Model Development Phase Template

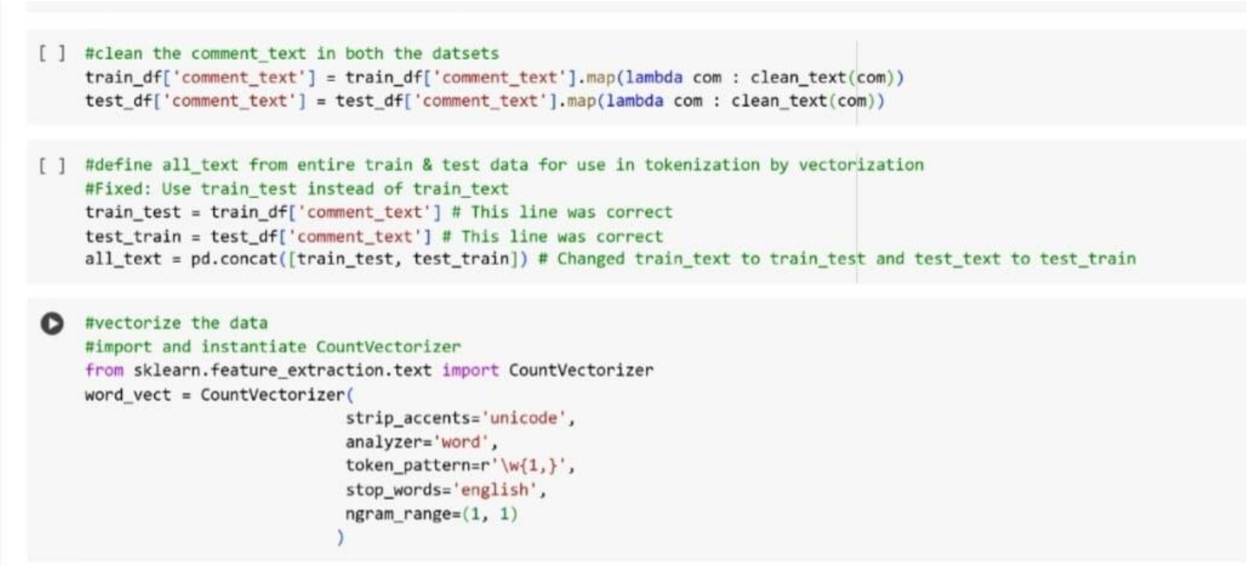
Date	25 October2024
Team ID	739695
Project Title	Toxic Comment Classification for Social Media Using NLP
Maximum Marks	10 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code for toxic comment classification on social media will be shared through a future screenshot, showcasing the process of text preprocessing, vectorization, and training logistic regression models. The model validation and evaluation report will summarize the performance of multiple models, including their accuracy, precision, recall, F1 score, and AUC-ROC metrics, with the details presented via respective screenshots.

Initial Model Training Code (5 marks):

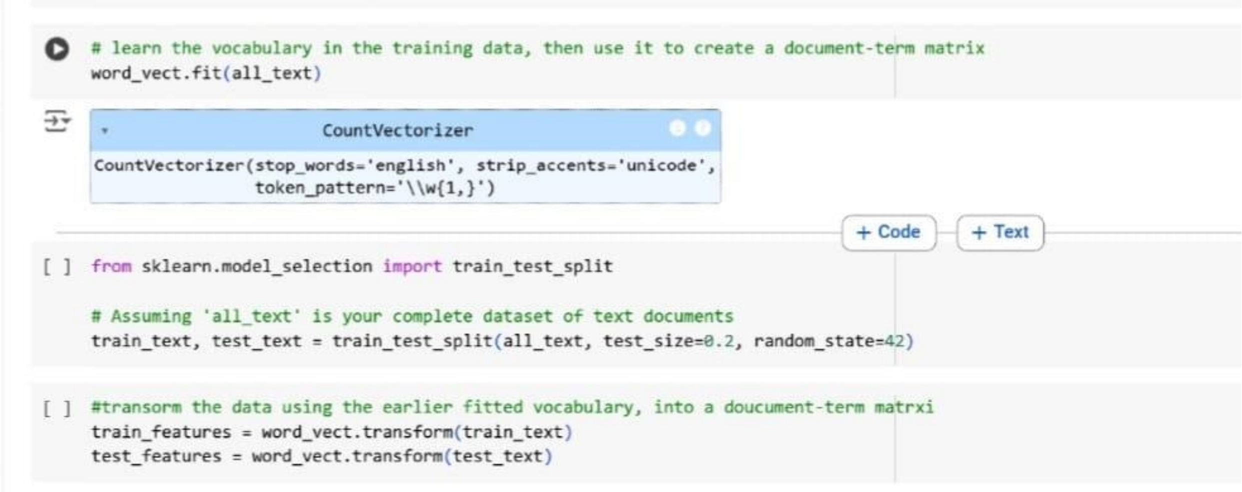
Paste the screenshot of the model training code



```
[ ] #clean the comment_text in both the datsets
    train_df['comment_text'] = train_df['comment_text'].map(lambda com : clean_text(com))
    test_df['comment_text'] = test_df['comment_text'].map(lambda com : clean_text(com))
```

```
[ ] #define all_text from entire train & test data for use in tokenization by vectorization
    #Fixed: Use train_test instead of train_text
    train_test = train_df['comment_text'] # This line was correct
    test_train = test_df['comment_text'] # This line was correct
    all_text = pd.concat([train_test, test_train]) # Changed train_text to train_test and test_text to test_train
```

```
#vectorize the data
#import and instantiate CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
word_vect = CountVectorizer(
                           strip_accents='unicode',
                           analyzer='word',
                           token_pattern=r'\w{1,}',
                           stop_words='english',
                           ngram_range=(1, 1)
                          )
```



```
# learn the vocabulary in the training data, then use it to create a document-term matrix
word_vect.fit(all_text)
```

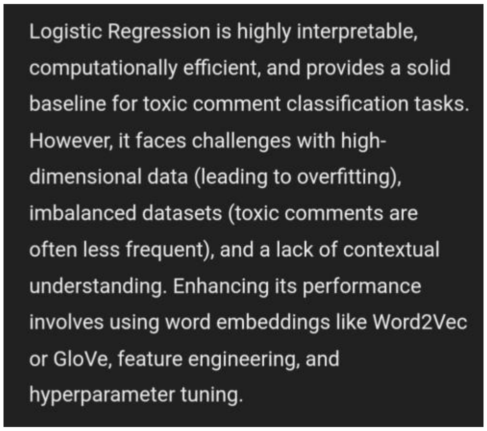
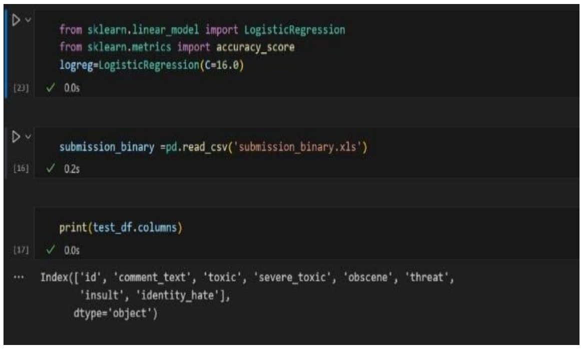
```
CountVectorizer
CountVectorizer(stop_words='english', strip_accents='unicode',
                token_pattern='\\w{1,}')
```

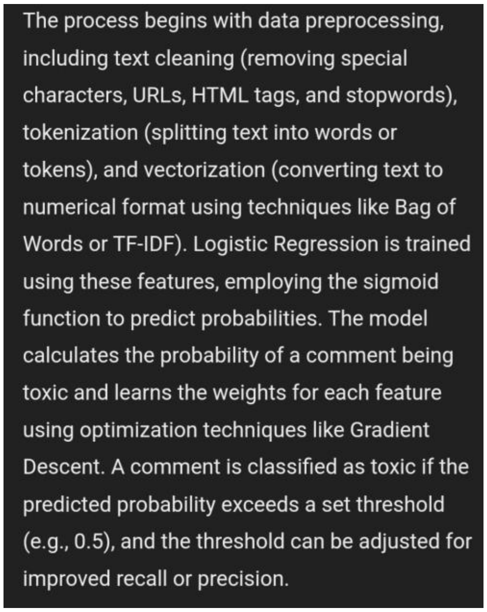
```
[ ] from sklearn.model_selection import train_test_split

    # Assuming 'all_text' is your complete dataset of text documents
    train_text, test_text = train_test_split(all_text, test_size=0.2, random_state=42)
```

```
[ ] #transorm the data using the earlier fitted vocabulary, into a doucument-term matrxi
    train_features = word_vect.transform(train_text)
    test_features = word_vect.transform(test_text)
```

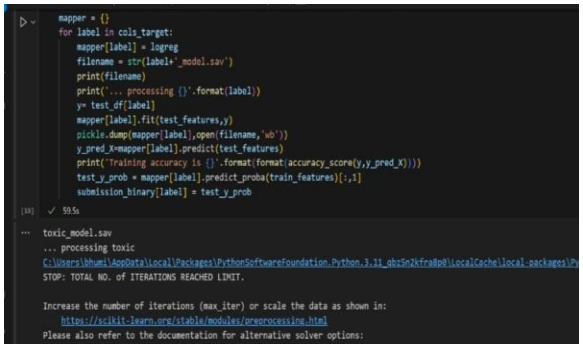
Model Validation and Evaluation Report (5 marks):

Model	Summary	Training and Validation Performance Metrics

Model 1	 Logistic Regression is highly interpretable, computationally efficient, and provides a solid baseline for toxic comment classification tasks. However, it faces challenges with high-dimensional data (leading to overfitting), imbalanced datasets (toxic comments are often less frequent), and a lack of contextual understanding. Enhancing its performance involves using word embeddings like Word2Vec or GloVe, feature engineering, and hyperparameter tuning.	 `from sklearn.linear_model import LogisticRegression`
		`from sklearn.metrics import accuracy_score`
		`logreg=LogisticRegression(C=16.0)`
		[23] ✓ 0.0s
		`submission_binary =pd.read_csv('submission_binary.xls')`
		[16] ✓ 0.2s
		`print(test_df.columns)`
		[17] ✓ 0.0s
		`Index(['id', 'comment_text', 'toxic', 'severe_toxic', 'obscene', 'threat',`
		`'insult', 'identity_hate'],`
		`dtype='object')`



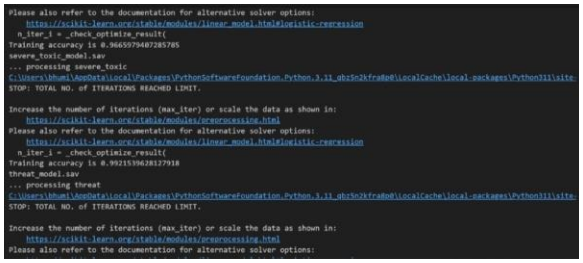
The process begins with data preprocessing, including text cleaning (removing special characters, URLs, HTML tags, and stopwords), tokenization (splitting text into words or tokens), and vectorization (converting text to numerical format using techniques like Bag of Words or TF-IDF). Logistic Regression is trained using these features, employing the sigmoid function to predict probabilities. The model calculates the probability of a comment being toxic and learns the weights for each feature using optimization techniques like Gradient Descent. A comment is classified as toxic if the predicted probability exceeds a set threshold (e.g., 0.5), and the threshold can be adjusted for improved recall or precision.



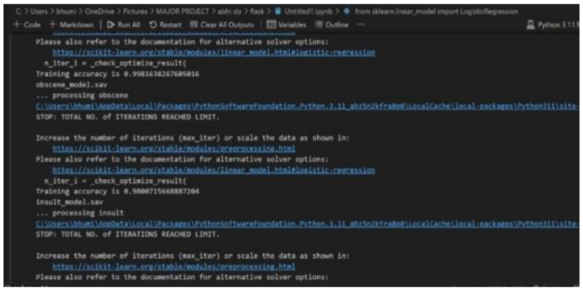
```
mapper = {}
for label in cols_target:
    mapper[label] = logreg
    filename = str(label+'_model.sav')
    print(filename)
    print('... processing {}'.format(label))
    y= test_df[label]
    mapper[label].fit(test_features,y)
    pickle.dump(mapper[label],open(filename,'wb'))
    y_pred_X=mapper[label].predict(test_features)
    print('Training accuracy is {}'.format(format(accuracy_score(y,y_pred_X))))
    test_y_prob = mapper[label].predict_proba(train_features)[:,1]
    submission_binary[label] = test_y_prob
```

[18] ✓ 59.5s

```
toxic_model.sav
... processing toxic
C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Py
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
```

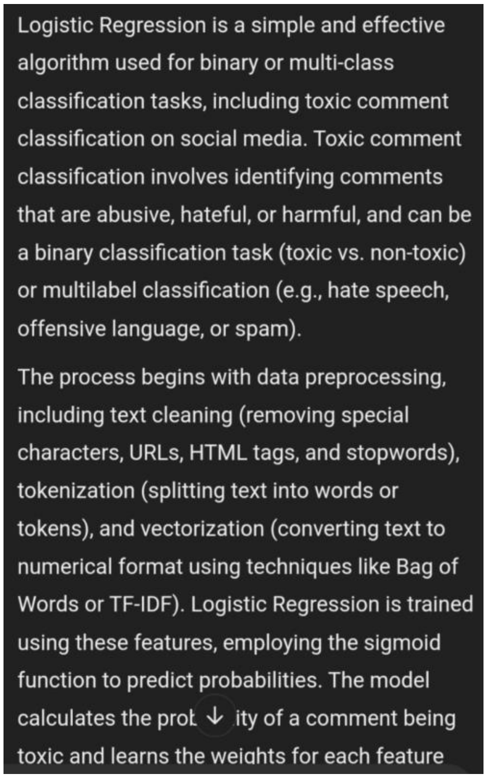
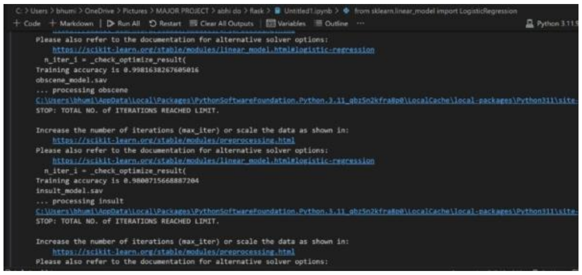


```
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Training accuracy is 0.9665979407285785
severe_toxic_model.sav
... processing severe_toxic
C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Training accuracy is 0.9921539628127918
threat_model.sav
... processing threat
C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
```



C: > Users > bhumi > OneDrive > Pictures > MAJOR PROJECT > abhi do > flask > Untitled1.ipynb > from sklearn.linear_model import LogisticRegression
+ Code
+ Markdown
Run All
Restart
Clear All Outputs
Variables
Outline
Python 3.11.9

```
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Training accuracy is 0.9981638267605016
obscene_model.sav
... processing obscene
C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Training accuracy is 0.9808715668887204
insult_model.sav
... processing insult
C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
```

	
	Logistic Regression is a simple and effective algorithm used for binary or multi-class classification tasks, including toxic comment classification on social media. Toxic comment classification involves identifying comments that are abusive, hateful, or harmful, and can be a binary classification task (toxic vs. non-toxic) or multilabel classification (e.g., hate speech, offensive language, or spam).
	The process begins with data preprocessing, including text cleaning (removing special characters, URLs, HTML tags, and stopwords), tokenization (splitting text into words or tokens), and vectorization (converting text to numerical format using techniques like Bag of Words or TF-IDF). Logistic Regression is trained using these features, employing the sigmoid function to predict probabilities. The model calculates the prob ↓ ity of a comment being toxic and learns the weights for each feature
		
		 C: > Users > bhumi > OneDrive > Pictures > MAJOR PROJECT > abhi do > flask > Untitled1.ipynb > from sklearn.linear_model import LogisticRegression
		+ Code + Markdown ▷ Run All ↻ Restart Clear All Outputs Variables Outline ···
		Python 3.11.9
		`Please also refer to the documentation for alternative solver options:`
		`    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression`
		`  n_iter_i = _check_optimize_result(`
		`Training accuracy is 0.9981638267605016`
		`obscene_model.sav`
		`... processing obscene`
		`C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-`
		`STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.`
		
		`Increase the number of iterations (max_iter) or scale the data as shown in:`
		`    https://scikit-learn.org/stable/modules/preprocessing.html`
		`Please also refer to the documentation for alternative solver options:`
		`    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression`
		`  n_iter_i = _check_optimize_result(`
		`Training accuracy is 0.9800715668887204`
		`insult_model.sav`
		`... processing insult`
		`C:\Users\bhumi\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-`
		`STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.`
		
		`Increase the number of iterations (max_iter) or scale the data as shown in:`
		`    https://scikit-learn.org/stable/modules/preprocessing.html`
		`Please also refer to the documentation for alternative solver options:`