Flirt Message Prediction Assignment (NLP)

Table Of Contents (TOC)

1. Creating Flirt Message Prediction Model

a. Identifying The Problem Statement:

Stage 1: Domain Selection- NLP, **Stage 2**: Learning Selection – Supervised, **Stage 3**: Supervised Type – Classification. This problem statement comes under **Supervised Learning - Classification(NLP)**

Approach:

Created a customized csv file with Message & Label as columns & labelled the messages manually whether it falls under flirt message or non-flirt message. Created a dataset with 105 flirt messages & 105 non flirt messages.

Customized csv file name: Flrt & n.flrt msg.csv

b) Research Methodology:

Created combinations of models using classification algorithms like Logistic, KNN, SVM ,DT , Multinomial NB, RF & Passive Aggressive Classifier with Counter Vector & IT-IDF & selected the best model out of all.

c) Final Model:

Best Model was selected based on F1 score & CM.

```
Decision Tree - TfidfVectorizer
Accuracy: 0.7205882352941176
Confusion Matrix:
 [[10 14]
 [ 5 39]]
Classification Report:
                    precision    recall  f1-score   support

    Flirt Message       0.67      0.42      0.51        24
Non Flirt Message       0.74      0.89      0.80        44

         accuracy                           0.72        68
        macro avg       0.70      0.65      0.66        68
     weighted avg       0.71      0.72      0.70        68
```

```
Passive Aggressive Classifier - TfidfVectorizer
Accuracy: 0.6176470588235294
Confusion Matrix:
 [[19  5]
 [21 23]]
Classification Report:
                    precision    recall  f1-score   support

    Flirt Message       0.47      0.79      0.59        24
Non Flirt Message       0.82      0.52      0.64        44

         accuracy                           0.62        68
        macro avg       0.65      0.66      0.62        68
     weighted avg       0.70      0.62      0.62        68
```

Both models seems to perform well. But am choose Decision Tree - CountVectorizer as the best model.

Saved best model file name: Finalized_Model_Random_Forest.sav

For more details on code employment refer IPYNB.

Ipython notebook name : Phase 1-Best Model(CV+TV+Models).ipynb

2. Checking Flirt Message Prediction Model:

Created Flirt Message Prediction Model was checked with WhatsApp chat's each row.

Tested WhatsApp txt file name: VV WhatsApp.txt

Proof:

```
In [11]: 1 df
```

Out[11]:

	Date	Time	Name	Message	Flirt_Prediction
1	11/08/2023	11:36 am	Vasumathi Aug23 வசுமதி	Hi mam	Non Flirt Message
2	11/08/2023	11:39 am	Hope Artificial Intelligence	It is online course	Non Flirt Message
3	11/08/2023	11:46 am	Vasumathi Aug23 வசுமதி	Wil there be any scenario to visit your instit...	Non Flirt Message
4	11/08/2023	11:50 am	Hope Artificial Intelligence	May I know your name please	Non Flirt Message
5	11/08/2023	11:51 am	Vasumathi Aug23 வசுமதி	Vasumathi	Non Flirt Message
...	...	...	...	...	...
283	20/11/2023	7:23 pm	Hope Artificial Intelligence	*Your form response has been received.	Non Flirt Message
284	22/11/2023	9:55 pm	Vasumathi Aug23 வசுமதி	video lecture for word cloud is missing .	Flirt Message
285	24/11/2023	10:19 am	Hope Artificial Intelligence	Will check that Mam	Non Flirt Message
286	25/11/2023	1:48 pm	Vasumathi Aug23 வசுமதி	<Media omitted>	Non Flirt Message
287	25/11/2023	1:53 pm	Vasumathi Aug23 வசுமதி	<Media omitted>	Non Flirt Message

287 rows x 5 columns

WhatsApp Data Analysis:

3) Finding Talkative & Less Talkative:

Proof:

To find Talkative & Less Talkative Person:

In [13]:
```
excluded_name=['Vasu']
filtered_df= df[~df['Name'].isin(excluded_name)]

talkative_person=filtered_df['Name'].value_counts().idxmax()
less_talkative_person= filtered_df['Name'].value_counts().idxmin()

message_counts = filtered_df['Name'].value_counts().reset_index()
message_counts.columns=['Name','Message Count']


print(f'Talkative Person: {talkative_person}')
print(f'Less Talkative Person: {less_talkative_person}')

message_counts

```

```
Talkative Person: Vasumathi Aug23 வசுமதி
Less Talkative Person: Hope Artificial Intelligence
```

Out[13]:

	Name	Message Count
0	Vasumathi Aug23 வசுமதி	144
1	Hope Artificial Intelligence	139

Graph Visualization:

In [14]:
```
import plotly.express as px
fig = px.bar(message_counts, x='Name', y='Message Count',
             text='Message Count',
             title='Number of Messages Sent by Each Person',
             labels={'Message Count': 'Number of Messages', 'Name': 'Name'},
             template='plotly_white', height=600)

# Rotating x-axis labels for better readability
fig.update_layout(xaxis_tickangle=-45)

# Showing the figure
fig.show()
```

4) Finding Most Active Day & Most Active Time:

Proof:

Finding Most Active Day and Most Active Time:

In [15]:
```
# Assuming df is your DataFrame
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

# Add a new column 'Day' to the DataFrame
df['Day'] = df['Date'].dt.day_name()

# Handling Time column with 'hh:mm am/pm' format
def extract_hour_am_pm(time_str):
    try:
        parsed_time = pd.to_datetime(time_str, format='%I:%M %p')
        return parsed_time.strftime('%I:%M %p')
    except (ValueError, AttributeError):
        return None

df['Hour'] = df['Time'].apply(extract_hour_am_pm)

# Filtering out rows with invalid 'Hour' values
df_valid_hours = df.dropna(subset=['Hour'])

# Finding most active day
most_active_day = df_valid_hours['Date'].dt.day_name().value_counts().idxmax()

# Finding most active time
most_active_time = df_valid_hours['Hour'].value_counts().idxmax()

print(f'Most Active Day: {most_active_day}')
print(f'Most Active Time: {most_active_time}')

```

```
Most Active Day: Monday
Most Active Time: 06:33 AM
```

In [16]:
```
df.head()
```

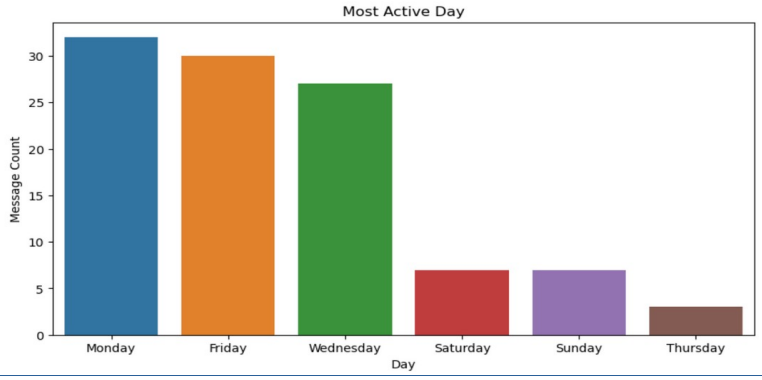
Out[16]:

	Date	Time	Name	Message	Flirt_Prediction	Day	Hour
1	2023-11-08	11:36 am	Vasumathi Aug23 வசுமதி	Hi mam	Non Flirt Message	Wednesday	11:36 AM
2	2023-11-08	11:39 am	Hope Artificial Intelligence	It is online course	Non Flirt Message	Wednesday	11:39 AM
3	2023-11-08	11:46 am	Vasumathi Aug23 வசுமதி	Wil there be any scenario to visit your instit...	Non Flirt Message	Wednesday	11:46 AM
4	2023-11-08	11:50 am	Hope Artificial Intelligence	May I know your name please	Non Flirt Message	Wednesday	11:50 AM
5	2023-11-08	11:51 am	Vasumathi Aug23 வசுமதி	Vasumathi	Non Flirt Message	Wednesday	11:51 AM

Graph Visualization:

In [17]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting the most active day
plt.figure(figsize=(10, 5))
sns.countplot(x=df['Day'], order=df['Day'].value_counts().index)
plt.title('Most Active Day')
plt.xlabel('Day')
plt.ylabel('Message Count')
plt.show()

# Plotting the most active time
plt.figure(figsize=(10, 5))
sns.countplot(x=df['Hour'], order=df['Hour'].value_counts().index)
plt.title('Most Active Time')
plt.xlabel('Hour')
plt.ylabel('Message Count')
plt.show()

```

```
/Users/viswanathanmuthu/anaconda3/envs/AIVE_Course/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead

/Users/viswanathanmuthu/anaconda3/envs/AIVE_Course/lib/python3.11/site-packages/seaborn/_oldcore.py:1498: FutureWarning:

is_categorical_dtype is deprecated and will be removed in a future version. Use isinstance(dtype, CategoricalDtype) instead
```



Most Active Time:

In [18]:

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

# Create a pivot table to get counts for each combination of Day and Hour
heatmap_data = df.pivot_table(index='Day', columns='Hour', aggfunc='size', fill_value=0)

# Create a heatmap
plt.figure(figsize=(14, 8))
sns.heatmap(heatmap_data, cmap='viridis', annot=True, fmt='d', linewidths=.5)
plt.title('Heatmap of Message Count by Day and Hour')
plt.show()
```

5.Finding Media Count Sent by Each Person:

Proof:

Finding Media Count Sent by Each Person:

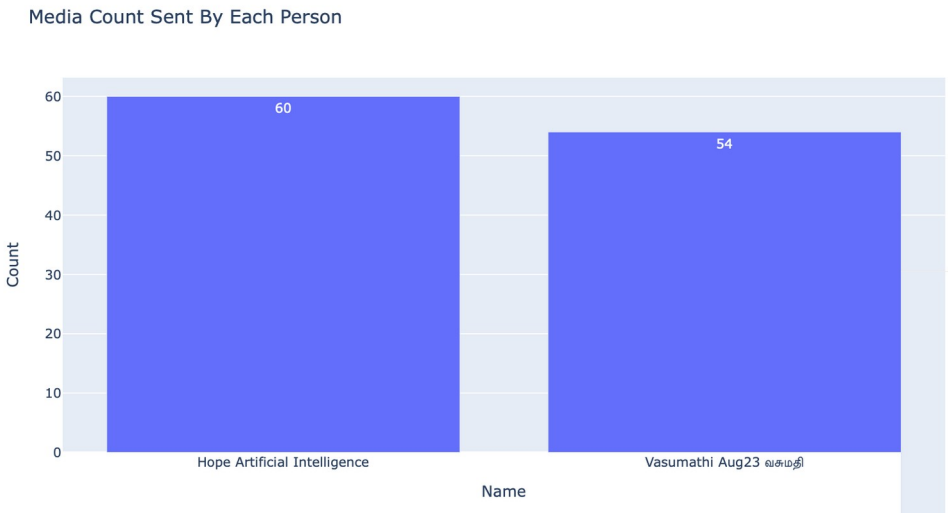
In [19]:
```
excluded_name=['Vasu']
filtered_df= df[~df['Name'].isin(excluded_name)]

media_counts=filtered_df[filtered_df['Message']=='<Media omitted>']['Name'].value_counts()

# Creating a DataFrame from the media counts
media_counts_df = pd.DataFrame({'Name': media_counts.index, 'Media Count': media_counts.values})
media_counts_df
```

Out[19]:

	Name	Media Count
0	Hope Artificial Intelligence	60
1	Vasumathi Aug23 வசுமதி	54

Graph Visualization:

In [20]:
```
fig=px.bar(media_counts_df,x='Name',y='Media Count', text= 'Media Count',
           title = 'Media Count Sent By Each Person', labels={'Media Count':'Count'})
fig.show()
```



6) Finding Missed Call- All Count:

Proof:

Finding Missed Call Count:

In [21]:
```
missedcall_count=df['Message'].str.contains('missed call',case=False).sum()
print(f'Missed Call Count: {missedcall_count}')
```

```
Missed Call Count: 0
```

Word Cloud Visualization

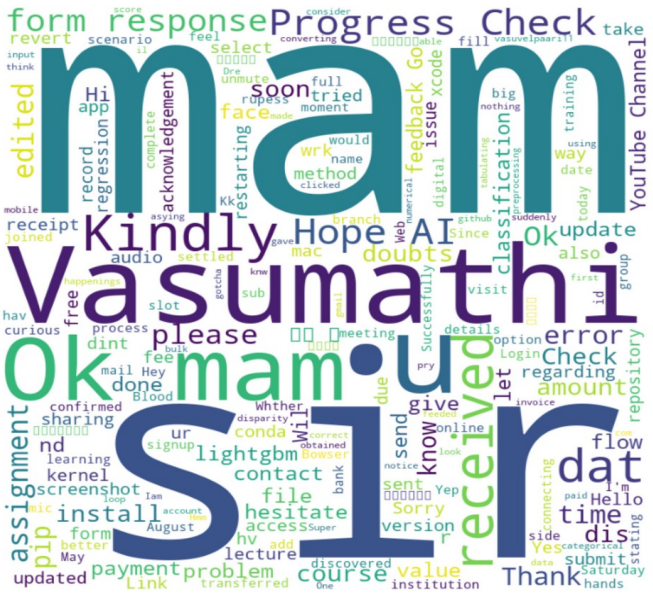
Word Cloud Visualization:

In [22]:
```
dataset=df
from wordcloud import WordCloud
from nltk.corpus import stopwords
import nltk
import matplotlib.pyplot as plt
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/viswanathanmuthu/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[22]: True

In [23]:
```
comment_words = []
stoplist = stopwords.words('english')
stoplist.extend(['omitted', 'voice','missed','call','video','deleted','media','message'])
wordcloudss="This function saves image"
dataset.index=range(dataset.shape[0])
for i in range(1,len(dataset)):
    comment_words.append(dataset['Message'][i])
    vv=" ".join(comment_words)
    wordcloud = WordCloud(width = 800, height = 800,
                          background_color ='white',
                              stopwords = stoplist,
                              min_font_size = 10).generate(vv)
plt.figure(figsize = (9, 7), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.savefig('wordcloud.PNG')
plt.show()
print("Successfully created")
wordcloudss="This function saves image"
```



For more details on code employment from 2 to 6 & wordcloud visualization refer IPYNB.

Ipython notebook name : Phase 2- Flirt Prediction Model Deployment & Analysis.ipynb

----------END OF THE DOCUMENT---------