CHAPTER 4

Initial Findings and Analysis

4.1 Overview

This chapter present a comprehensive overview of the initial findings and analytical process for this project that focused on detecting potential mental health crises using Malaysian Reddit data. This chapter involved several key stages, including web scrapping, data preprocessing, text cleaning, exploratory data analysis (EDA), emotion classification using DistilBERT, high-risk post identification and model interpretation using XAI techniques like LIME.

4.2 Web Scraping

In this project, the dataset the social media text post from Malaysia Reddit post. Thus, in this project web scraping for social media Reddit was done. Library 'PRAW' was used to scrape the data from Reddit. The first step of the web scraping is to define the key words, this will make sure when scrape the post, it will only take the post that consist of the key words. Due to this project is about mental health crises, the key words set are "mental health", "depression", "anxiety", and "stress". Other than that, to scrape only Malaysian post, subreddit was set to "Malaysia" to try searching r/Malaysia for each keyword.

```
search_terms = ['mental health', 'depression', 'anxiety', 'stress']
posts = []
```

Figure 4.1: Key word to scrape for social media post.

In this project, it did not just scrape for the post title and text, but it also scrapes for the post comments. Thus, the dataset for web scraping consists of 12 columns which is “post_id”, “title”, “selftext”, “post_score”, “upvote_ratio”, “num_comments”, “created_utc”, “comment_id”, “comment_body”, “comment_score”, “comment_awards”, and “comment_created_utc”.

```python
with open('mentalhealth.csv', 'w', newline='', encoding='utf-8') as f:
    writer = csv.writer(f)
    writer.writerow([
        'post_id', 'title', 'selftext', 'post_score', 'upvote_ratio',
        'num_comments', 'created_utc',
        'comment_id', 'comment_body', 'comment_score',
        'comment_awards', 'comment_created_utc'
    ])

    for submission in posts:
        submission.comments.replace_more(limit=0)
        for comment in submission.comments.list():
            writer.writerow([
                submission.id,
                submission.title,
                submission.selftext.replace('\n', ' ').replace('\r', ''),
                submission.score,
                submission.upvote_ratio,
                submission.num_comments,
                submission.created_utc,
                comment.id,
                comment.body.replace('\n', ' ').replace('\r', '').strip(),
                comment.score,
                comment.total_awards_received,
                comment.created_utc
            ])
```

Figure 4.2: Code for the scraping details.

After scrape the data was saved into a CSV file for further used. In this project, the web scraping has successfully scrape for 13,723 data. Thus, the final scrape dataset consists of 13,723 rows and 12 columns.

	post_id	title	selftext	post_score	upvote_ratio	num_comments	created_utc	comment_id	comment_body	comment_score	comment_awards	com
0	1h704et	Malaysian psychiatrist with 'promising career'...	NaN	137	0.94	46	1.733371e+09	m0j5twr	Raped. He raped a minor entrusted under his ca...	27	0	
1	1h704et	Malaysian psychiatrist with 'promising career'...	NaN	137	0.94	46	1.733371e+09	m0hiscs	Apparently this dude is bro of Dr halina wife ...	74	0	
2	1h704et	Malaysian psychiatrist with 'promising career'...	NaN	137	0.94	46	1.733371e+09	m0hs8q8	Like my mom always asked, 'Anak siapa ni?'	18	0	
3	1h704et	Malaysian psychiatrist with 'promising career'...	NaN	137	0.94	46	1.733371e+09	m0hpmsh	> She reportedly said the married Amirul Arif ...	28	0	
4	1h704et	Malaysian psychiatrist with 'promising career'...	NaN	137	0.94	46	1.733371e+09	m0hjkkv	nerakazens are doing their job at x. hehehe ...	12	0	
...	...	...	...	...	...	...	...	...	...	...	...	
13718	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan stress	NaN	183	0.94	15	1.626157e+09	h51smfn	Hahahaa so funny eh? What about the pakages? T...	-7	0	
13719	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan stress	NaN	183	0.94	15	1.626157e+09	h50tvo7	No wonder my package missing!	15	0	
13720	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan stress	NaN	183	0.94	15	1.626157e+09	h51h5pi	Haha just allowed it bro	1	0	
13721	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan stress	NaN	183	0.94	15	1.626157e+09	h51fswb	Biar lambat asalkan selamat	4	0	
13722	1ar8mxe	Free Stress Management Workshops!	NaN	8	1.00	1	1.707977e+09	kqxxdwj	UPDATE: We only have very few spaces left for ...	1	0	

13723 rows × 12 columns

Figure 4.3: Data that scrape from Reddit.

4.3 Data Preprocessing

Data preprocessing involving some step like check for missing value and fill in with suitable things. The first step of data preprocessing is check for the dataset info and check for missing value. In this project these two step was run together. Form the figure below, there have the info for the dataset, which data type for each column was showed and below it has the missing value for each column. The column 'post_id', 'title', 'selftext', 'comment_id', and 'comment_body' was object, which mean that the data was words. Columns like 'post_score', 'num_comments', 'comment_score', and 'comment_awards' data type was integer, while column 'upvote_ratio', 'created_utc', and 'comment_created_utc' data type was float number. The dataset also has missing value for column 'selftext' which have 4757 missing values.

```
mentalhealth.info()
mentalhealth.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13723 entries, 0 to 13722
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   post_id             13723 non-null  object 
 1   title               13723 non-null  object 
 2   selftext            8966 non-null   object 
 3   post_score          13723 non-null  int64  
 4   upvote_ratio        13723 non-null  float64
 5   num_comments        13723 non-null  int64  
 6   created_utc         13723 non-null  float64
 7   comment_id          13723 non-null  object 
 8   comment_body        13723 non-null  object 
 9   comment_score       13723 non-null  int64  
 10  comment_awards      13723 non-null  int64  
 11  comment_created_utc 13723 non-null  float64
dtypes: float64(3), int64(4), object(5)
memory usage: 1.3+ MB
```

```
post_id                0
title                  0
selftext            4757
post_score             0
upvote_ratio           0
num_comments           0
created_utc            0
comment_id             0
comment_body           0
comment_score          0
comment_awards         0
comment_created_utc    0
dtype: int64
```

Figure 4.4: Info and count of missing value of the dataset.

To handle missing values in the 'selftext' column, empty entries were replaced with whitespace.

```
mentalhealth_drop['full_content'] = mentalhealth_drop['title'] + ' ' + mentalhealth_drop['selftext'] + ' ' + mentalhealth_drop['comment_body']
mentalhealth_drop
```

	post_id	title	selftext	post_score	upvote_ratio	num_comments	created_utc	comment_id	comment_body	comment_score	comment_awards	comm
0	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0j5twr	Raped. He raped a minor entrusted under his ca...	27	0	
1	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hiscs	Apparently this dude is bro of Dr halina wife ...	74	0	
2	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hs8q8	Like my mom always asked, 'Anak siapa ni?'	18	0	
3	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hpmsh	> She reportedly said the married Amirul Arif ...	28	0	
4	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hjkkv	nerakazens are doing their job at x. hehehe ...	12	0	
...	...	...	...	...	...	...	...	...	...	...	...	
13718	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan stress 😄		183	0.94	15	1.626157e+09	h51smfn	Hahahaa so funny eh? What about the pakages? T...	-7	0	
13719	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan		183	0.94	15	1.626157e+09	h50tvo7	No wonder my package missing!	15	0	

Figure 4.5: Fill in the missing value with white space.

Before do the data cleaning for the text data, the column that have important information like 'title', 'selftext' and 'comment_body' were combine together to make it become more meaning full.

```
mentalhealth_drop['full_content'] = mentalhealth_drop['title'] + ' ' + mentalhealth_drop['selftext'] + ' ' + mentalhealth_drop['comment_body']
mentalhealth_drop
```

	post_id	title	selftext	post_score	upvote_ratio	num_comments	created_utc	comment_id	comment_body	comment_score	comment_awards	comm
0	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0j5twr	Raped. He raped a minor entrusted under his ca...	27	0	
1	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hiscs	Apparently this dude is bro of Dr halina wife ...	74	0	
2	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hs8q8	Like my mom always asked, 'Anak siapa ni?'	18	0	
3	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hpmsh	> She reportedly said the married Amirul Arif ...	28	0	
4	1h704et	Malaysian psychiatrist with 'promising career'...		137	0.94	46	1.733371e+09	m0hjkkv	nerakazens are doing their job at x. hehehe ...	12	0	
...	...	...	...	...	...	...	...	...	...	...	...	
13718	oj9ooa	Hidup kena happy.. Rehat jap.. Hilangkan stress 😄		183	0.94	15	1.626157e+09	h51smfn	Hahahaa so funny eh? What about the pakages? T...	-7	0	

Figure 4.6: Combine 3 columns into one new column.

4.4 Data cleaning

Data cleaning was the most important step before the emotion classification. In this phase, it will do several steps to clean the data to make sure it ready for the model to do the emotion classification. The first step of data cleaning was finding all the text noise that were not alphabet. Based on the Figure 4.7, there was many non-alphabetic characters find in the dataset. And the output show that have digits, multiple space and links in the dataset.

```python
def inspect_text_noise(text):

  full_text = ' '.join(text.astype(str))

  non_alpha = sorted(set(re.findall(r'[^a-zA-Z\s]', full_text)))
  has_digits = any(char.isdigit() for char in full_text)
  has_multiple_spaces = bool(re.search(r'\s{2,}', full_text))
  urls = re.findall(r'http[s]?://\S+', full_text)


  print("Non-alphabetic characters:", non_alpha)
  print("Contains digits:", has_digits)
  print("Contains multiple spaces:", has_multiple_spaces)
  print("Sample URLs found:", urls[:5])


inspect_text_noise(mentalhealth_drop['full_content'])
```

```
Non-alphabetic characters: ['!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9',
':', ';', '<', '=', '>', '?', '@', '[', '\\', ']', '^', '_', '`', '{', '|', '}', '~', '£', '\xad', '¯', '°', '²', '×', 'à', 'é', 'ï', 'ö', 'ø', 'ù', 'ü',
[illegible]
'த', 'ள', 'ப', 'ம', 'வ', 'ோ', 'ி', [illegible], '\u200d', '–', '—', '‘', '’', '“', '”', '•', '…', '‼', '™', '↕', '≠', [illegible]
'⊙', '♀', '♂', '♡', '♥', '⛔', '✂', '✅', '✊', '✌', '✨', '❌', '❖', '❤', '⭐', 'エ', 'ツ', 'ロ', '上', '不', '中', '书', '了', '亞', '人', '來',
'個', '做', '兄', '割', '动', '华', '同', '呀', '命', '啊', '啦', '嗨', '固', '完', '宝', '弟', '得', '心', '快', '情', '意', '感', '执', '放', '文', '棄',
'樣', '死', '永', '没', '潤', '物', '独', '生', '的', '看', '算', '素', '红', '肉', '能', '菜', '華', '蛋', '西', '誌', '請', '质', '进', '這', '韭', '馬',
[illegible]
Contains digits: True
Contains multiple spaces: True
Sample URLs found: ['https://www.malaysiakini.com/news/489216', 'http://miasa.org.my/', 'https://mmha.org.my/', 'https://mmha.org.my/find-help/psychologi
cal-support-services/', 'https://mmha.org.my/contacts/']
```

Figure 4.7: The code to find the noise of the text and the noise find.

To make sure the dataset ready for analysis, the text cleaning was done such as removing the links, mentions, hashtags, digit, and multiple spaces. This step also makes sure that only remain the alphabet and lower down the text make sure the text all in lowercase. The Figure 4.8, show the code to remove all the text noise.

```python
def clean_text(text):

  text = re.sub(r'https?://\S+|www\.\S+', '', text) #remove url
  text = re.sub(r'@\w+', '', text) # remove mentions
  text = re.sub(r'#\w+', '', text) # remove hashtags
  text = re.sub(r'\d+', '', text) # remove digit
  text = re.sub(r'[^a-zA-Z\s]', '', text) # keep only english letter
  text = re.sub(r'\s+', ' ', text) #remove multiple spaces
  text = text.strip() #remove space at beginning and end of string
  text = text.lower()

  return text

mentalhealth_drop['cleaned_text'] = mentalhealth_drop['full_content'].astype(str).apply(clean_text)
```

Figure 4.8: Remove all the text noise.

The last step of data cleaning is normalization text. In this step all the stop words were remove and done the lemmatize for the data. The Figure 4.9 show the code to do the normalization of the data and display the normalized text.

```
def normalize_text(text):

  words = text.split()
  stop_words = set(stopwords.words('english'))
  words = [word for word in words if word not in stop_words]  # Remove stopwords
  lemmatizer = WordNetLemmatizer()
  words = [lemmatizer.lemmatize(word) for word in words]      # Lemmatize

  return ' '.join(words)


mentalhealth_drop['normalized_text'] = mentalhealth_drop['cleaned_text'].astype(str).apply(normalize_text)
```

```
mentalhealth_drop['normalized_text']
```

```
0        malaysian psychiatrist promising career convic...
1        malaysian psychiatrist promising career convic...
2        malaysian psychiatrist promising career convic...
3        malaysian psychiatrist promising career convic...
4        malaysian psychiatrist promising career convic...
                               ...
13718    hidup kena happy rehat jap hilangkan stress ha...
13719    hidup kena happy rehat jap hilangkan stress wo...
13720    hidup kena happy rehat jap hilangkan stress ha...
13721    hidup kena happy rehat jap hilangkan stress bi...
13722    free stress management workshop update space 1...
Name: normalized_text, Length: 13723, dtype: object
```

Figure 4.9: Normalization for the data.

4.5 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is the initial step of the data analysis. EDA help summarized and investigated using statistical graphics and other visualization methods. EDA helps uncover patterns, identify outliers.

```
import pandas as pd

mentalhealth_drop['created_utc'] = pd.to_datetime(mentalhealth_drop['created_utc'], unit='s')

import matplotlib.pyplot as plt

mentalhealth_drop.resample('D', on='created_utc').size().plot()
plt.title("Number of Posts Over Time")
plt.xlabel("Date")
plt.ylabel("Number of Posts")
plt.show()
```

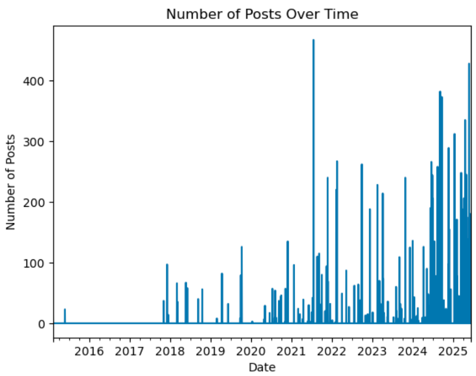

Figure 4.10: The number of the posts over time.

Based on the figure 4.10, the post web scrape dataset has the most post on 2021 which over 400 posts. In 2024 until now, the post number was more than 250 post in each time. The bar chart also shows that the number of posts was increasing in year while before 2018 there was no post in this dataset except in 2015 there were a few posts.

```
mentalhealth_drop['title_length'] = mentalhealth_drop['title'].str.len()
mentalhealth_drop['post_text_length'] = mentalhealth_drop['selftext'].str.len()
mentalhealth_drop['comment_body_length'] = mentalhealth_drop['comment_body'].str.len()

mentalhealth_drop[['title_length', 'post_text_length','comment_body_length']].hist(bins=50, figsize=(10,6))
plt.show()
```

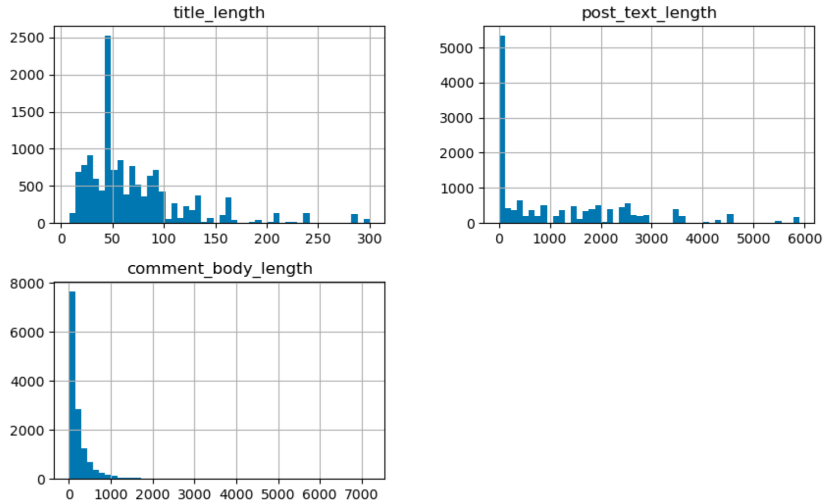

Figure 4.11: Text length for title, post body and comment.

Based on the figure 4.11, it has 3 bar chart that show the text length for the title, post body, and comment body. Form the chart 1 which is the title length chart, show that 2500 of the data have 50 words in their title, the most number words of the in the data was 300 words, and most of the title words was between 10-100 words. Form the post text length chart, it shows that most of the post text have little words like 0-100 words for the post body which it rich more than 5000 post have this situation. Other post body length was between 300 until 3000 words. For the comment body chart, it shows that most of the comment body words length is between 0 – 100 words which almost 8000 posts. Other comment words are between 250 until 1500 words. From the 3 charts, it shows that title words length is the shorter and the post body and comment body words length are longer.

```python
def plot_wordcloud(text, title, colormap='viridis'):
    wordcloud = WordCloud(width=800, height=400, background_color='black', colormap=colormap).generate(text)

    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(title, fontsize=16)
    plt.tight_layout()
    plt.show()
```

```python
title = " ".join(comment for comment in mentalhealth_drop['title'])

plot_wordcloud(title, title="Word Cloud - title", colormap='Blues')
```

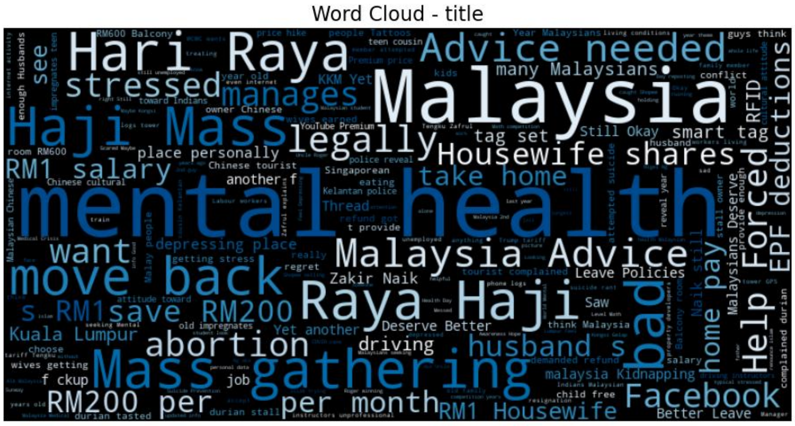

Figure 4.12: Word cloud for title.

To see the most comment words show in the title, post body and the comment, word cloud was used. By using word cloud the words that most frequency detect in the dataset will show bigger. For the title, it shows that most frequence word detect in the Malaysian Reddit post are mental health, Malaysia, Hari Raya, Raya Haji, Mass, gathering and other words. The title word cloud shows in Figure 4.12.

```python
post_text = " ".join(comment for comment in mentalhealth_drop['selftext'])

plot_wordcloud(title, title="Word Cloud - post text", colormap='Purples')
```

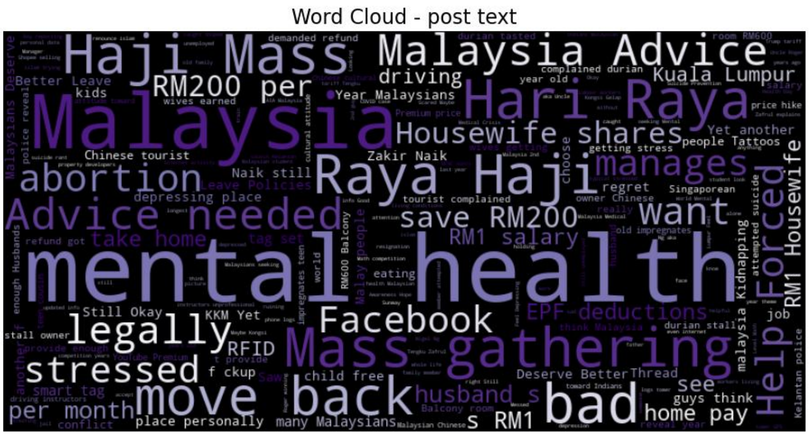

Figure 4.13: Word cloud for post body.

Figure 4.13 show the word cloud for the post body of Malaysian Reddit post. The word cloud shows that, the most frequent show words are mental health, Malaysia, Raya Haji, Mass gathering, Facebook, Hari Raya, legally and more.

```
comment_text = " ".join(comment for comment in mentalhealth_drop['comment_body'])

plot_wordcloud(comment_text, title="Word Cloud - comment text", colormap='Reds')
```

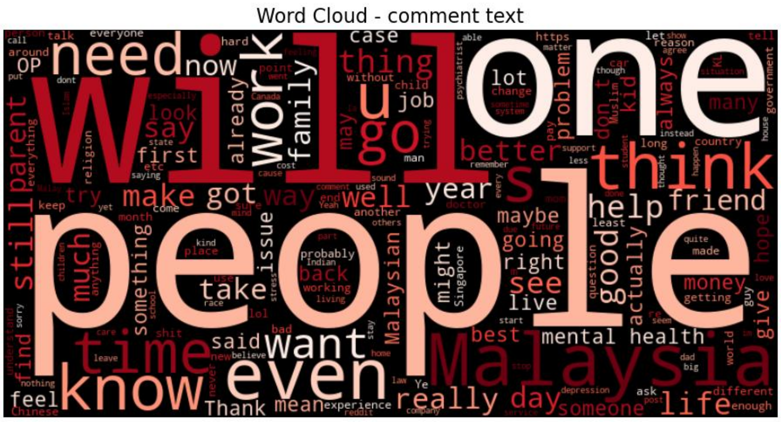

Figure 4.14: Word cloud for comment.

Figure 4.14 show the word cloud for the comment body. From the word cloud, it shows the words like people, will, one, think, Malaysia, time, even and more.

4.6 Emotion Classification Model Training

In this project, it trains the model for emotion classification. The model use was DistilBERT based model. The actual model's name is bhadresh-savani/distilbert-base-uncased-emotion. This is a pretrain model that use for emotion classification. This model can classify 6 emotion such as sadness, joy, love, anger, fear and surprise. This model will give the probabilities for each emotion based on the text given.

Before start train the dataset, the first step is to load the model. Figure 4.15 show that the code to load the model into notebook.

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

tokenizer = AutoTokenizer.from_pretrained("bhadresh-savani/distilbert-base-uncased-emotion")
model = AutoModelForSequenceClassification.from_pretrained("bhadresh-savani/distilbert-base-uncased-emotion")
```

Figure 4.15: Code for load the model into the notebook.

After that, setup the detail of the model to predicts emotions from text. In this model it will predict the probability for each model, but in this project, it will just show the emotion of probability more than 0.3. This is to make sure the emotion predict is more accurate. Figure 4.16 show the code to setup for the emotion prediction.

```
def predict_emotion(text):
    """
    Predicts emotions from text.
    Returns dictionary of emotion: probability if > 0.3
    """
    inputs = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    with torch.no_grad():
        outputs = model(**inputs)

    probs = torch.sigmoid(outputs.logits).cpu().numpy()[0]
    labels = ['anger', 'fear', 'joy', 'love', 'sadness', 'surprise']
    emotion_probs = dict(zip(labels, probs))

    # Return only emotions with confidence > 0.3
    return {emotion: round(float(prob), 3) for emotion, prob in emotion_probs.items() if prob > 0.3}
```

Figure 4.16: Code for setup before emotion prediction.

The figure 4.17 show the outcome of the emotion prediction. From the outcome, the first data have 3 emotions that the probability more than 0.3 which are anger is 0.545, love is 0.828 and sadness is 0.969. Thus, the first data have most probability show emotion sadness.

	normalized_text	emotion
0	malaysian psychiatrist promising career convic...	{'anger': 0.545, 'love': 0.828, 'sadness': 0.969}
1	malaysian psychiatrist promising career convic...	{'anger': 0.309, 'love': 0.945, 'sadness': 0.95}
2	malaysian psychiatrist promising career convic...	{'fear': 0.318, 'love': 0.965, 'sadness': 0.931}
3	malaysian psychiatrist promising career convic...	{'fear': 0.956, 'love': 0.591, 'sadness': 0.386}
4	malaysian psychiatrist promising career convic...	{'anger': 0.431, 'love': 0.917, 'sadness': 0.943}

Figure 4.17: The outcome for the emotion classification model.

The emotion prediction for one post was more than one emotion, thus the emotion that have the highest probability become the top emotion of the post. This will make one data only have one emotion. Figure 4.18 show that the highest probability emotion becomes the top emotion for each post.

```python
def get_top_emotion(emotion_dict):
    """
    Returns the emotion with the highest probability.
    If empty dict (unlikely), returns 'neutral'
    """
    if not emotion_dict:
        return 'neutral'
    return max(emotion_dict, key=emotion_dict.get)
```

```python
mentalhealth_df['top_emotion'] = mentalhealth_df['emotion'].apply(get_top_emotion)
```

```python
mentalhealth_df[['normalized_text', 'emotion','top_emotion']].head()
```

	normalized_text	emotion	top_emotion
0	malaysian psychiatrist promising career convic...	{'anger': 0.545, 'love': 0.828, 'sadness': 0.969}	sadness
1	malaysian psychiatrist promising career convic...	{'anger': 0.309, 'love': 0.945, 'sadness': 0.95}	sadness
2	malaysian psychiatrist promising career convic...	{'fear': 0.318, 'love': 0.965, 'sadness': 0.931}	love
3	malaysian psychiatrist promising career convic...	{'fear': 0.956, 'love': 0.591, 'sadness': 0.386}	fear
4	malaysian psychiatrist promising career convic...	{'anger': 0.431, 'love': 0.917, 'sadness': 0.943}	sadness

Figure 4.18: Show only the highest probability for each data.

4.7 High risk post analysis

To determine the post that have high risk for mental health crises, filter for emotion like sadness and fear was done. This will help to determine what words that have high risk to show mental health crises. In figure 4.19, it shows the post that have the top emotion like sadness and fear.

```python
high_risk_df = df[df['top_emotion'].isin(['sadness', 'fear'])]
high_risk_df.head()
```

utc	comment_id	comment_body	comment_score	comment_awards	comment_created_utc	full_content	cleaned_text	normalized_text	emotion	top_emotion
+09	m0j5twr	Raped. He raped a minor entrusted under his ca...	27	0	1.733404e+09	Malaysian psychiatrist with 'promising career'...	malaysian psychiatrist with promising career c...	malaysian psychiatrist promising career convic...	{'anger': 0.545, 'love': 0.828, 'sadness': 0.969}	sadness
+09	m0hiscs	Apparently this dude is bro of Dr halina wife ...	74	0	1.733371e+09	Malaysian psychiatrist with 'promising career'...	malaysian psychiatrist with promising career c...	malaysian psychiatrist promising career convic...	{'anger': 0.309, 'love': 0.945, 'sadness': 0.95}	sadness
+09	m0hpmsh	> She reportedly said the married Amirul Arif ...	28	0	1.733374e+09	Malaysian psychiatrist with 'promising career'...	malaysian psychiatrist with promising career c...	malaysian psychiatrist promising career convic...	{'fear': 0.956, 'love': 0.591, 'sadness': 0.386}	fear
+09	m0hjkkv	nerakazens are doing their job at x. hehehe ...	12	0	1.733371e+09	Malaysian psychiatrist with 'promising career'...	malaysian psychiatrist with promising career c...	malaysian psychiatrist promising career convic...	{'anger': 0.431, 'love': 0.917, 'sadness': 0.943}	sadness
+09	m0hz13h	On the bright side, he can be a promising inma...	5	0	1.733378e+09	Malaysian psychiatrist with 'promising career'...	malaysian psychiatrist with promising career c...	malaysian psychiatrist promising career convic...	{'fear': 0.983, 'love': 0.394, 'sadness': 0.548}	fear

Figure 4.19: Filter out the data that have high probability in sadness and fear.

In the dataset, it has 6035 posts that can be define as high-risk post. This mean that in the dataset, it more than half of the dataset was define as high-risk post.

```
num_high_risk_posts = high_risk_df['comment_id'].nunique()

print(f"Number of high-risk posts: {num_high_risk_posts}")
```

```
Number of high-risk posts: 6035
```

Figure 4.20: Number of high-risk posts.

In the high-risk posts, in have 2 emotion label. From the bar chart in figure 4.21, it shows that posts that be predict fear emotion is more than sadness.

```
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style='whitegrid')

emotion_counts = high_risk_df['top_emotion'].value_counts()

plt.figure(figsize=(8, 5))
sns.barplot(x=emotion_counts.index, y=emotion_counts.values, palette="viridis")
plt.title('Distribution of High-Risk Emotions in Comments')
plt.xlabel('Emotion')
plt.ylabel('Number of Comments')
plt.show()
```

```
/tmp/ipython-input-12-3168115124.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=emotion_counts.index, y=emotion_counts.values, palette="viridis")
```

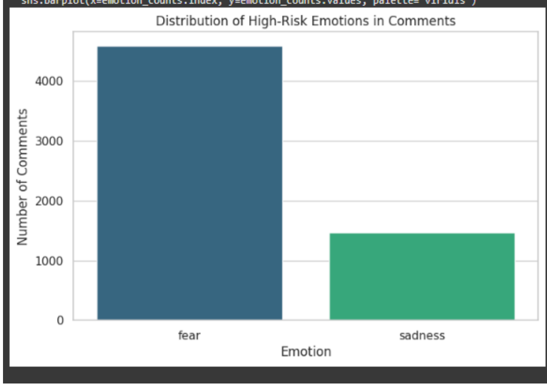

Figure 4.21: Distribution of high-risk emotions.

To determine the common words in the high-risk post, word cloud was used. The word cloud was done for two high-risk emotions: sadness and fear. For the emotion ‘sadness ’the words like stall owner, mother daughter demanded, police, refused pay, pay remaining were the common words show in the high-risk post. The word cloud for sadness-labelled data.

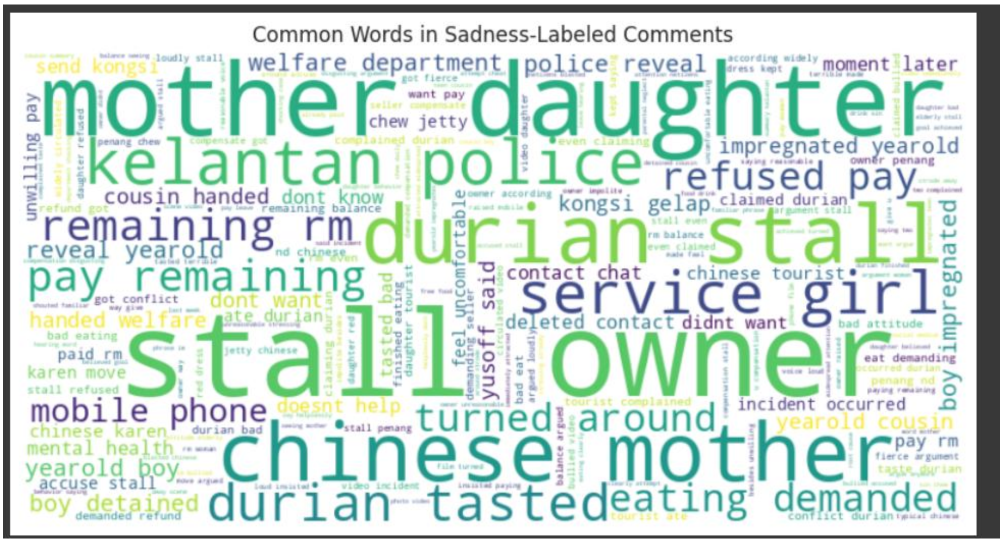

Figure 4.22: Common words for Sadness-labelled data.

For fear emotion post, it shows the words like mental health, due religion, stable job, trying best, want help, trying and other words. The word cloud for fear-labelled data shown in figure 4.23.

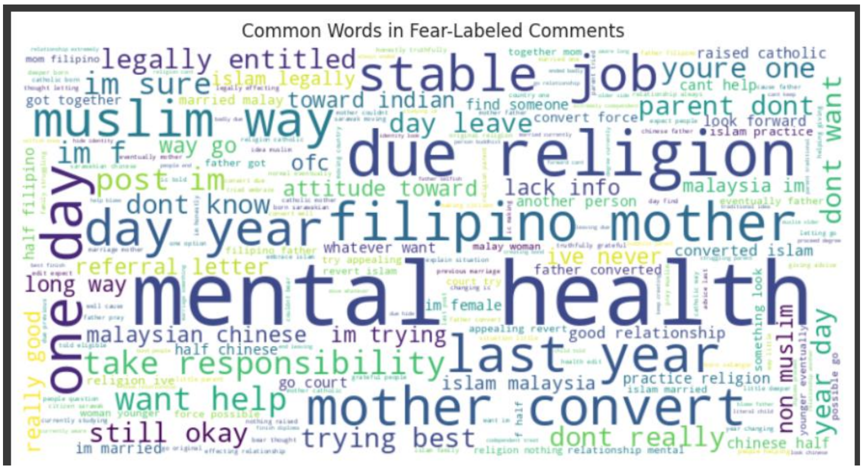

Figure 4.23: Common words for Fear-labelled data.

Based on the common words in word clouds, it has some words that make attention for mental health crises like mental health, demanded, trying best, want help, still okay, and don’t want. These words give the attention for the reader that the writer might face some problem in life and makes the writer have high-risk to have mental health crises.

4.8 XAI interpretation

In this project, the dataset used was scrape from the social media Reddit thus it did not have the label. XAI method LIME was used to explain how the emotion prediction be done for the post. This will to increasing the trustworthy of the prediction outcome. The LIME model will provide a visual breakdown of how the DistilBERT model classified a given text as ‘sadness’ or ‘fear’.

In figure 4.24 it was using XAI to interpret the sadness prediction for the first post. On the left side of the outcome, it shows the emotion prediction probabilities, for this post, the prediction probability for sadness was 0.77 while for love is 0.20 and 0.02 for anger. It shows that this post and comment have the highest probability to have emotion sadness which is high-risk to have mental health crises. In the middle of the outcome, the top 10 words that contributes to the prediction of “sadness” was show. The words have two colours which is purple and min green. The purple colour words is the positive contribution for the “sadness” emotion. In this case, it shows that ‘raped’ have strong positive contribution which is +0.53, other words like ‘entrusted’, ‘minor’, and abusing’ also contribute positively. For the mint green words show the words which have negative contribution. The words like supporters and career have slightly negative contribution (-0.09) and (-0.05) for the emotion prediction.

The model predicts “sadness” because the text contains strong keywords related to sexual abuse, violence and betrayal like ‘raped’, ‘minor’, and ‘entrusted’. These words show strong emotional responses, particularly sadness when combined in this context. The presence of words like supporters and career slightly dilute the overall sadness but do not outweigh the dominant negative cues.

```
exp = explainer.explain_instance(
    text_instance=text_sample,
    classifier_fn=predict_proba,
    num_features=10,
    top_labels=1,
    num_samples=500  # increase for more accuracy
)
```

```
exp.show_in_notebook(text=True)
```

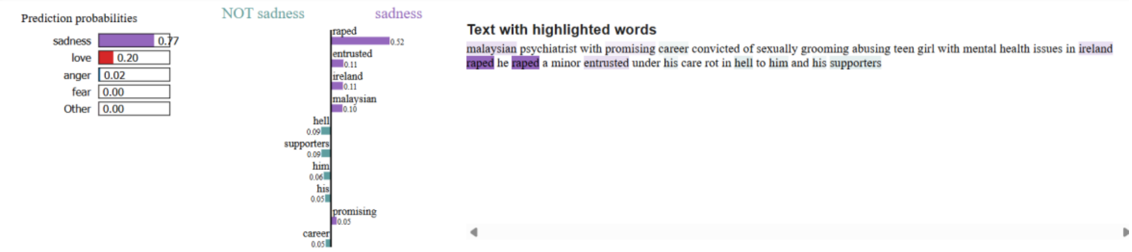

Figure 4.24: The outcome of LIME for the first data.

LIME was tried for another random chosen post. For the LIME in figure 4.25, it was tried for the high-risk post for 2200th post. On the left side of the outcome, it shows the prediction probabilities for each class, it has probability 0.89 for sadness emotion and 0.10 for love emotion. This indicates that the model is highly confident in predicting "sadness" for this text with probability of 89%. On the right side, the input text is displayed with words highlighted in purple and mint green. The purple colour words show the positive contribution for "sadness". The word "alerts" and "landfall" show the strong positive contribution which were 0.23 and 0.10. other words like 'lack', 'cause' also contribute positively. For the word that give slightly negative contribution include 'kiss'.

The model predicts "sadness" because the text contains strong keywords related to natural disasters and tragedy like words "landfall", "lack", and "cause". These words show strong emotional responses particularly sadness when combined in this context. The presence of words like "kiss: slightly dilutes the overall sadness but does not outweigh the dominant negative cues.

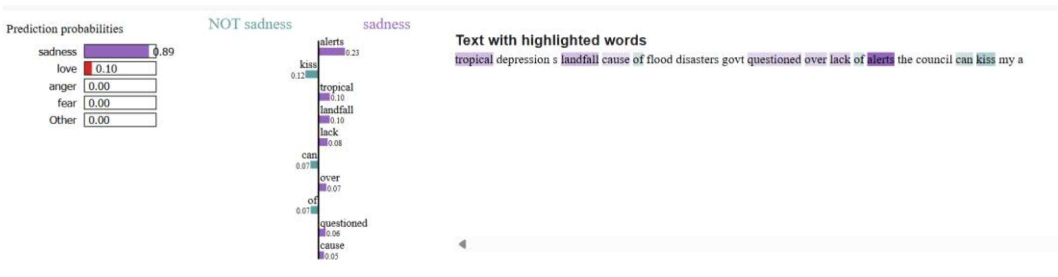

Figure 4.24: The outcome of LIME for the 2200th data.

4.9 Summary

This chapter detailed the end-to-end workflow of collecting and analysing Reddit posts related to mental health from the r/Malaysia subreddit. A total of 13,723 posts with comments were collected using PRAW library. The post was collected by searching for posts that containing keywords such as “mental health”, “depression”, “anxiety”, and “stress”. The dataset was structured into 12 columns capturing the metadata and textual content of the post.

To ensure the quality and usability of the dataset, missing value were handled and some text preprocessing was performed. The text preprocessing included remove the non-alphabetic characters normalizing text, eliminating stop words and lemmatization. These steps ensured that the data was ready for downstream analysis and modelling.

EDA done for trends over time, word frequency patterns and length distribution across title, post body and comments. Word clouds used to highlighted frequently occurring terms in the title, post body and comments. The words like “mental health”, “Malaysia”, “Hari Raya”, and “gathering” were the words that common show in the data collected. This offering the insights into the cultural and contextual themes present in the data,

An emotion classification model based on DistilBERT was used to classify post into six emotions. Posts labelled with sadness and fear were identified as high-risk for mental health crises. Out of the dataset, there were 6,035 posts were classified as high-risk. Word clouds for high-risk emotions showed that words like “refused pay”, “police”, and “demand” link to sadness, while the words like “mental health”, “want help”, and “stable job” were common in fear-labelled posts.

Lastly, LIME was applied to interpret the DistilBERT’s predictions, showing which words contributed most to each emotion classification. This added transparency and helped validate the model’s decision.

In summary, this chapter established a solid foundation for detecting potential mental health concerns using real-world social media data and explainable AI techniques.