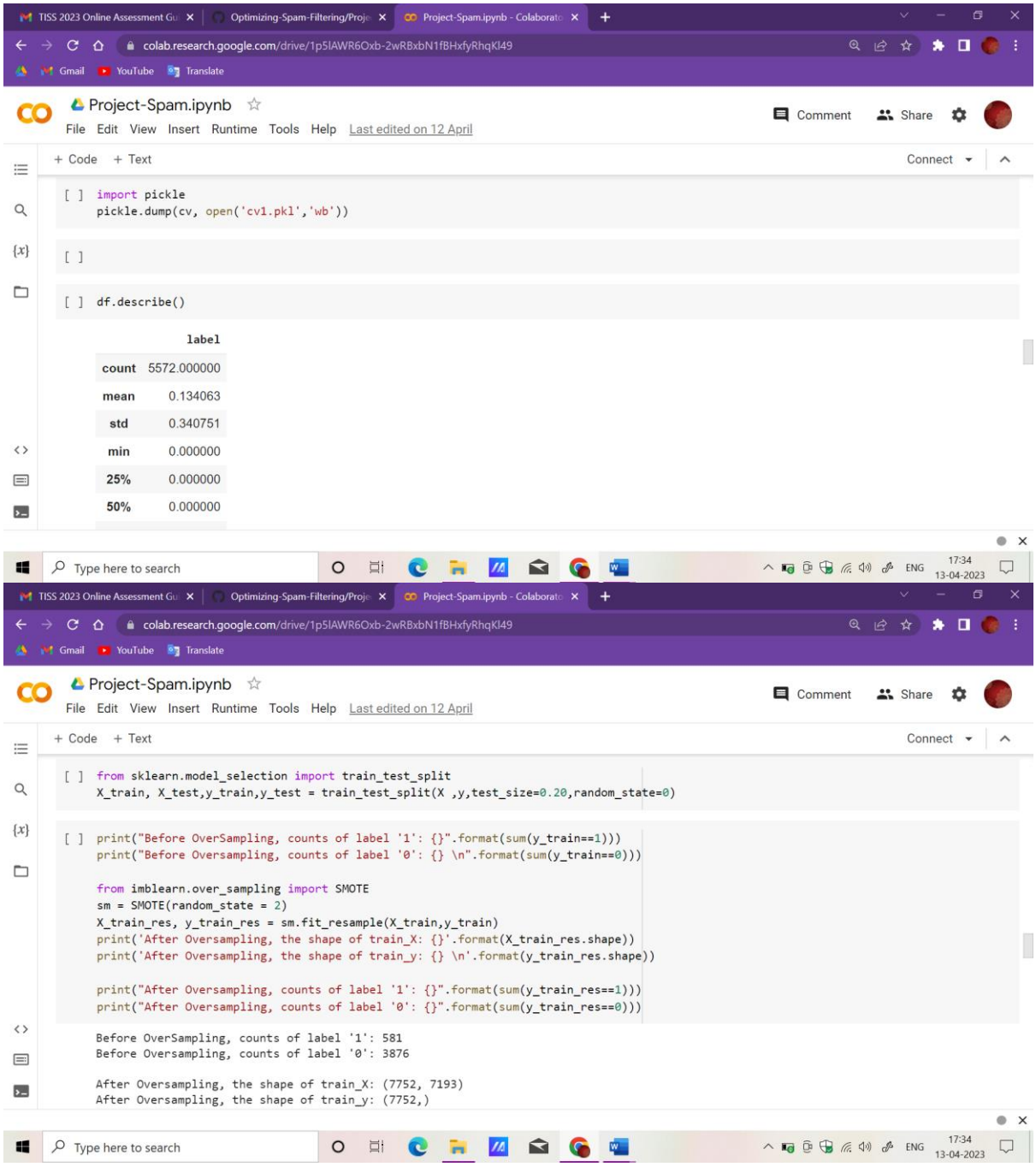
Type here to search ENG 17:34 13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] import re
corpus = []
length = len(df)
[ ] for i in range(0,length):
text = re.sub("[^a-zA-z0-9]"," ",df["text"][i])
text = text.lower()
text = text.split()
pe = PorterStemmer()
stopword = stopwords.words("english")
text = [pe.stem(word) for word in text if not word in set (stopword)]
text = " ".join(text)
corpus.append(text)
[ ] corpus
['go jurong point crazi avail bugi n great world la e buffet cine got amor wat',
'ok lar joke wif u oni',
'free entri 2 wkli comp win fa cup final tkt 21st may 2005 text fa 87121 receiv entri question std txt rate c appli
Type here to search
ENG
17:34
13-04-2023

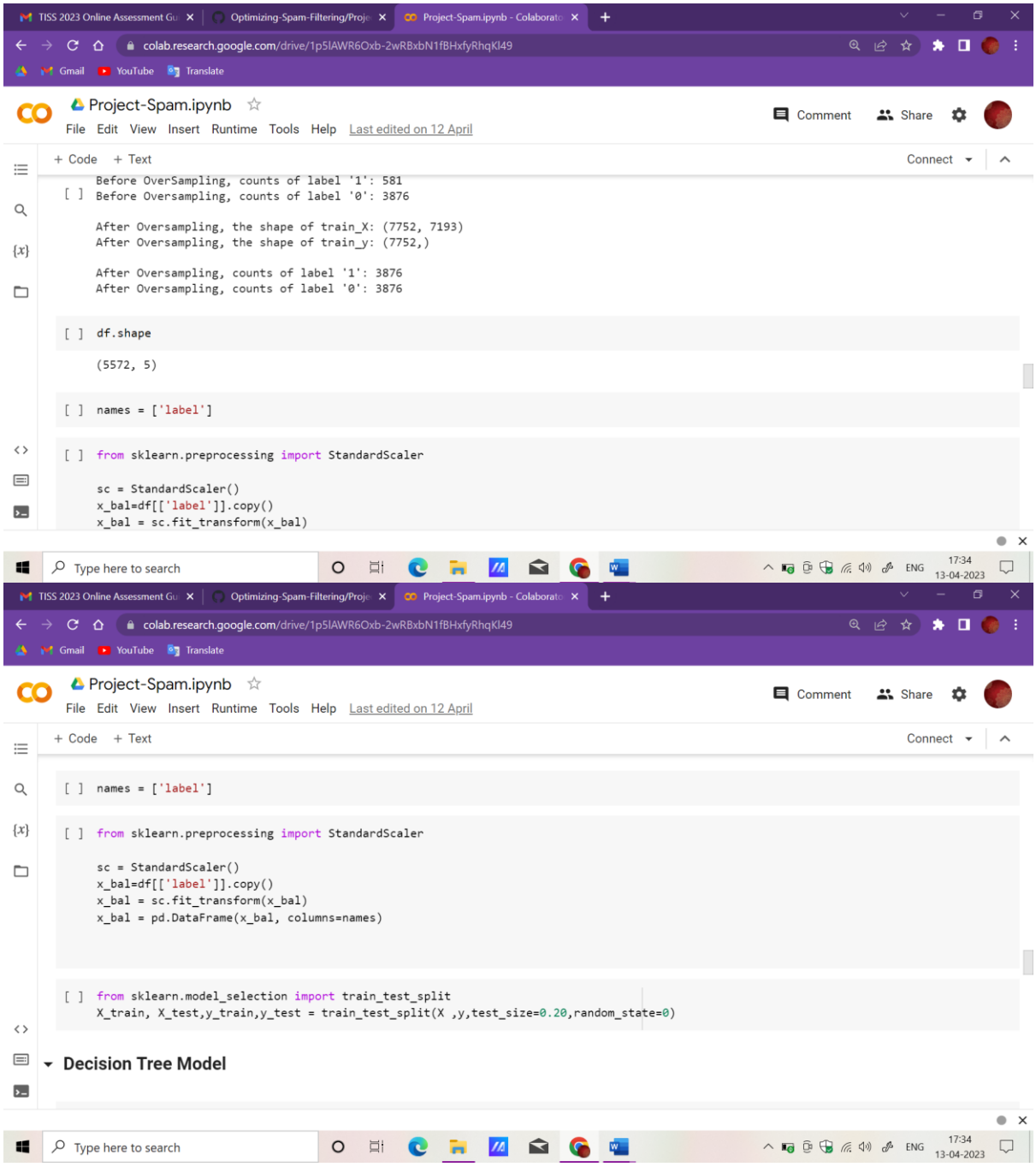
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] #ham
df[df['label']==0][['text',]].describe()
text
count 4825
unique 4516
top Sorry, I'll call later
freq 30
[ ] #spam
df[df['label']==1][['text',]].describe()
text
count 747
Type here to search
ENG
17:34
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
count 747
unique 653
top Please call our customer service representativ...
freq 4
from sklearn.feature_extraction.text import CountVectorizer
cv = CountVectorizer(max_features=35000)
X=cv.fit_transform(corpus).toarray()
X
array([[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
...,
Type here to search
ENG
17:34
13-04-2023

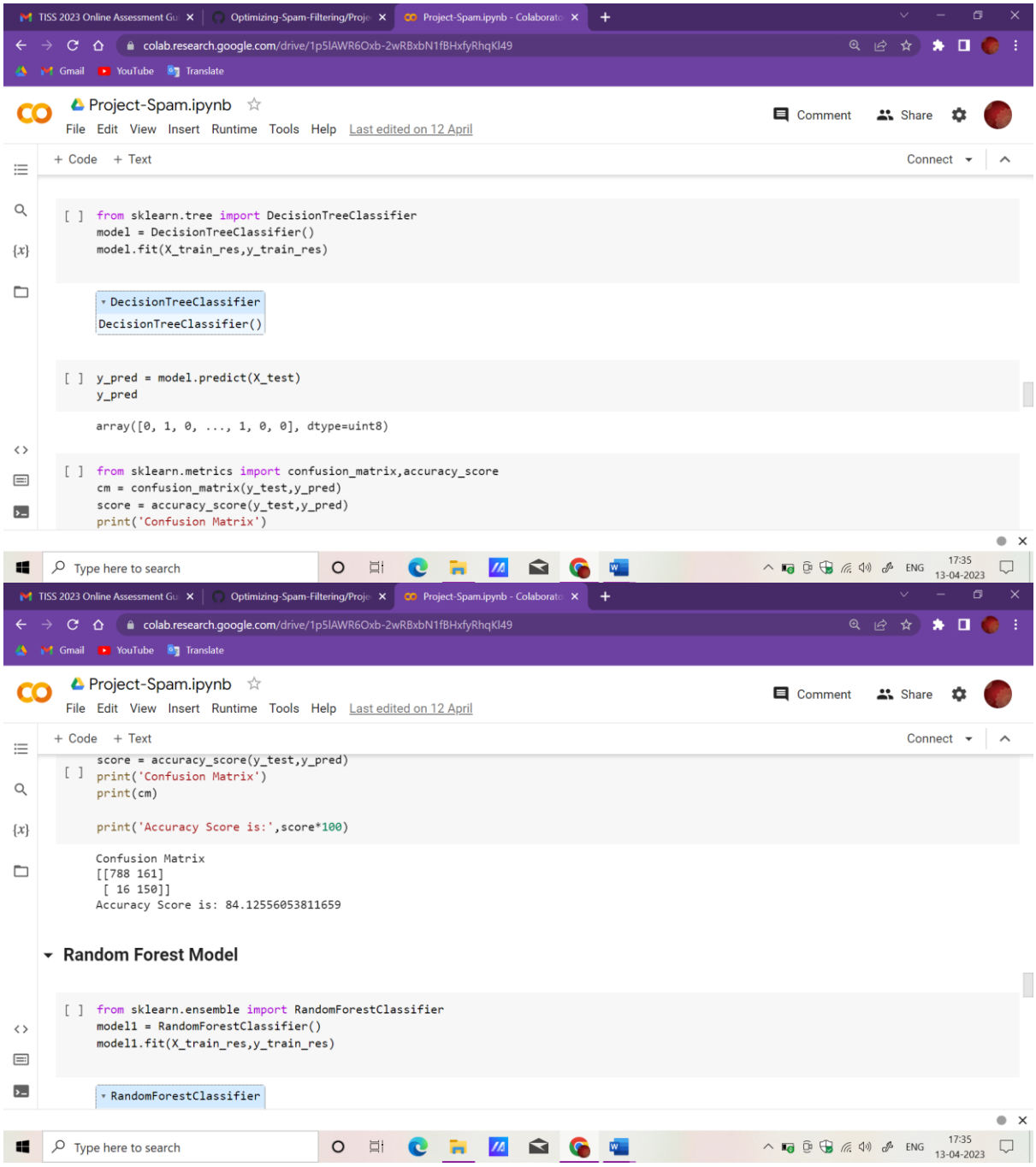
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
X
array([[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
...,
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0],
[0, 0, 0, ..., 0, 0, 0]])
Double-click (or enter) to edit
y=pd.get_dummies(df['label'])
y=y.iloc[:,1].values
y
Type here to search
ENG
17:34
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] import pickle
pickle.dump(cv, open('cv1.pkl','wb'))
[ ]
[ ] df.describe()
label
count 5572.000000
mean 0.134063
std 0.340751
min 0.000000
25% 0.000000
50% 0.000000
Type here to search
ENG
17:34
13-04-2023

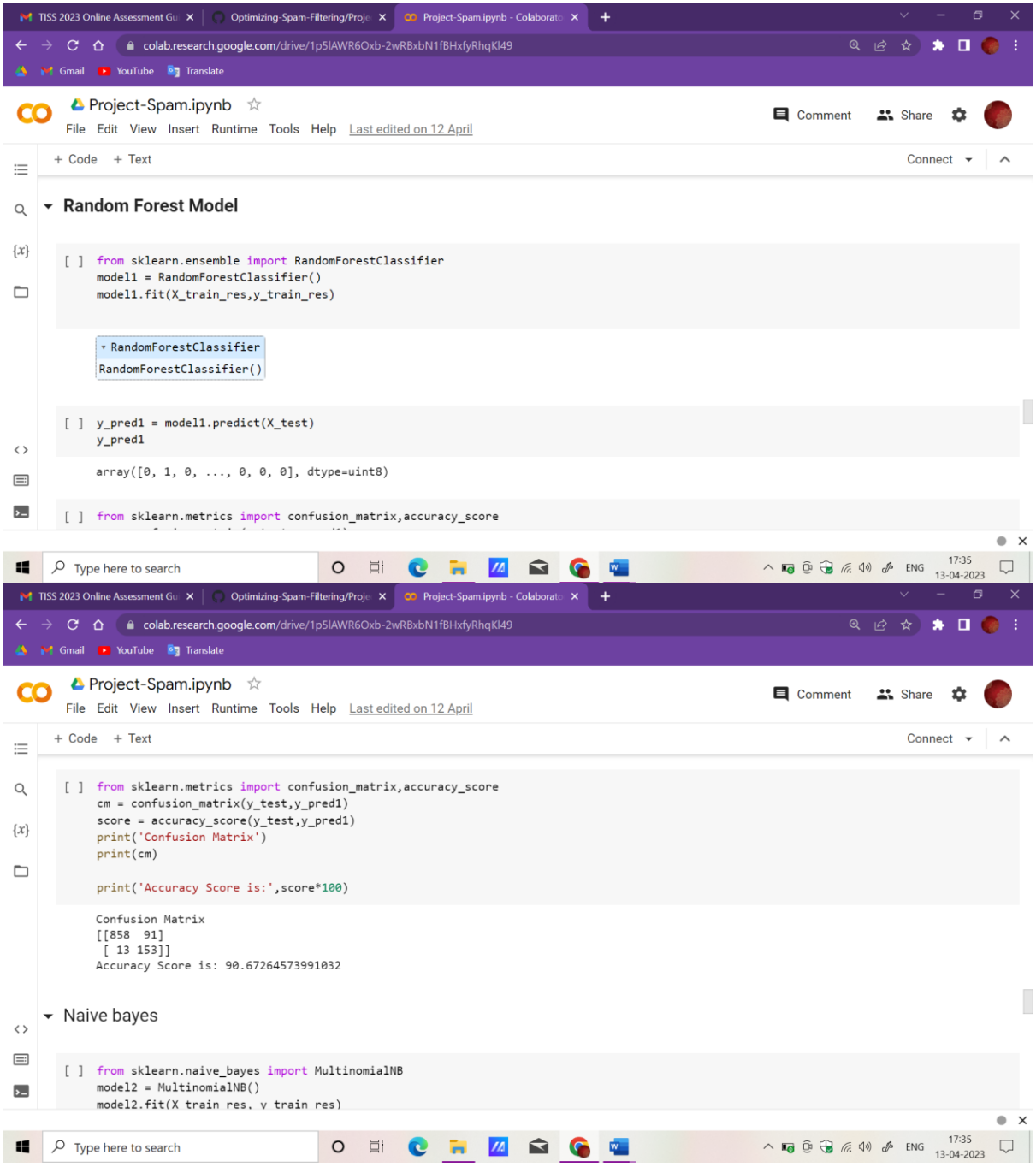
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test = train_test_split(X ,y,test_size=0.20,random_state=0)
[ ] print("Before OverSampling, counts of label '1': {}".format(sum(y_train==1)))
print("Before Oversampling, counts of label '0': {} \n".format(sum(y_train==0)))
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 2)
X_train_res, y_train_res = sm.fit_resample(X_train,y_train)
print('After Oversampling, the shape of train_X: {}'.format(X_train_res.shape))
print('After Oversampling, the shape of train_y: {} \n'.format(y_train_res.shape))
print("After Oversampling, counts of label '1': {}".format(sum(y_train_res==1)))
print("After Oversampling, counts of label '0': {}".format(sum(y_train_res==0)))
Before OverSampling, counts of label '1': 581
Before Oversampling, counts of label '0': 3876
After Oversampling, the shape of train_X: (7752, 7193)
After Oversampling, the shape of train_y: (7752,)
Type here to search
ENG
17:34
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
Before OverSampling, counts of label '1': 581
Before Oversampling, counts of label '0': 3876
After Oversampling, the shape of train_X: (7752, 7193)
After Oversampling, the shape of train_y: (7752,)
After Oversampling, counts of label '1': 3876
After Oversampling, counts of label '0': 3876
df.shape
(5572, 5)
names = ['label']
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_bal=df[['label']].copy()
x_bal = sc.fit_transform(x_bal)
Type here to search
ENG
17:34
13-04-2023

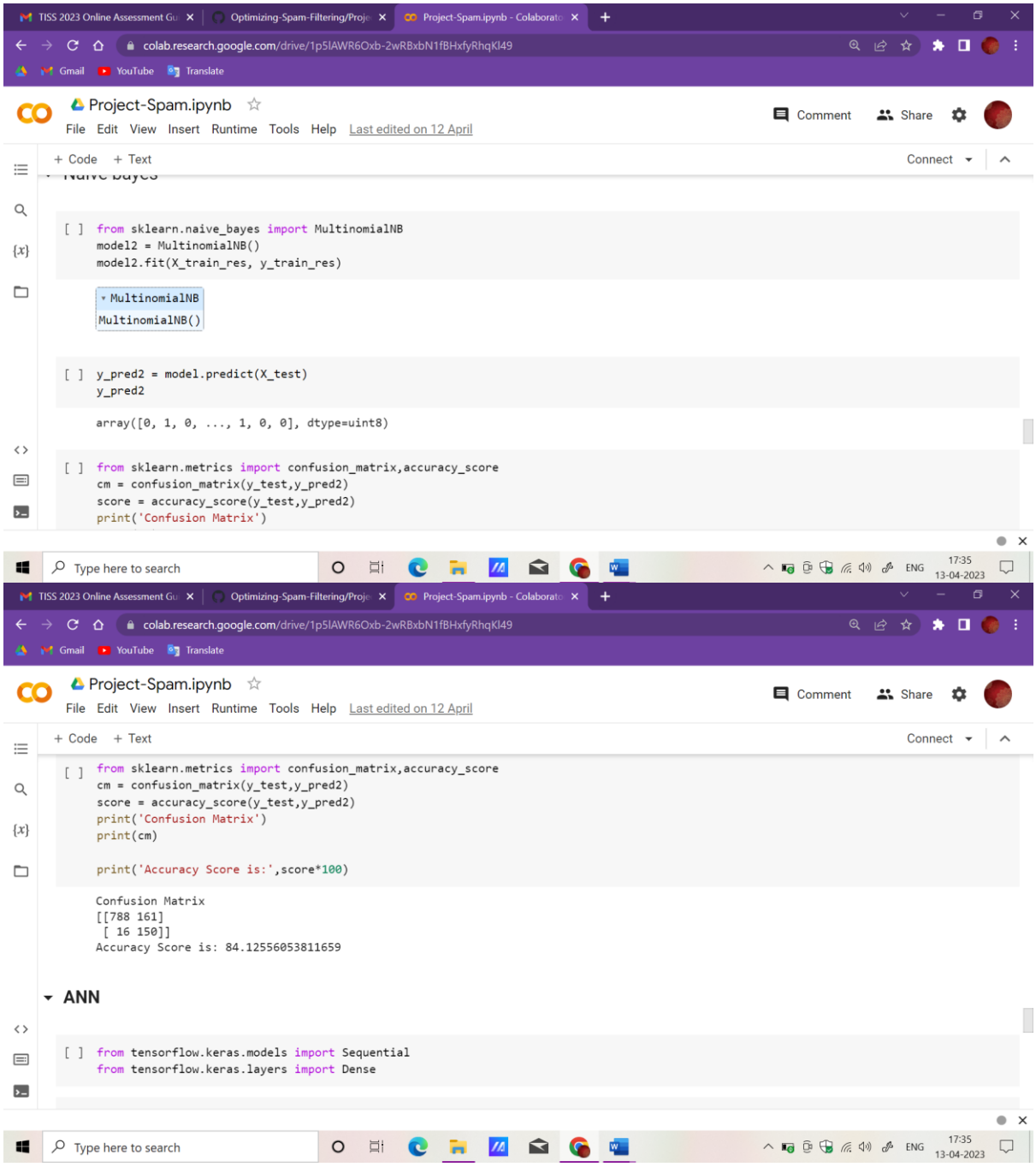
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
names = ['label']
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_bal=df[['label']].copy()
x_bal = sc.fit_transform(x_bal)
x_bal = pd.DataFrame(x_bal, columns=names)
from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test = train_test_split(X ,y,test_size=0.20,random_state=0)
Decision Tree Model
Type here to search
ENG
17:34
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] from sklearn.tree import DecisionTreeClassifier
model = DecisionTreeClassifier()
model.fit(X_train_res,y_train_res)
DecisionTreeClassifier
DecisionTreeClassifier()
[ ] y_pred = model.predict(X_test)
y_pred
array([0, 1, 0, ..., 1, 0, 0], dtype=uint8)
[ ] from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred)
score = accuracy_score(y_test,y_pred)
print('Confusion Matrix')
Type here to search
ENG
17:35
13-04-2023

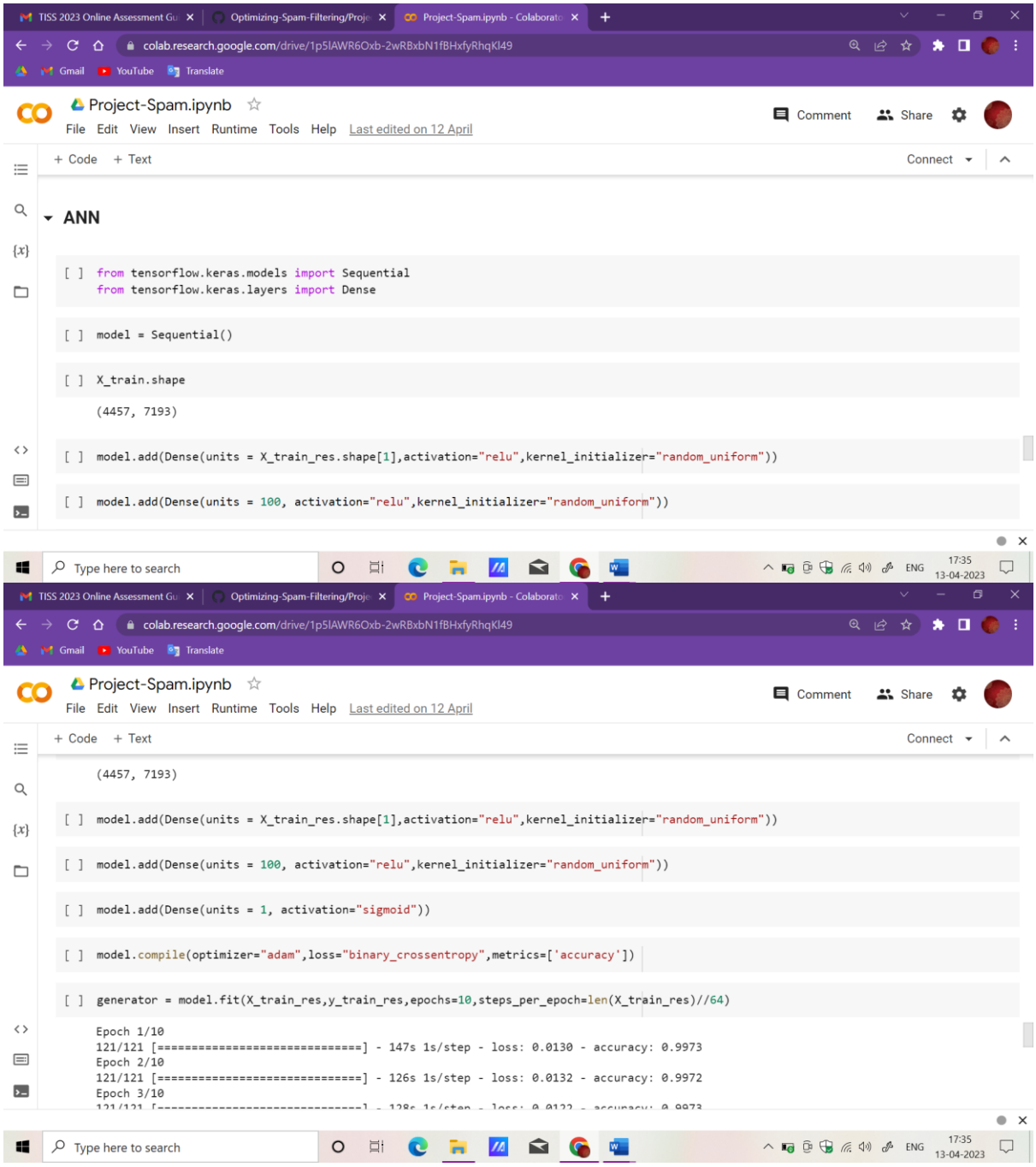
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
score = accuracy_score(y_test,y_pred)
[ ] print('Confusion Matrix')
print(cm)
print('Accuracy Score is:',score*100)
Confusion Matrix
[[788 161]
[16 150]]
Accuracy Score is: 84.12556053811659
Random Forest Model
[ ] from sklearn.ensemble import RandomForestClassifier
model1 = RandomForestClassifier()
model1.fit(X_train_res,y_train_res)
RandomForestClassifier
Type here to search
ENG
17:35
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
Random Forest Model
[ ] from sklearn.ensemble import RandomForestClassifier
model1 = RandomForestClassifier()
model1.fit(X_train_res,y_train_res)
RandomForestClassifier
RandomForestClassifier()
[ ] y_pred1 = model1.predict(X_test)
y_pred1
array([0, 1, 0, ..., 0, 0, 0], dtype=uint8)
[ ] from sklearn.metrics import confusion_matrix,accuracy_score
Type here to search
ENG
17:35
13-04-2023

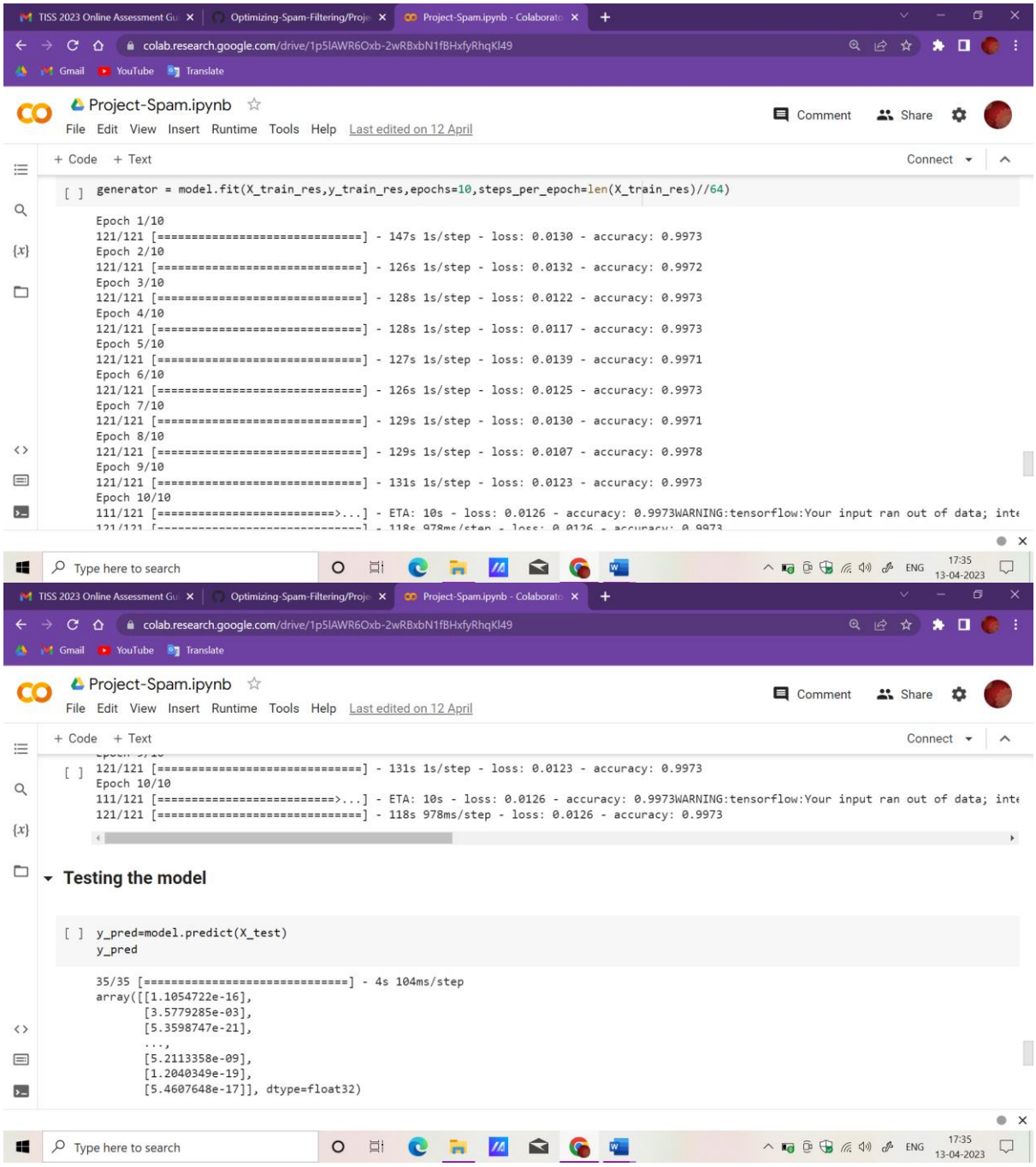
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKl49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred1)
score = accuracy_score(y_test,y_pred1)
print('Confusion Matrix')
print(cm)
print('Accuracy Score is:',score*100)
Confusion Matrix
[[858 91]
[13 153]]
Accuracy Score is: 90.67264573991032
Naive bayes
[ ] from sklearn.naive_bayes import MultinomialNB
model2 = MultinomialNB()
model2.fit(X_train_res, y_train_res)
Type here to search
ENG
17:35
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] from sklearn.naive_bayes import MultinomialNB
model2 = MultinomialNB()
model2.fit(X_train_res, y_train_res)
MultinomialNB
MultinomialNB()
[ ] y_pred2 = model.predict(X_test)
y_pred2
array([0, 1, 0, ..., 1, 0, 0], dtype=uint8)
[ ] from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred2)
score = accuracy_score(y_test,y_pred2)
print('Confusion Matrix')
Type here to search
ENG
17:35
13-04-2023

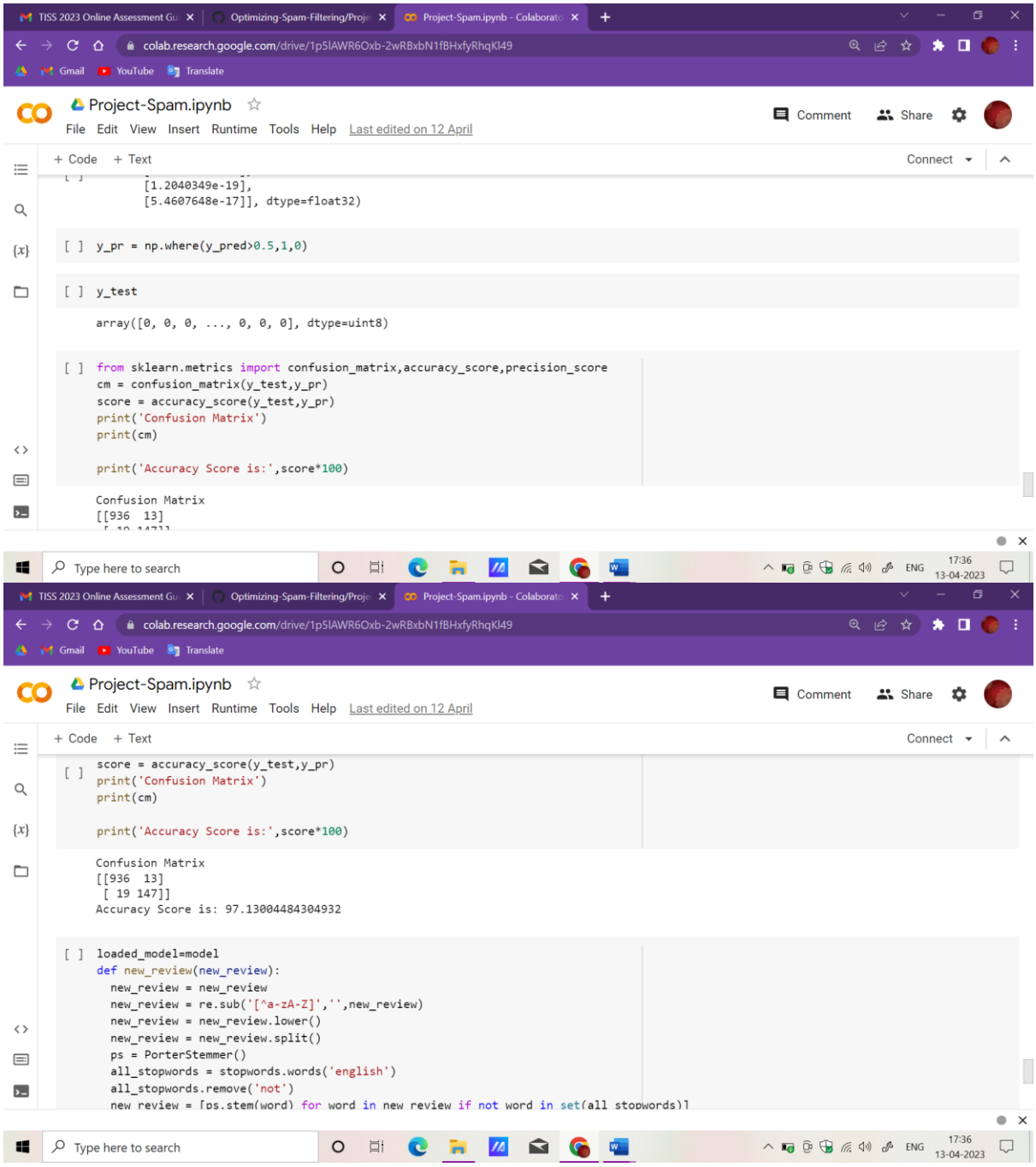
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] from sklearn.metrics import confusion_matrix,accuracy_score
cm = confusion_matrix(y_test,y_pred2)
score = accuracy_score(y_test,y_pred2)
print('Confusion Matrix')
print(cm)
print('Accuracy Score is:',score*100)
Confusion Matrix
[[788 161]
[16 150]]
Accuracy Score is: 84.12556053811659
ANN
[ ] from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
Type here to search
ENG
17:35
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
ANN
[ ] from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
[ ] model = Sequential()
[ ] X_train.shape
(4457, 7193)
[ ] model.add(Dense(units = X_train_res.shape[1],activation="relu",kernel_initializer="random_uniform"))
[ ] model.add(Dense(units = 100, activation="relu",kernel_initializer="random_uniform"))
Type here to search
ENG
17:35
13-04-2023

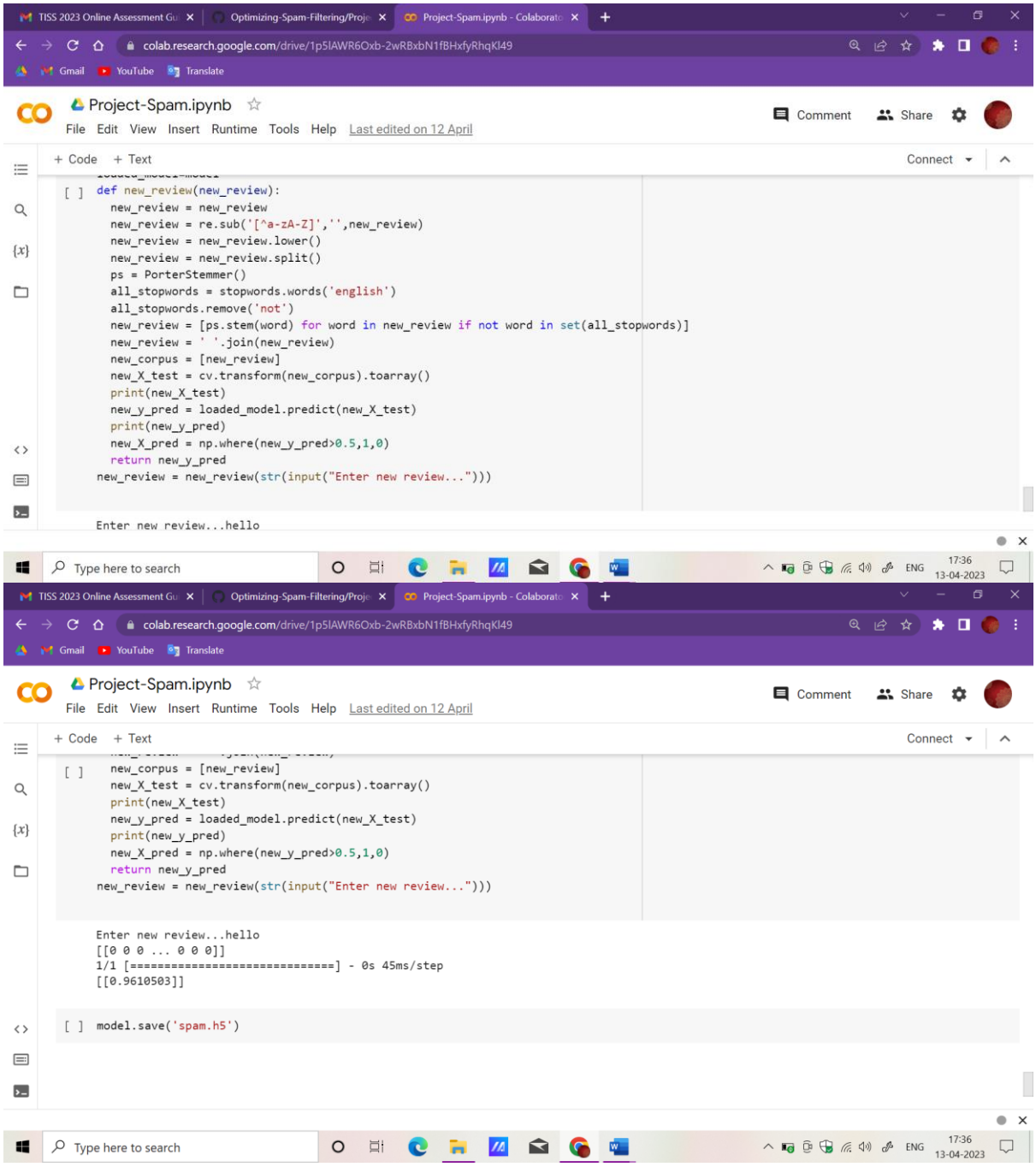
TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
(4457, 7193)
[ ] model.add(Dense(units = X_train_res.shape[1],activation="relu",kernel_initializer="random_uniform"))
[ ] model.add(Dense(units = 100, activation="relu",kernel_initializer="random_uniform"))
[ ] model.add(Dense(units = 1, activation="sigmoid"))
[ ] model.compile(optimizer="adam",loss="binary_crossentropy",metrics=['accuracy'])
[ ] generator = model.fit(X_train_res,y_train_res,epochs=10,steps_per_epoch=len(X_train_res)//64)
Epoch 1/10
121/121 [==============================] - 147s 1s/step - loss: 0.0130 - accuracy: 0.9973
Epoch 2/10
121/121 [==============================] - 126s 1s/step - loss: 0.0132 - accuracy: 0.9972
Epoch 3/10
Type here to search
ENG
17:35
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] generator = model.fit(X_train_res,y_train_res,epochs=10,steps_per_epoch=len(X_train_res)//64)
Epoch 1/10
121/121 [==============================] - 147s 1s/step - loss: 0.0130 - accuracy: 0.9973
Epoch 2/10
121/121 [==============================] - 126s 1s/step - loss: 0.0132 - accuracy: 0.9972
Epoch 3/10
121/121 [==============================] - 128s 1s/step - loss: 0.0122 - accuracy: 0.9973
Epoch 4/10
121/121 [==============================] - 128s 1s/step - loss: 0.0117 - accuracy: 0.9973
Epoch 5/10
121/121 [==============================] - 127s 1s/step - loss: 0.0139 - accuracy: 0.9971
Epoch 6/10
121/121 [==============================] - 126s 1s/step - loss: 0.0125 - accuracy: 0.9973
Epoch 7/10
121/121 [==============================] - 129s 1s/step - loss: 0.0130 - accuracy: 0.9971
Epoch 8/10
121/121 [==============================] - 129s 1s/step - loss: 0.0107 - accuracy: 0.9978
Epoch 9/10
121/121 [==============================] - 131s 1s/step - loss: 0.0123 - accuracy: 0.9973
Epoch 10/10
111/121 [==========================>...] - ETA: 10s - loss: 0.0126 - accuracy: 0.9973WARNING:tensorflow:Your input ran out of data; inte
Type here to search
ENG
17:35
13-04-2023

TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
+ Code + Text
Connect
[ ] 121/121 [==============================] - 131s 1s/step - loss: 0.0123 - accuracy: 0.9973
Epoch 10/10
111/121 [==========================>...] - ETA: 10s - loss: 0.0126 - accuracy: 0.9973WARNING:tensorflow:Your input ran out of data; inte
121/121 [==============================] - 118s 978ms/step - loss: 0.0126 - accuracy: 0.9973
Testing the model
[ ] y_pred=model.predict(X_test)
y_pred
35/35 [==============================] - 4s 104ms/step
array([[1.1054722e-16],
[3.5779285e-03],
[5.3598747e-21],
...,
[5.2113358e-09],
[1.2040349e-19],
[5.4607648e-17]], dtype=float32)
Type here to search
ENG
17:35
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[1.2040349e-19],
[5.4607648e-17]], dtype=float32)
[ ] y_pr = np.where(y_pred>0.5,1,0)
[ ] y_test
array([0, 0, 0, ..., 0, 0, 0], dtype=uint8)
[ ] from sklearn.metrics import confusion_matrix,accuracy_score,precision_score
cm = confusion_matrix(y_test,y_pr)
score = accuracy_score(y_test,y_pr)
print('Confusion Matrix')
print(cm)
print('Accuracy Score is:',score*100)
Confusion Matrix
[[936 13]
Type here to search
ENG
17:36
13-04-2023

TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] score = accuracy_score(y_test,y_pr)
print('Confusion Matrix')
print(cm)
print('Accuracy Score is:',score*100)
Confusion Matrix
[[936 13]
[19 147]]
Accuracy Score is: 97.13004484304932
[ ] loaded_model=model
def new_review(new_review):
new_review = new_review
new_review = re.sub('[^a-zA-Z]','',new_review)
new_review = new_review.lower()
new_review = new_review.split()
ps = PorterStemmer()
all_stopwords = stopwords.words('english')
all_stopwords.remove('not')
Type here to search
ENG
17:36
13-04-2023


TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] def new_review(new_review):
new_review = new_review
new_review = re.sub('[^a-zA-Z]',' ',new_review)
new_review = new_review.lower()
new_review = new_review.split()
ps = PorterStemmer()
all_stopwords = stopwords.words('english')
all_stopwords.remove('not')
new_review = [ps.stem(word) for word in new_review if not word in set(all_stopwords)]
new_review = ' '.join(new_review)
new_corpus = [new_review]
new_X_test = cv.transform(new_corpus).toarray()
print(new_X_test)
new_y_pred = loaded_model.predict(new_X_test)
print(new_y_pred)
new_X_pred = np.where(new_y_pred>0.5,1,0)
return new_y_pred
new_review = new_review(str(input("Enter new review...")))
Enter new review...hello
Type here to search
ENG
17:36
13-04-2023

TISS 2023 Online Assessment Gu
Optimizing-Spam-Filtering/Proje
Project-Spam.ipynb - Colaborato
colab.research.google.com/drive/1p5IAWR6Oxb-2wRBxbN1fBHxfyRhqKI49
Gmail
YouTube
Translate
Project-Spam.ipynb
File Edit View Insert Runtime Tools Help Last edited on 12 April
Comment
Share
Connect
+ Code + Text
[ ] new_corpus = [new_review]
new_X_test = cv.transform(new_corpus).toarray()
print(new_X_test)
new_y_pred = loaded_model.predict(new_X_test)
print(new_y_pred)
new_X_pred = np.where(new_y_pred>0.5,1,0)
return new_y_pred
new_review = new_review(str(input("Enter new review...")))
Enter new review...hello
[[0 0 0 ... 0 0 0]]
1/1 [==============================] - 0s 45ms/step
[[0.9610503]]
[ ] model.save('spam.h5')
Type here to search
ENG
17:36
13-04-2023