Assignment-16

Lohendra P
2406CYS124

Q) You are tasked with developing a Python code for sentiment extraction utilizing a provided sample dataset. The dataset consists of textual data annotated with labels categorizing sentiments into four categories: "rude," "normal," "insult," and "sarcasm." Dataset:

- Real News:

https://drive.google.com/file/d/1FL2HqgLDAP5550nd1h_8iBhAV

ISTnzr/view?usp=sharing

- Fake News:

https://drive.google.com/file/d/1EdI_HyUeI_Fi2nld7rQnnGEpQqn_BwM

/view? usp=sharing

1. Outline the key steps involved in developing a sentiment extraction Algorithm using Python.
2. Describe the structure and format of the sample dataset required for Sentiment extraction.
3. Implement the Python code to read and pre-process the sample dataset for Sentiment analysis. Ensure that the code correctly handles text data and Labels.
4. Discuss the process of classifying sentiments into the specified categories: "Rude," "normal," "insult," and "sarcasm." Explain any techniques or Algorithms employed for this classification task.
5. Evaluate the effectiveness of the sentiment extraction algorithm on the Provided sample dataset. Consider metrics such as accuracy, precision, Recall, and F1-score.
6. Propose potential enhancements or modifications to improve the Performance of the sentiment extraction algorithm. Justify your Recommendations.
7. Reflect on the ethical considerations associated with sentiment analysis,

<u>Assignment-16</u>

Lohendra P
2406CYS124

Particularly regarding privacy, bias, and potential misuse of extracted Sentiments.

8. Write a complete code for this assignment.

Developing a sentiment extraction algorithm using Python involves several key steps.

Certainly! Let me rephrase the provided text for you:

---

1. **Data Collection**:
   - Obtain a dataset with labeled sentiment data. This dataset should include various text samples (such as reviews, tweets, or news articles) along with corresponding sentiment labels (positive, negative, or neutral).
2. **Data Preprocessing**:
   - Clean and preprocess the text data:
     - Remove punctuation marks.
     - Convert text to lowercase.
     - Eliminate stopwords (common words like "the," "and," etc.).
     - Handle any other necessary text transformations.
3. **Feature Extraction**:
   - Convert preprocessed text data into numerical feature vectors:
     - Techniques include:
       - **Bag-of-Words**: Represent each document as a vector of word frequencies.
       - **TF-IDF (Term Frequency-Inverse Document Frequency)**: Reflects the importance of words in a document relative to the entire dataset.
       - **Word Embeddings (e.g., Word2Vec, GloVe)**: Dense vector representations of words.
4. **Model Selection**:
   - Choose an appropriate machine learning algorithm for sentiment analysis:
     - Options include:
       - **Logistic Regression**: Simple and interpretable.
       - **Naive Bayes**: Effective for text classification.
       - **Support Vector Machines (SVM)**: Good for linear separation.
       - **Deep Learning Models (RNN, CNN)**: Capture complex patterns.
5. **Model Training**:
   - Split your dataset into training and testing sets:
     - Train the model using the training set:
       - Fit it to the feature vectors and corresponding sentiment labels.
6. **Model Evaluation**:
   - Assess the performance of your sentiment extraction model:
     - Common evaluation metrics:
       - **Accuracy**: Overall correctness.
       - **Precision**: Proportion of true positives among predicted positives.
       - **Recall**: Proportion of true positives among actual positives.
       - **F1-score**: Harmonic mean of precision and recall.
7. **Fine-tuning and Optimization**:
   - Iterate on your model:
     - Fine-tune hyperparameters (e.g., learning rate, regularization strength).
     - Explore different feature extraction techniques or model architectures.
     - Experiment with data augmentation or ensemble methods.

<u>Assignment-16</u>

Lohendra P
2406CYS124

8. **Deployment**:
   - Once satisfied with performance, deploy the sentiment extraction algorithm:
     - Create APIs or integrate it into a larger application for real-world use.

The structure and format of a sample dataset required for sentiment extraction can vary, but it typically consists of two main components:

1. Text Data: The dataset should include a collection of text samples or documents on which sentiment analysis will be performed. Each text sample represents a piece of content (such as reviews, tweets, comments, or product descriptions) that expresses opinions or sentiments.

Assignment-16

Lohendra P
2406CYS124

Sentiment Labels: Along with the text data, the dataset should also include sentiment labels associated with each text sample. These labels indicate the sentiment expressed in the corresponding text, such as positive, negative, or neutral. Sometimes, sentiment labels are represented as numerical values (e.g., 0 for negative, 1 for neutral, and 2 for positive).

Here is an example of how the dataset might be organized in a tabular format:

| Text Data | Sentiment Label |

├-----------┼---------·|

| I loved the movie! | Positive |

| This book is boring.| Negative |

| The product is okay.| Neutral |

| Fantastic experience| Positive |

| Disappointed with the service | Negative |

In this sample dataset, each row represents a text sample, and the corresponding sentiment label indicates the sentiment expressed in the text. This structure allows the sentiment extraction algorithm to learn patterns and make predictions based on the text and sentiment relationship.

It's worth noting that datasets for sentiment extraction can vary in size, domain, and annotation quality. It is essential to ensure that the dataset is representative and sufficiently labeled to train an effective sentiment extraction algorithm.

```
import pandas as pd import

re

import nltk

from nltk.corpus import stopwords

from sklearn.model_selection import train_test_split


# Read the dataset into a pandas DataFrame

df = pd.read_csv('sample_dataset.csv')  # Replace 'sample_dataset.csv' with the actual file
name


# Preprocessing steps  def

preprocess_text(text):
```

Assignment-16

Lohendra P
2406CYS124

```
    # Remove special characters and numbers text =
    re.sub('[^a-zA-Z]', ' ', text)

    # Convert text to lowercase text =
    text.lower()

    # Tokenize the text
    tokens = nltk.word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))
    tokens = [token for token in tokens if token not in stop_words]

    # Join the tokens back into a single string
    preprocessed_text = ' '.join(tokens)

    return preprocessed_text

# Preprocess the text data
df['preprocessed_text'] = df['text'].apply(preprocess_text)

# Split the data into train and test sets train_data,
test_data, train_labels, test_labels =
train_test_split(df['preprocessed_text'], df['label'], test_size=0.2, random_state=42)

# Further processing or model training can be performed on the preprocessed data
```

Classifying Sentiment: Rude, Normal, Insult, and Sarcasm

Assignment-16

Lohendra P
2406CYS124

Sentiment analysis, also known as opinion mining, aims to understand the emotional tone behind text data. Classifying sentiment into specific categories like "rude," "normal," "insult," and "sarcasm" can be challenging due to the nuances of human language. Here's a breakdown of the process and techniques used:

1. Data Preprocessing:

- **Text Cleaning:** Removing noise like punctuation, stop words (common words like "the" or "a"), and converting text to lowercase is essential.
- **Lemmatization/Stemming:** Reducing words to their base form (e.g., "running" becomes "run") improves consistency.

2. Feature Engineering:

- **Lexicon-based Approach:** Words are assigned sentiment scores based on pre-built sentiment lexicons (lists of words with positive, negative, or neutral sentiment).
- **N-grams:** Analyzing sequences of words (bigrams, trigrams) can capture context. "Great job" is positive, but "big mistake" is negative.

3. Machine Learning Models:

- **Supervised Learning:**
  - Training data with labeled examples (e.g., a sentence marked as "rude") is fed to models like Support Vector Machines (SVMs) or Naive Bayes.
  - The model learns to identify patterns associated with each sentiment category.
- **Deep Learning:** Advanced techniques like Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks can analyze the sequence of words and context more effectively, especially for sarcasm detection.

Challenges of Classifying Specific Categories:

- **Subjectivity:** "Rude" can be subjective. "That was a bold choice" might be rude depending on context.
- **Sarcasm:** Identifying sarcasm requires understanding the context and often relies on nonverbal cues like tone of voice, which text lacks.
  - Techniques like identifying inconsistencies between the literal meaning and the sentiment expressed, or the use of exclamation points (!) and question marks (?) can help.

Additional Techniques:

- **Emojis and Sentiment Analysis:** Emojis can convey strong sentiment. Sentiment lexicons can be expanded to include emojis with positive or negative connotations.
- **Hybrid Approaches:** Combining lexicon-based methods with machine learning can improve accuracy.

Assignment-16

Lohendra P
2406CYS124

Overall, sentiment classification is an evolving field. While models can achieve good accuracy for basic sentiment (positive, negative, neutral), identifying nuances like rudeness and sarcasm requires ongoing development and consideration of context.

Evaluating Sentiment Extraction Algorithm with "Rude," "Normal," "Insult," and "Sarcasm" Labels

Here's how to evaluate the effectiveness of the sentiment extraction algorithm on your dataset:

Metrics:

- **Accuracy:** Overall percentage of correctly classified samples across all categories ("rude," "normal," "insult," and "sarcasm").
- **Precision:** For each sentiment category, the proportion of samples the algorithm classified as that category that actually belong to that category (avoiding false positives).
- **Recall:** For each sentiment category, the proportion of samples that actually belong to that category that the algorithm correctly classified (avoiding false negatives).
- **F1-score:** Harmonic mean of precision and recall, combining both metrics into a single score.

Evaluation Process:

1. **Split the dataset:** Divide your data into a training set (used to train the algorithm) and a testing set (used to evaluate its performance).
2. **Train the model:** Train your sentiment extraction algorithm on the training set.
3. **Evaluate on the testing set:** Make predictions on the testing set using the trained model.
4. **Calculate evaluation metrics:** Using the ground truth labels (actual sentiment) and the model's predictions on the testing set, calculate accuracy, precision, recall, and F1-score for each category ("rude," "normal," "insult," and "sarcasm").

Challenges:

- **Balanced Dataset:** The effectiveness of these metrics depends on a balanced dataset. If most data belongs to the "normal" category, the model might achieve high overall accuracy but struggle with less frequent categories like "insult" or "sarcasm." Analyze precision and recall for each category to identify potential weaknesses.
- **Class Imbalance Techniques:** If the dataset is imbalanced, consider using techniques like oversampling (replicating data from the minority class) or under sampling (removing data from the majority class) to create a more balanced training set.

Assignment-16

Lohendra P
2406CYS124

Interpretation:

- A high accuracy score indicates the model performs well overall.
- High precision for a category like "insult" means the model rarely misclassifies other types of text as insults (reducing false positives).
- High recall for "sarcasm" means the model identifies most sarcastic comments (reducing false negatives).
- F1-score provides a balanced view of precision and recall.

Additional Considerations:

- **Error Analysis:** Analyse the types of errors the model makes to understand its weaknesses. Are there specific types of sarcasm it struggles with? Does it misclassify neutral comments as rude?
- **Visualization Techniques:** Consider using confusion matrices to visualize how the model performed on each category classification.

By evaluating sentiment extraction algorithm using these metrics and considering the challenges, you can gain valuable insights into its effectiveness for classifying "rude," "normal," "insult," and "sarcasm" sentiments in your specific dataset.

To improve the performance of the sentiment extraction algorithm, we can consider the following potential enhancements or modifications:

1. Integration of Domain-specific Language Models:
Incorporating domain-specific language models such as specialized sentiment lexicons or dictionaries can enhance the algorithm's understanding of industry- specific language nuances and sentiment expressions. By integrating domain- specific knowledge, the algorithm can more accurately classify sentiments within the context of the target domain.

2. Fine-tuning Pretrained Language Models:
Fine-tuning pretrained language models like BERT, RoBERTa, or ALBERT on domain-specific datasets can improve the algorithm's performance by adapting to the specific sentiment patterns and vocabulary of the target domain. Fine-tuning allows the model to capture domain-specific sentiment nuances and context, leading to more accurate sentiment extraction.

3. Data Augmentation Techniques:
Augmenting the training data through techniques like back translation, synonym replacement, or data synthesis can increase the diversity and quantity of training examples. By exposing the algorithm to a wider range of sentiment expressions, data augmentation can improve the model's ability to generalize and accurately classify sentiments in real-world text.

4. Ensemble Learning:
Implementing ensemble learning techniques such as bagging, boosting, or model stacking can enhance the robustness and generalization capability of the sentiment extraction algorithm. By combining multiple sentiment classifiers or models,

Lohendra P
2406CYS124

ensemble methods can mitigate individual model biases and errors, leading to improved sentiment classification performance.

5. Attention Mechanisms:
Leveraging attention mechanisms in neural network architectures can allow the algorithm to focus on critical words or phrases that contribute most to sentiment classification decisions. Attention mechanisms help the model capture important sentiment-bearing tokens and dependencies, improving the interpretability and performance of sentiment extraction.

6. Multi-task Learning:
Employing multi-task learning by training the sentiment extraction model on related tasks such as sentiment intensity prediction or aspect-based sentiment analysis can lead to a more holistic understanding of text sentiment. By jointly optimizing multiple sentiment-related objectives, the algorithm can capture nuanced sentiment information and improve overall sentiment classification accuracy.

7. Active Learning:
Implementing active learning strategies to iteratively select and label the most informative data points can enhance the efficiency and effectiveness of sentiment extraction model training. By prioritizing the annotation of crucial data samples, active learning can facilitate the algorithm's learning process and improve sentiment classification performance with limited labeled data.

By incorporating these enhancements and modifications, we can enhance the sentiment extraction algorithm's performance by leveraging domain-specific knowledge, fine-tuning models, augmenting data, utilizing ensemble methods, attention mechanisms, multi-task learning, and active learning techniques. These strategies can collectively improve the algorithm's accuracy, robustness, and generalization capability in sentiment analysis tasks.

Sentiment analysis, like any other AI technology, raises important ethical considerations that need to be carefully addressed. **Let's reflect on the key ethical considerations associated with sentiment analysis:**

1. Privacy: Sentiment analysis often requires access to large amounts of personal data, which can include sensitive information. It is crucial to respect individuals' privacy rights by obtaining informed consent, anonymizing data, and ensuring secure storage and transmission of data. Transparent privacy policies and adherence to data protection regulations are imperative to maintain trust.

2. Bias: Bias in sentiment analysis can arise from various sources, such as biased training data, algorithmic design, or societal prejudices. Biased sentiment analysis systems may perpetuate discrimination, reinforce stereotypes, or produce unfair outcomes. Regular auditing and diverse representation in the development of sentiment analysis models can help mitigate bias and ensure more equitable results.

Assignment-16

Lohendra P
2406CYS124

3. Data source representation: Sentiment analysis models heavily rely on training data. If the training data is unrepresentative or lacks diversity, the model may fail to capture sentiments from different demographic groups, cultural backgrounds, or languages. Efforts should focus on collecting diverse and inclusive datasets that accurately represent the intended user base.

4. Transparency and explainability: The opacity of sentiment analysis algorithms can lead to concerns about accountability and fairness. Organizations should strive for transparency in disclosing the methodology, training data sources, and limitations of sentiment analysis systems. Providing explanations for the sentiment predictions can help users understand and evaluate the validity of the results.

5. Misuse of extracted sentiments: Sentiment analysis can have unintended consequences if the extracted sentiments are misused. It is essential to use sentiment analysis responsibly and ethically, respecting the privacy and well-being of individuals. Safeguards should be in place to prevent the misuse of sentiment analysis for purposes such as manipulating public opinion, fueling discrimination, or infringing on people's rights.

Addressing these ethical considerations requires collaborative efforts between developers, researchers, policymakers, and the wider community. Striving for transparency, fairness, inclusivity, and ongoing monitoring of sentiment analysis systems can help mitigate potential ethical risks and ensure that sentiment analysis is used in a responsible and beneficial manner.

Coding part for the given files:

```
1)pip install nltk scikit-learn pandas pip install nltk scikit-learn pandas

# Import all the Libraries

import nltk

import pandas as pd

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.svm import SVC

from sklearn.metrics import classification_report

nltk.download("punkt") ##Fucntion: Smilies
nltk.download("stopwords") ##Function: Build Customed Stopwords:: Specific to Domain
```

Assignment-16

Lohendra P
2406CYS124

```
from google.colab import drive


drive.mount("/content/drive", force_remount=True)

file_path="/content/drive"+"/My Drive/"+ "JNTUSessions/Fake.csv"
data = pd.read_csv(file_path)
data.head()

df = pd.DataFrame(data) ## etl():: AWS, Facebook, X, GCP
df
```

	title	text	subject	date
0	Donald Trump Sends Out Embarrassing New Year'...	Donald Trump just couldn t wish all Americans ...	News	December 31, 2017
1	Drunk Bragging Trump Staffer Started Russian ...	House Intelligence Committee Chairman Devin Nu...	News	December 31, 2017
2	Sheriff David Clarke Becomes An Internet Joke...	On Friday, it was revealed that former Milwauk...	News	December 30, 2017
3	Trump Is So Obsessed He Even Has Obama's Name...	On Christmas day, Donald Trump announced that ...	News	December 29, 2017
4	Pope Francis Just Called Out Donald Trump Dur...	Pope Francis used his annual Christmas Day mes...	News	December 25, 2017
...	...	...	...	...
23476	McPain: John McCain Furious That Iran Treated ...	21st Century Wire says As 21WIRE reported earl...	Middle-east	January 16, 2016
23477	JUSTICE? Yahoo Settles E-mail Privacy Class-ac...	21st Century Wire says It s a familiar theme. ...	Middle-east	January 16, 2016
23478	Sunnistan: US and Allied 'Safe Zone' Plan to T...	Patrick Henningsen 21st Century WireRemember ...	Middle-east	January 15, 2016

Assignment-16

Lohendra P
2406CYS124

	title	text	subject	date
23479	How to Blow $700 Million: Al Jazeera America F...	21st Century Wire says Al Jazeera America will...	Middle-east	January 14, 2016
23480	10 U.S. Navy Sailors Held by Iranian Military ...	21st Century Wire says As 21WIRE predicted in ...	Middle-east	January 12, 2016

23481 rows × 4 columns

```
stopwords = set(nltk.corpus.stopwords.words('english')) ## Translator APIs->Oxform NLP, Google Translator API
stemmer = nltk.stem.PorterStemmer()


def preprocess_text(text):
   tokens = nltk.word_tokenize(text.lower())
   tokens = [stemmer.stem(token) for token in tokens if token.isalnum() and token not in stopwords]
   return ' '.join(tokens)


df['processed_text'] = df['text'].apply(preprocess_text)

tfidf_vectorizer = TfidfVectorizer()
X = tfidf_vectorizer.fit_transform(df['processed_text'])
X


svm_classifier = SVC(kernel='linear') ## Non-Linear, ReLu, Leaky ReLu, Logistic
svm_classifier.fit(X, df['label'])

predictions = svm_classifier.predict(X)
predictions

df['predicted_level']=predictions
```

svm_classifier = SVC(kernel='linear') ## Non-Linear, ReLu, Leaky ReLu, Logistic

svm_classifier.fit(X, df['label'])

predictions = svm_classifier.predict(X) predictions

Assignment-16

Lohendra P
2406CYS124

df['predicted_level']=predictions

```
2)import pandas as pd import
numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder from keras
import Sequential
from keras.layers import Embedding, Dense, LSTM from
keras.preprocessing.text import one_hot  from keras.utils
import pad_sequences

import nltk
from nltk.stem.snowball import SnowballStemmer import
regex as re
from nltk.tokenize import sent_tokenize
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.model_selection import train_test_split import
warnings

warnings.filterwarnings('ignore') from
nltk.corpus import stopwords

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
stop_words = stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]    Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]    Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
```

addCode
addText

```
from google.colab import drive

drive.mount("/content/drive", force_remount=True)

drive.mount("/content/drive", force_remount=True)
```

Mounted at /content/drive

```
file_path = "/content/drive" + "/My Drive/" + "JNTUSessions/"
file_path = "/content/drive/My Drive/JNTUSessions/"
df_fake = pd.read_csv(file_path + "Fake.csv")
file_path = "/content/drive/My Drive/JNTUSessions/"
df_true = pd.read_csv(file_path + "True .csv")
df_fake.head()
```

Assignment-16

Lohendra P
2406CYS124

title text subject date

0 Donald Trump Sends Out Embarrassing New Year’... Donald Trump just couldn t wish all Americans ... News December 31, 2017

1 Drunk Bragging Trump Staffer Started Russian ... House Intelligence Committee Chairman Devin Nu... News December 31, 2017

2 Sheriff David Clarke Becomes An Internet Joke... On Friday, it was revealed that former Milwauk... News December 30, 2017

3 Trump Is So Obsessed He Even Has Obama’s Name... On Christmas day, Donald Trump announced that ... News December 29, 2017

4 Pope Francis Just Called Out Donald Trump Dur... Pope Francis used his annual Christmas Day mes... News December 25, 2017

```
df_true.head()
```

title text subject date

0 As U.S. budget fight looms, Republicans flip t... WASHINGTON (Reuters) - The head of a conservat... politicsNews December 31, 2017

1 U.S. military to accept transgender recruits o... WASHINGTON (Reuters) - Transgender people will... politicsNews December 29, 2017

2 Senior U.S. Republican senator: 'Let Mr. Muell... WASHINGTON (Reuters) - The special counsel inv... politicsNews December 31, 2017

3 FBI Russia probe helped by Australian diplomat... WASHINGTON (Reuters) - Trump campaign adviser ... politicsNews December 30, 2017

4 Trump wants Postal Service to charge 'much mor... SEATTLE/WASHINGTON (Reuters) - President Donal... politicsNews December 29, 2017

```
df_fake['status']=1
df_true['status']=0
df=pd.concat([df_true,df_fake])
df.drop(['subject', 'text', 'date'], axis=1, inplace=True)
def logest_sentence_length(text):
  return len(text.split())

ramdom_idexes=np.random.randint(0, len(df), len(df))
df = df.iloc[ramdom_idexes].reset_index(drop=True)

pd.set_option('display.max_colwidth', 500)
random = np.random.randint(0, len(df), 20)
df.iloc[random]

df.isnull().sum()
```

Assignment-16

Lohendra P
2406CYS124

```
df['maximum_length']=df['title'].apply(lambda x: logest_sentence_length(x)) max_length
= max(df['maximum_length'].values)
max_length

text_cleaning = "\b0\S*|\b[^A-Za-z0-9]+"

def preprocess_filter(text, stem=False):
  text = re.sub(text_cleaning, " ", str(text.lower()).strip()) tokens = []
  for token in text.split():
    if token not in stop_words:
      if stem:
        stemmer = SnowballStemmer(language='english') token
      = stemmer.stem(token) tokens.append(token)
  return " ".join(tokens)

def one_hot_encoded (text, vocab_size=5000, max_length=40): hot_encodeded =
  one_hot(text, vocab_size)
  return hot_encodeded

def word_embedding(text):
  preprocessed_text=preprocess_filter(text)
  hot_encoded=one_hot_encoded(preprocessed_text) return
  hot_encoded

embedded_features = 40
model = Sequential()
model.add(Embedding(5000,embedded_features,input_length=max_length))
model.add(LSTM(100))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss = 'binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
```

```
Model: "sequential"

____________________________________________________________________
 Layer (type)                Output Shape              Param #

===================================================================
=

 embedding (Embedding)       (None, 42, 40)            200000
```

Assignment-16

Lohendra P
2406CYS124

lstm (LSTM) (None, 100) 56400

dense (Dense) (None, 1) 101

===
=

Total params: 256501 (1001.96 KB)

Trainable params: 256501 (1001.96 KB)

Non-trainable params: 0 (0.00 Byte)

---

[36]

5m

one_hot_encoded_title=df['title'].apply(lambda x : word_embedding(x)).values

padded_encoded_title = pad_sequences(one_hot_encoded_title, maxlen=max_length,padding = "pre")

X = padded_encoded_title Y =

df['status'].values

Y = np.array(Y)

X.shape

Y.shape

X_train, X_test, Y_train, Y_test=train_test_split(X, Y, random_state=42)

model.fit(X_train,Y_train,validation_data=(X_test, Y_test), epochs=5, batch_size=64)

def best_threshold_value(thresholds:list, X_test): accuracies

= []

Lohendra P
2406CYS124

for thresh in thresholds: ypred=model.predict(X_test)

ypred = np.where(ypred > thresh,1,0)

accuracies.append(accuracy_score(Y_test, ypred)) return

pd.DataFrame({

'Threshold' : thresholds,

'Accuracy' : accuracies

})

best_threshold_value([0.4, 0.5, 0.6, 0.7, 0.8, 0.9], X_test)

Y_pred=model.predict(X_test)

Y_pred=np.where(Y_pred > 0.5, 1, 0)

confusion_matrix(Y_pred, Y_test)

classification_report(Y_pred, Y_test)

Epoch 1/5
527/527 [==============================] - 50s 89ms/step - loss: 0.2088 - accuracy: 0.9156 - val_loss: 0.1258 - val_accuracy: 0.9521 Epoch 2/5
527/527 [==============================] - 42s 79ms/step - loss: 0.0831 - accuracy: 0.9696 - val_loss: 0.1130 - val_accuracy: 0.9604 Epoch 3/5
527/527 [==============================] - 42s 80ms/step - loss: 0.0483 - accuracy: 0.9836 - val_loss: 0.1121 - val_accuracy: 0.9620 Epoch 4/5
527/527 [==============================] - 43s 81ms/step - loss: 0.0282 - accuracy: 0.9909 - val_loss: 0.1317 - val_accuracy: 0.9653 Epoch 5/5
527/527 [==============================] - 41s 77ms/step - loss: 0.0190 - accuracy: 0.9938 - val_loss: 0.1448 - val_accuracy: 0.9662

Assignment-16

Lohendra P
2406CYS124

351/351 [==============================] - 6s 15ms/step
351/351 [==============================] - 5s 13ms/step
351/351 [==============================] - 5s 15ms/step
351/351 [==============================] - 4s 12ms/step
351/351 [==============================] - 5s 13ms/step
351/351 [==============================] - 5s 15ms/step
351/351 [==============================] - 5s 13ms/step '
precision recall f1-score support\n\n 0 0.97 0.96 0.96 5403\n 1 0.97 0.97 0.97 5822\n\n accuracy 0.97 11225\n macro avg 0.97 0.97 0.97 11225\nweighted avg 0.97 0.97 0.97 11225\n

```
def prediction_input_processing(text):
  encoded=word_embedding(text)
  padded_encoded_title=pad_sequences([encoded], maxlen=max_length,
padding='pre')
  output=model.predict(padded_encoded_title)
  output=np.where(0.5>output,1,0)
  if output[0][0] == 1:
    return 'The News is Fake'
  return 'The News is True'

news_str1="Americans are more concerned over Indian fake open source
contributions"
news_str2="Trump Just Sent Michelle Obama a Bill She will Never Be able to
pay in her lifetime"
news_str3="Donald Trump Sends Out Embarrassing New Year's Eve Message"
prediction_input_processing(news_str3)
```

1/1 [==============================] - 0s 51ms/step '
The News is True