Machine learning Task EmoInt WASSA-2017

-Snitik Swaroop

1. Abstract:

This report presents a methodology for determining the intensity of emotion X in tweets, with the aim of providing a nuanced understanding of emotional content in online communication. Emotion intensity detection is crucial for capturing the degree of specific emotions expressed in text, allowing for more precise sentiment analysis and interpretation. Our methodology integrates feature extraction techniques, machine learning models, and evaluation metrics to predict emotion intensity scores ranging from 0 to 1. We leverage emotion lexicons and databases to enhance feature extraction and explore various machine learning algorithms, including regression and ensemble models. Evaluation metrics such as Mean Absolute Error and Pearson Correlation Coefficient are utilized to assess model performance. Through this approach, we contribute to the advancement of sentiment analysis techniques and deepen our understanding of human emotion in digital discourse.

2. Introduction:

The task given was to find the intensity of the tweet and its primary emotion, the aim is to detect the intensity of tweets.
Four categories separate the emotions:
joy, sadness, anger, and fear.
The intensity is ranked from 0 to 1, with 1 denoting the highest intensity and 0 the lowest. Three types of data are provided: training, development, and test data.Here I used a Regression model that can be trained to predict test data intensity using training and development data.

It is essential to comprehend the emotions that are included into textual input. Although text can be categorized into broad emotion classes using classical emotion analysis, subtle differences in emotional intensity are frequently missed. Emotion intensity detection, which attempts to measure the strength of particular emotions portrayed in text, has arisen in response to this. The goal of this report is to ascertain the level of emotion X in tweets. In this work, we propose a methodology that combines evaluation measures, machine learning models, and feature extraction techniques with the goal of better understanding human emotion in digital discourse

3. Objectives:

1. **Thorough Parameter investigation:**

This research aims to create and assess machine learning models for tweet emotion intensity prediction. The study analyzes tweet data related to various emotions, such as anger, fear, sadness, and joy, in order to investigate the relationship between linguistic content and emotional expression. The main objective is to reliably estimate emotion intensity ratings by training regression models using different methods such Support Vector Regression (SVR), Ridge Regression, and Linear Regression.

2. **Model Creation and Assessment:**

   The study aims to offer insights into how well these models capture the subtleties of emotional expression in text data through data preprocessing, feature engineering, training, and evaluation. The study also seeks to uncover possible uses of emotion intensity prediction in sentiment analysis, social media monitoring, and other domains where analysis and decision-making depend heavily on the comprehension of emotional states in textual data.

3. **Improvement of Current Prediction Systems:**

   Enhancing the existing emotion intensity prediction systems in tweets requires multiple major improvements. First, model performance can be improved by honing feature engineering techniques to capture more subtle features of textual input, like sentiment, context, and linguistic patterns. Furthermore, adding sophisticated natural language processing (NLP) methods, including transformer-based models like BERT or contextual embeddings, can enhance the models' comprehension of tweet semantics. Prediction accuracy can also be increased by investigating ensemble learning techniques to merge predictions from several models or by adding domain-specific knowledge. Furthermore, more reliable and equitable prediction systems may result from correcting bias and class imbalance in training data, especially for underrepresented emotions.

4. Problem Definition:

The task at hand is to accurately forecast the level of emotion in tweets. Understanding the underlying emotions represented in the massive amounts of textual data created on social media platforms every day has become increasingly relevant for a variety of applications, such as sentiment analysis, mental health monitoring, and brand perception analysis. But given the ambiguity, variety, and complexity of human language, measuring the intensity of emotions effectively in tweets is a considerable task. Current methods frequently do less well than ideal when it comes to forecasting emotion intensity scores because they are unable to fully capture the subtle nuances of emotional expression.

The difficulty is further increased by problems including class imbalance, bias in training data, and the dynamic character of language usage. Therefore, the challenge is to create reliable and precise prediction models that can measure and record the strength of emotions in tweets, overcoming the drawbacks and flaws of existing methods.

5. Data set Description

The dataset that was selected for research includes a variety of personal characteristics, including labels text, the intensity itself and the id. The following is a thorough description of the attributes of the dataset:

1. **ID:** A special identification number that is given to every dataset record.
2. **Text:** Any sentence that has some emotions on it.
3. **Label:** The emotion of the tweet like for eg. joy, Fear, Angry,etc
4. **Intensity:** The term "intensity" describes the level or force of an emotion conveyed in a certain tweet.

6. Overview of machine learning model and optimisation technique

The model that is being given is a hybrid of various machine learning algorithms and optimization strategies for the prediction intensity of the tweets . This all-inclusive model covers a number of phases, such as data pretreatment, exploratory data analysis, model creation, metrics for evaluation, and deployment features. Logistic regression, Support Vector Regression (SVR),are the main machine learning models used in the model.

Logistic Regression:

One of the most important statistical methods is logistic regression, which is mostly applied to binary classification and probability prediction for binary outcomes. The simplicity and interpretability of this approach are its defining features. It functions by simulating the correlation between one or more independent variables and a binary dependent variable.

Application in the Project:

Since logistic regression is usually used for binary classification tasks, it is not immediately suitable in this research for predicting continuous emotion intensity ratings in tweets. It can still be helpful, though, in situations when tweets on a particular emotion are binary classified into high and low intensity groups. Because of its interpretable coefficients, logistic regression can also help with feature importance analysis and interpretation. When paired with other models, it can also function as a base learner in ensemble approaches, improving overall prediction performance.

Optimization Techniques and Data Preprocessing:

Preprocessing data:

1. **Inaccurate Data Processing:** Model performance may be impacted by missing values in a number of characteristics, To deal with missing values, imputation techniques such as mean, median, or mode replacement may be used.
2. **Transformation of Data:** Numerical formatting of categorical attributes is required. Machine learning algorithms benefit from the representation of categorical variables through the use of strategies such as one-hot encoding.
3. **Tokenization**: To enable more processing and analysis, tokenize the text into discrete words or tokens.
   Utilizing methods such as WordPunctTokenizer or regular expressions, divide text into tokens while properly managing special characters and punctuation.
4. **Stopword Removal:** removing frequently used stopwords that may add noise to the analysis and don't have any semantic significance, such as "and," "the," and "is."modifying stopword lists in accordance with the unique context of the data or using pre-made lists from libraries like NLTK or spaCy.

7. Platform and Tools Used:

1. Jupyter Notebook:

It is a well known open-source interactive computing environment that permits its users to make and share archives containing live code, equations, visualizations, and narrative text, making it a favored choice for data scientists and analysts.

2. Google Colab:

Google Colab brief for Google Colaboratory, is a free cloud-based Jupyter notebook platform that gives access to GPU and TPU hardware for machine learning and data analysis, making it a helpful choice for collaborative ventures and data science within the cloud.

3. Python Libraries:

```
import pandas as pd
import numpy as np
import string
import re
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
#nlp
from collections import Counter
```

```
import nltk
from nltk.tokenize import WordPunctTokenizer
from nltk.corpus import stopwords
import sklearn
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
#regression model
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import BayesianRidge
from sklearn.linear_model import Ridge
from sklearn import svm
from sklearn import tree
from sklearn import neighbors
from sklearn import metrics
import scipy
```

8. Software Requirement Specifications:

8.1 - Introduction and Purpose

In this section, we delve into the core aspects of our project's introduction and purpose.. These elements are vital as they set the foundation for our entire project..

Introduction:

Our project revolves around data analysis and machine learning tasks implemented using Python within the Jupyter Notebook and Google Colab environments. The primary goal is to address a specific real-world problem, which, in our case, is the prediction emotion intensity based on the provided data

8.2 - Software Tools

Jupyter Notebook:

Jupyter Notebook serves as one of the primary software tools for your project.. It is an open-source interactive computing environment that facilitates the creation and sharing of documents containing live code, equations, visualizations, and narrative text.. With Jupyter Notebook, we can efficiently develop and document our data analysis and machine learning code..

Google Colab:

Google Colab, short for Google Colaboratory, complements Jupyter Notebook by offering a cloud-based environment for collaborative work.. It provides access to GPU and TPU hardware, making it a valuable resource for machine learning tasks, especially when dealing with large datasets.. Google Colab enables seamless collaboration and allows team members to work on the same project concurrently, fostering efficient teamwork..

8.3 - Hardware Tools

Local Machine:

The local machine, which is a laptop, serves as our personal workstation.. These provide the necessary hardware support to run Jupyter Notebooks.. A local machine with a robust processor and memory allows us to efficiently develop, test, and debug your code. Additionally, we will be able to easily use Python libraries and easily visualize our data..

Cloud-based servers:

For tasks that require large amounts of computing power and storage, we rely on cloud-based servers.. These servers are hosted remotely and accessible over the internet, making them suitable for running code in Google Colab.. They provide powerful hardware resources such as GPUs and TPUs that allow us to process data, train machine learning models, and perform complex calculations.. Cloud-based servers allow projects to scale and process large datasets effectively..

These hardware tools, combined with our software tools, provide a comprehensive infrastructure to support our project's data analysis and machine learning needs.. This combination of hardware and software tools allows us to work efficiently and achieve our project goals.

8.4 Work Structure:

This section partitions the project into distinctive stages and tasks that must be completed to attain the project objectives.. This working structure served as a guide for the project and helped us remain organized all through the improvement preparation.

1. Data Collection:

Data collection/gathering was the primary step of our project. I obtained the dataset from CodeLab and it made a .txt file, a well-known platform for data science and machine learning datasets. The dataset is titled "fear.txt, anger.txt, joy.txt, sadness.txt and many more for development and test and train".

2. Data Preprocessing:

Before continuing with analysis and modeling, we would like to ensure that our dataset is clean and well organized. Data preprocessing includes a few errands:

- Processing categorical data:

Categorical data was converted to numerical format, utilizing one-hot encoding to make it appropriate for machine learning.

3. Data Examination:

Usually where the actual exploration and understanding of the data takes place. Using Python libraries such as Pandas, NumPy, Matplotlib, and Seaborn, We did the following:

5. Results and Visualization:

After we constructed and assessed our machine learning model, presented our results through visualization and performance metrics. This step incorporates:

- Making ROC curves to assess model performance..
- Generate a heatmap to visualize the confusion matrix and give insights into the accuracy of your model..
- Calculated RMSE, MSE, MAE scores, and accuracy to measure our model's performance.

8.5 Functional Requirements:

Functional requirements are the essential skills and behaviors needed to achieve the project goals.These form the basis of the functional and operational aspects of the project. More specifically, the project's functional requirements include the following aspects:

Data Manipulation Skills:

Our projects require the ability to deftly manipulate and transform data.. This includes using Pandas, a Python library for performing tasks such as filtering, sorting, and aggregating data.

Statistical Skills:

Our project includes a number of statistical techniques to uncover patterns and relationships in data.. Use NumPy to perform advanced mathematical operations on datasets, calculate means and variances, and enable accurate statistical tests..

Beautiful Data Representation:

Creating visually appealing and informative data visualizations using the Matplotlib and Seaborn libraries to present our results.. These libraries allow us to create line graphs, histograms, and scatter plots, making our projects not only informative but also visually appealing..

8.6.Non-Functional Requirements:

Non-functional requirements include qualities and characteristics that improve the overall efficiency and quality of our projects, going beyond mere functionality.. These include the following aspects:

Optimized performance:

We prioritize the efficiency of our code, ensuring it performs well, even when dealing with large datasets.This commitment to optimal performance delivers rapid execution and response..

Flexible portability:

The code we develop is designed to be flexible and adaptable, making it compatible with Jupyter Notebook and Google Colab environments.. This flexibility allows for seamless transitions between on-premises and cloud workspaces..

User-friendly documentation:

We understand how crucial accessibility and clarity are. Our projects come with extensive documentation that covers every aspect of functionality and usability. This documentation enhances the user experience and fosters productive teamwork.

Data Collection and Mining:

Initially, the project involves gathering data and importing essential Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, and many more' to effectively handle unbalanced datasets.

To access the dataset, Google Drive is mounted using the Google Colab environment.

Data preprocessing:

The preparation of datasets is at the center of the project.

The dataset is imported and stored in a DataFrame, denoted as df, using Pandas.

This can come after the first phase of data discovery, which comprises showing the dataset's initial structure, line and column names, outline measurements, form, and data identification, as well as counting to start.

Data visualization and analysis:

Creating a series of data visualizations with Matplotlib and Seaborn is what defines this level.

The goal of these visualizations is to find underlying patterns, relationships, and conveyances in a dataset by combining histograms, count plots, line plots, box plots, and pair plots.

Data transformation and preparation:

A number of transformations are applied to the dataset, including the one-time encoding of categorical features.

Additionally, the feature and target tables (x and y) are constructed to ensure that they work with machine learning models.

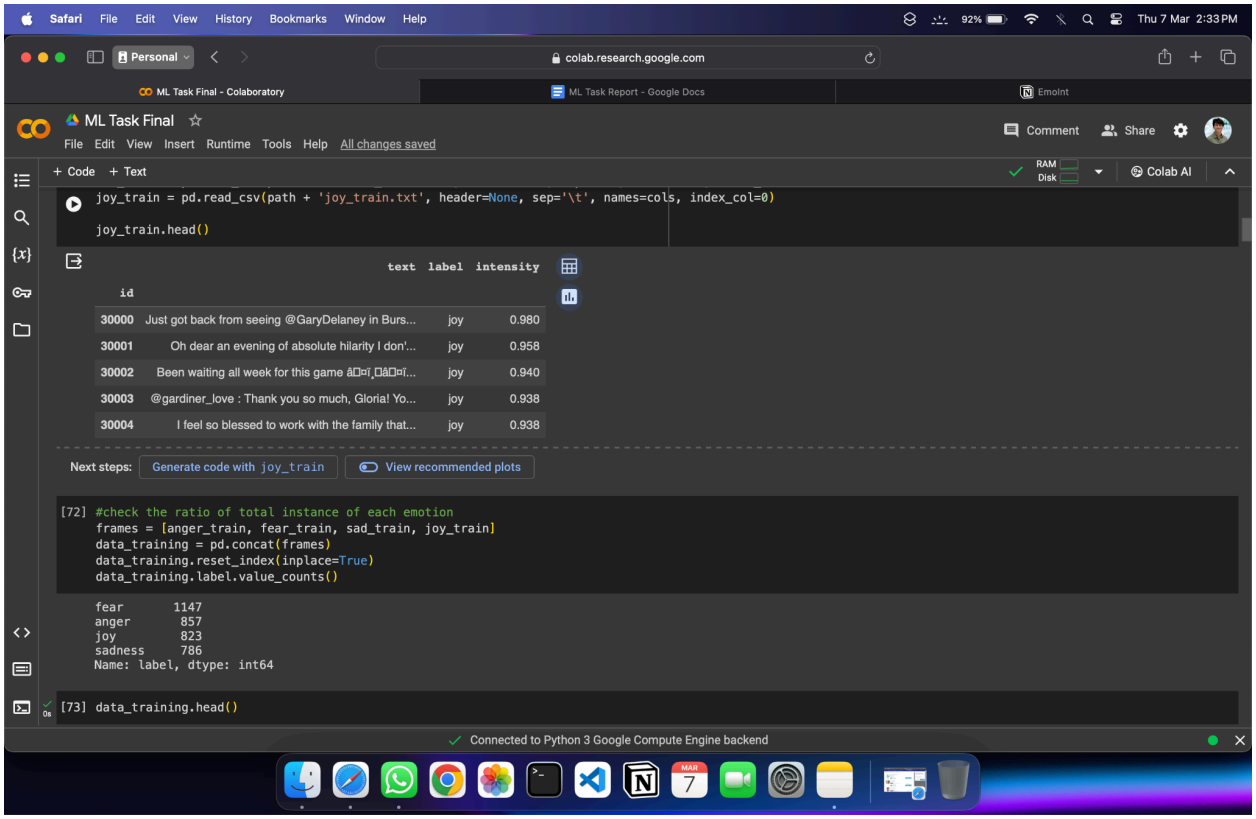
Data Partitioning:

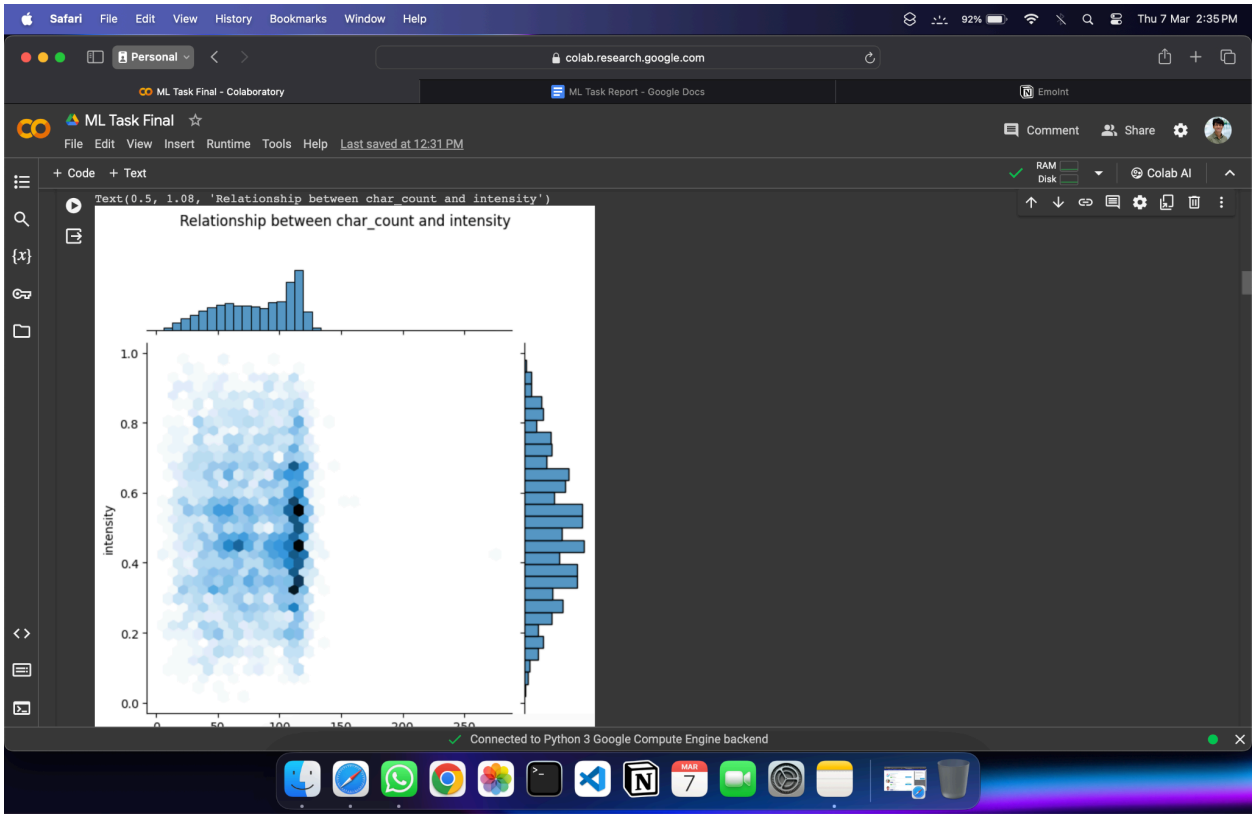
A critical step in model development, data partitioning is accomplished through the utilization of 'train_test_split' from the scikit-learn library. This partitioning isolates the dataset into a training set and a test set.

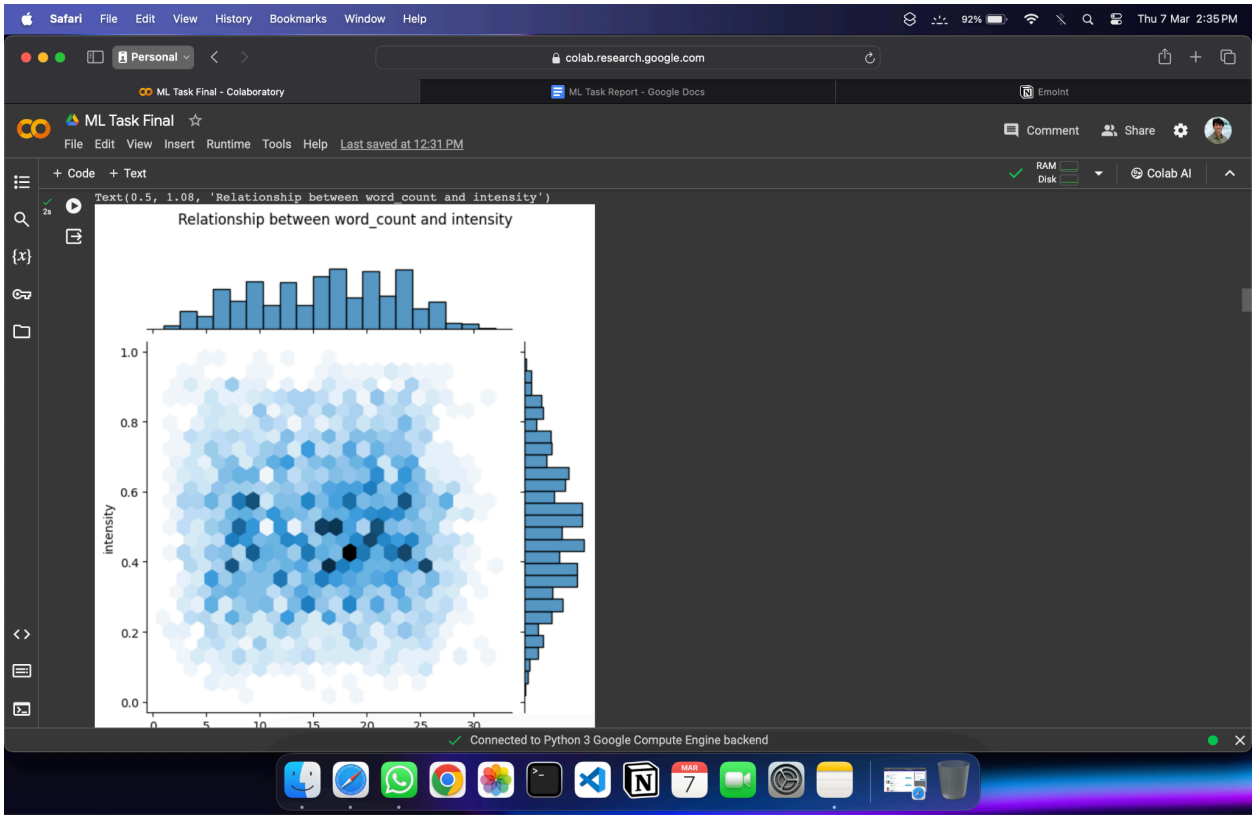
System output:

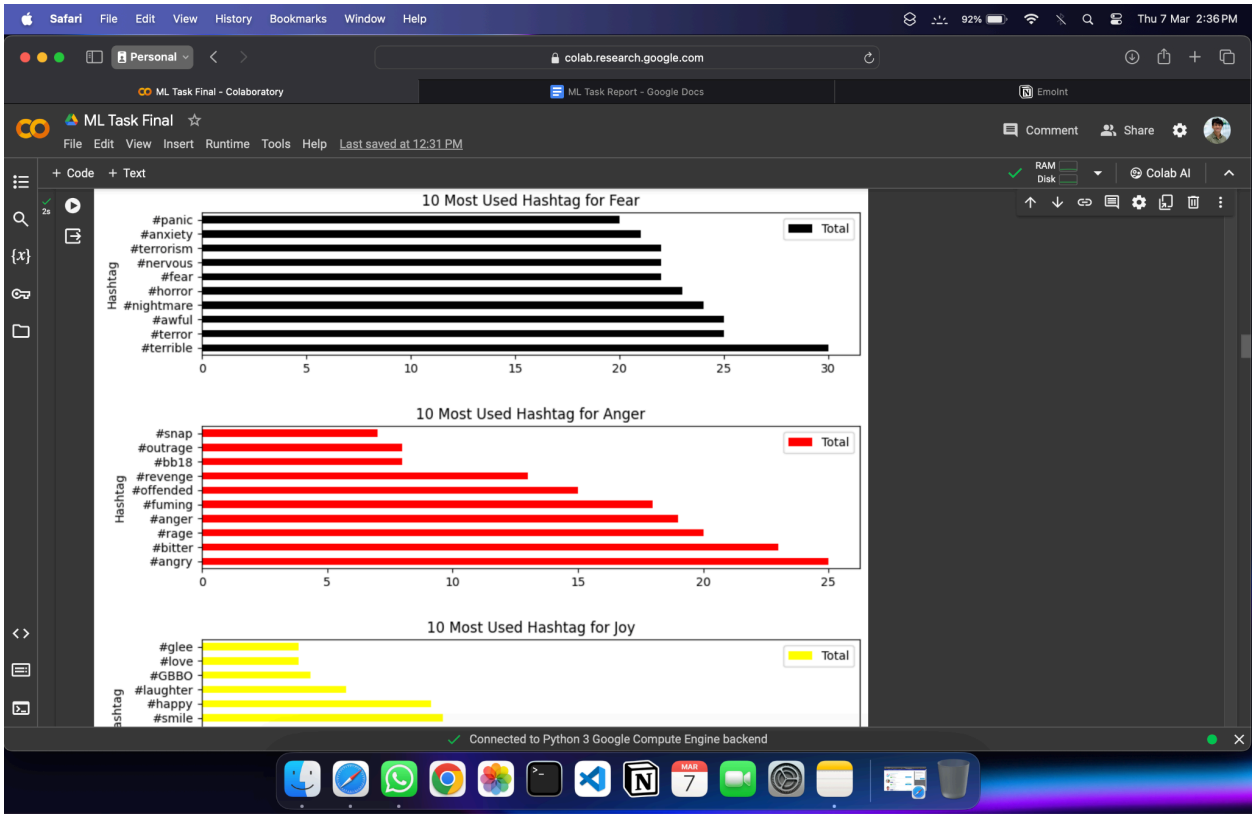
In summary, the project report presents conclusive results on the precision and predictive performance of each model, providing valuable information for informed decision-making.

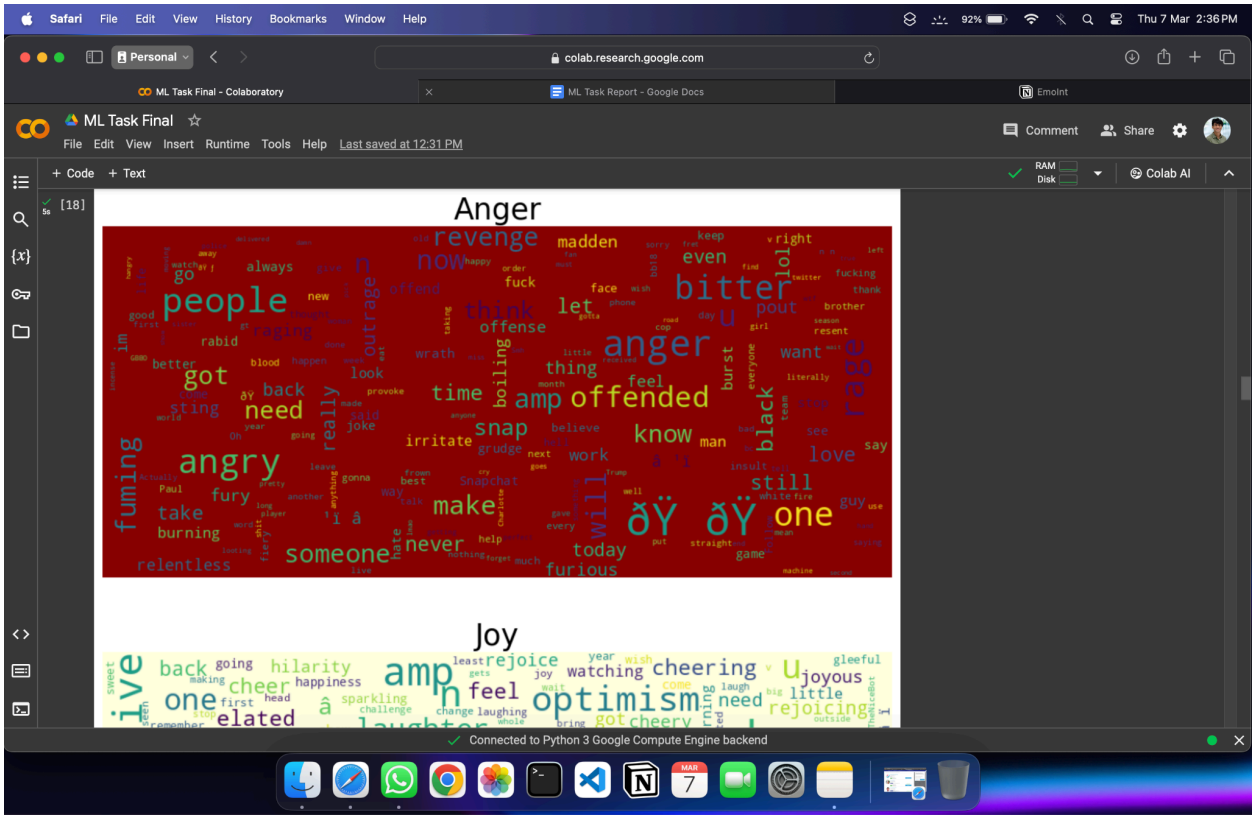
10. Implementation:




Safari File Edit View History Bookmarks Window Help
Thu 7 Mar 2:35 PM
colab.research.google.com
ML Task Final - Colaboratory
ML Task Report - Google Docs
Emolnt
ML Task Final
File Edit View Insert Runtime Tools Help Last saved at 12:31 PM
Comment
Share
+ Code + Text
RAM
Disk
Colab AI
Text(0.5, 1.08, 'Relationship between char_count and intensity')
Relationship between char_count and intensity
intensity
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
250
Connected to Python 3 Google Compute Engine backend


Safari File Edit View History Bookmarks Window Help
Thu 7 Mar 2:35 PM
colab.research.google.com
ML Task Final - Colaboratory
ML Task Report - Google Docs
Emolnt
ML Task Final
File Edit View Insert Runtime Tools Help Last saved at 12:31 PM
Comment
Share
+ Code + Text
RAM
Disk
Colab AI
Text(0.5, 1.08, 'Relationship between word_count and intensity')
Relationship between word_count and intensity
intensity
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Connected to Python 3 Google Compute Engine backend

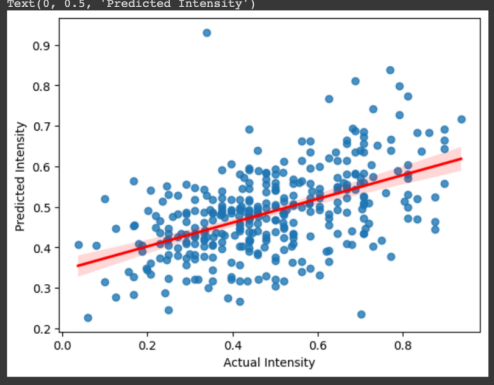

Safari File Edit View History Bookmarks Window Help
Thu 7 Mar 2:36 PM
colab.research.google.com
ML Task Final - Colaboratory
ML Task Report - Google Docs
Emolnt
ML Task Final
File Edit View Insert Runtime Tools Help Last saved at 12:31 PM
Comment Share
+ Code + Text
Colab AI
10 Most Used Hashtag for Fear
Hashtag
#panic
#anxiety
#terrorism
#nervous
#fear
#horror
#nightmare
#awful
#terror
#terrible
Total
0 5 10 15 20 25 30
10 Most Used Hashtag for Anger
Hashtag
#snap
#outrage
#bb18
#revenge
#offended
#fuming
#anger
#rage
#bitter
#angry
Total
0 5 10 15 20 25
10 Most Used Hashtag for Joy
#glee
#love
#GBBO
#laughter
#happy
#smile
Total
Connected to Python 3 Google Compute Engine backend


Safari File Edit View History Bookmarks Window Help
Thu 7 Mar 2:36 PM
colab.research.google.com
ML Task Final - Colaboratory
ML Task Report - Google Docs
Emolnt
ML Task Final
File Edit View Insert Runtime Tools Help Last saved at 12:31 PM
Comment Share
+ Code + Text
Colab AI
[18]
Anger
people
anger
bitter
offended
angry
revenge
madden
fuming
Joy
optimism
cheering
rejoice
joyous
rejoicing
Connected to Python 3 Google Compute Engine backend

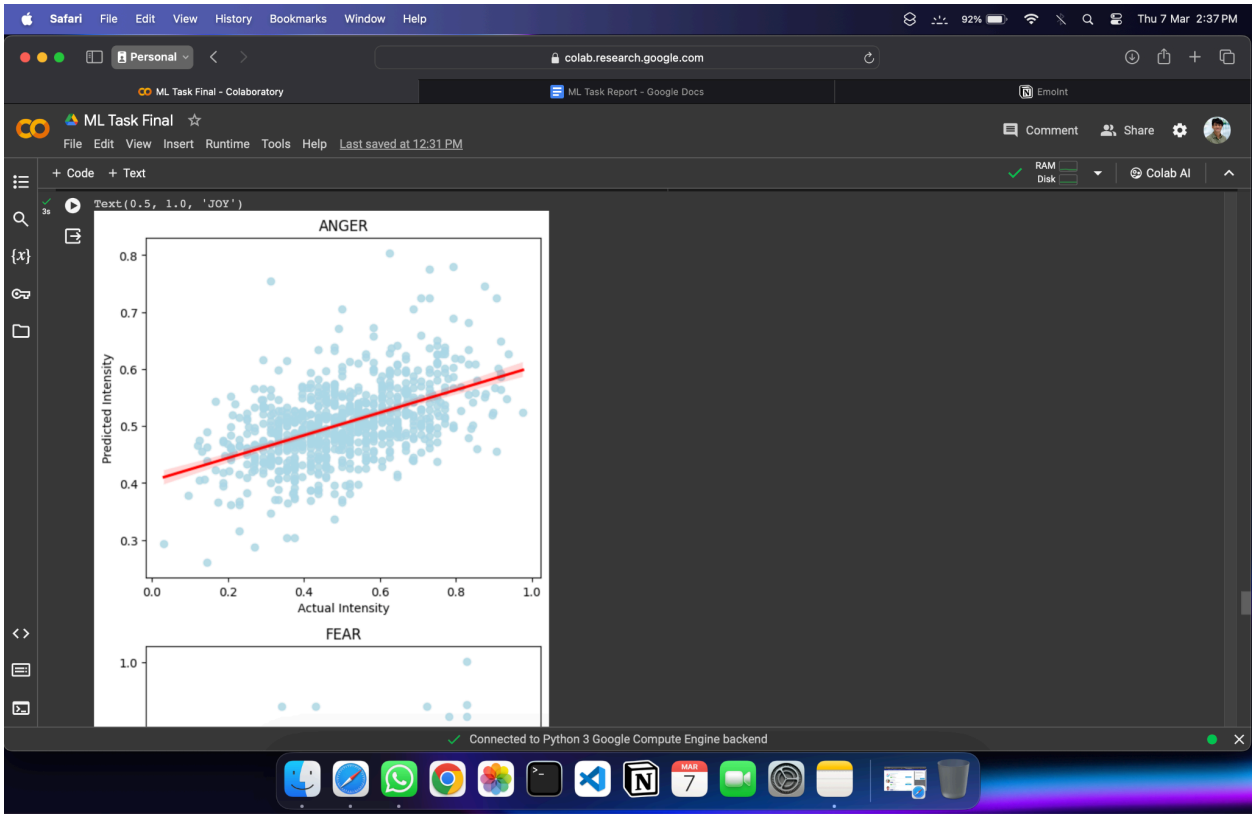
```
Mean Squared Error: 0.03738940950655319
Root Mean Squared Error: 0.19336341305053856
--------------------------------------------------------------------------------
Tfidf-Linear Regression

Mean Absolute Error: 0.14956344560518733
Mean Squared Error: 0.03647894621052591
Root Mean Squared Error: 0.19099462351209237
--------------------------------------------------------------------------------
BoW-Ridge Regression

Mean Absolute Error: 0.14353399764269448
Mean Squared Error: 0.033005283211300494
Root Mean Squared Error: 0.18167356224239326
--------------------------------------------------------------------------------
Tfidf-Ridge Regression

Mean Absolute Error: 0.2465235398821465
Mean Squared Error: 0.10510137600845543
Root Mean Squared Error: 0.32419342375880394
--------------------------------------------------------------------------------
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a den
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:768: UserWarning: pandas.DataFrame with sparse columns found.It will be converted to a den
  warnings.warn(
BoW-Knn Regression

Mean Absolute Error: 0.14846224783861672
Mean Squared Error: 0.03472618259365994
Root Mean Squared Error: 0.18634962461368132
--------------------------------------------------------------------------------
Tfidf-Knn Regression

Mean Absolute Error: 0.15290144092219024
Mean Squared Error: 0.036061280691642654
Root Mean Squared Error: 0.18898807974711764
--------------------------------------------------------------------------------
```

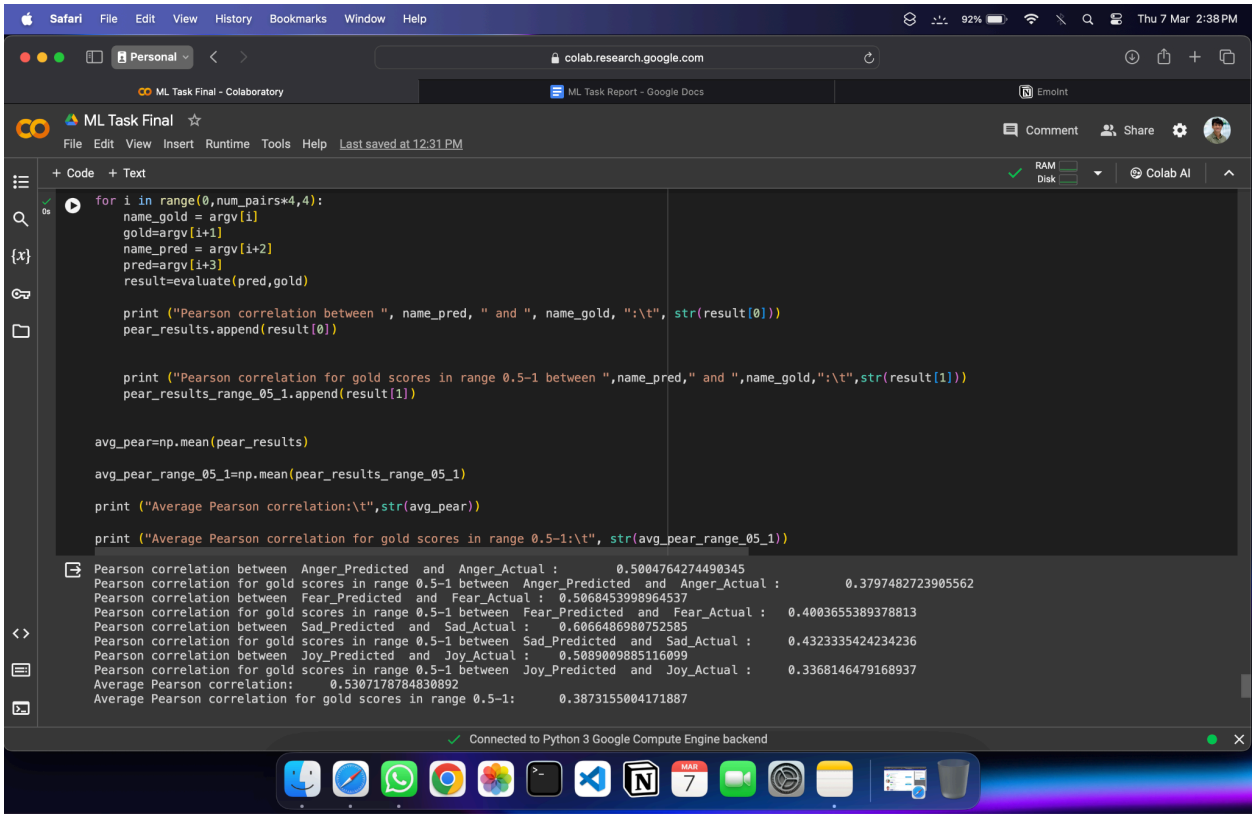
```
plt.ylabel("Predicted Intensity")
```

```
Text(0, 0.5, 'Predicted Intensity')
```




```
[34] #Training the final model with the combination of training and development data
     combined_training = pd.concat([data_training[['id', 'text', 'label', 'intensity']], data_dev]).reset_index()
     combined_training.shape

     (3960, 5)
```


Safari File Edit View History Bookmarks Window Help
Thu 7 Mar 2:37 PM
colab.research.google.com
ML Task Final - Colaboratory
ML Task Report - Google Docs
EmoInt
ML Task Final
File Edit View Insert Runtime Tools Help Last saved at 12:31 PM
Comment
Share
+ Code + Text
RAM
Disk
Colab AI
Text(0.5, 1.0, 'JOY')
ANGER
Predicted Intensity
Actual Intensity
0.0
0.2
0.4
0.6
0.8
1.0
0.3
0.4
0.5
0.6
0.7
0.8
FEAR
1.0
Connected to Python 3 Google Compute Engine backend



pear_results
spear_results

```
Pearson correlation between  Anger_Predicted  and  Anger_Actual :        0.5004764274490345
Pearson correlation for gold scores in range 0.5-1 between  Anger_Predicted  and  Anger_Actual :        0.3797482723905562
Pearson correlation between  Fear_Predicted  and  Fear_Actual :  0.5068453998964537
Pearson correlation for gold scores in range 0.5-1 between  Fear_Predicted  and  Fear_Actual :   0.4003655389378813
Pearson correlation between  Sad_Predicted  and  Sad_Actual :    0.6066486980752585
Pearson correlation for gold scores in range 0.5-1 between  Sad_Predicted  and  Sad_Actual :     0.4323335424234236
Pearson correlation between  Joy_Predicted  and  Joy_Actual :    0.5089009885116099
Pearson correlation for gold scores in range 0.5-1 between  Joy_Predicted  and  Joy_Actual :     0.3368146479168937
Average Pearson correlation:     0.5307178784830892
Average Pearson correlation for gold scores in range 0.5-1:      0.3873155004171887
```

11. Results:

The analysis's conclusions include a number of significant discoveries. First off, there is no discernible difference in the average strength of the four emotions analyzed, according to the Exploratory Data Analysis (EDA). Furthermore, there is no discernible relationship between intensity and the length of the text or the use of punctuation, therefore these parameters are not included in the model training.

Testing several vectorization methods and algorithms was part of the evaluation process for regression models; the Support Vector Regression (SVR) model, which used TF-IDF vectorization, had the lowest error rate and was chosen as the final model. Emotion-specific models were created, and evaluation metrics indicated that they performed better than chance; however, in terms of error rates, the anger model fared better than the pleasure model. Pearson The results of correlation analysis showed that the anticipated and actual intensity scores had a positive linear relationship, while the correlation coefficients

for tweets with higher intensities were smaller. Limitations were noted despite the encouraging results, especially in correctly anticipating tweets with severe intensity levels.

Spelling-checking tweet text, growing the training dataset, and experimenting with several algorithms for each emotion are some suggested improvement tactics. Though the models are promising overall, more improvement and optimization are required to improve prediction accuracy and resolve current issues with emotion intensity prediction from twitter data.