INTRODUCTION:

In these tasks we will demonstrate how to create hand gesture recognition app in python. While creating we will use three different kinds of classifiers to classify the dataset. Task 3 will introduce you to text mining techniques of machine learning. In this task we will use google API to get reviews of different companies. We will use these reviews to find top ten words used in each companies review. Using these reviews data, we will implement a word cloud Illustration using python.

SCOPE OF THE DOCUMENTATION:

This document will provide overview of procedures followed and the tools used. The procedure discusses on how to create gesture recognition app. It will also introduce you to the text mining techniques and API creation

Methods:

Part 1

We load & manipulate the data using Panda's package.

Before building the models, we need to separate the target variable that is the sign from the independent variables, the pixels. we use iloc to select the cells we want.

```
X, y = data.iloc[:, 1:], data.iloc[:, 0].to_frame()
```

This application will have three different models that are:

Logistic regression
Support vector machine
Random forest classifier

We will start by fitting the models on the train dataset then test its performance on the test dataset (33% test size)

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42)
```

For each model we use the following methods:
Fit: to estimate the optimal model parameters
Score: to evaluate the model accuracy

1. import the pandas

```
import pandas as pd
```

2. loaded the dataset

```
data = pd.read_csv('/content/sign_mnist (2).csv')
data.head()
```

3. After that downloaded the packages for training the dataset

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

4. Applying random forest classifier

```
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)
```

```
rfc.score(X_test, y_test)
```

5. Applied logistic regression

```
lr = LogisticRegression()
lr.fit(X_train, y_train)
```

```
round(lr.score(X_test, y_test), 2)
```

6. Applying Support vector

```
svc = SVC()
svc.fit(X_train, y_train)
svc.score(X_test, y_test)
```

7. applied support vector machine and using the svc predict method to predict the images.

```
svc.predict(X_test.iloc[20:32,:])
```

Part 2

Step 1 is to load the data using pandas

Step 2 is to import the libraries namely,

- import requests
- import json
- import time
- import matplotlib.pyplot as plt
- import pandas as pd
- import numpy as np

- import collections

```
# Importing libraries
import requests
import json
import time
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import collections
```

- Step3 is to working on Google Place Reviews from Googlemap Api

```
# Importing libraries to make word cloud
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
import collections
```

```
# Combining all reviews of google place api
Total_reviews = [reviews_list,reviews_list1,reviews_list2,reviews_list3]
My_Reviews = []
for i in range(len(Total_reviews)):
  My_Reviews.extend(Total_reviews[i])


Reviews = pd.DataFrame(My_Reviews,columns=['Food_Reviews'])
```

```
Reviews.head()
```

	Food_Reviews
0	Great service, friendly customer service, food...
1	So Stephanie was our waitress and aside from t...
2	Dinner and drinks. Food is a bit pricey but is...
3	We aren't from NYC but we loved the Hard Rock ...
4	There food is delicious and also the staff is ...

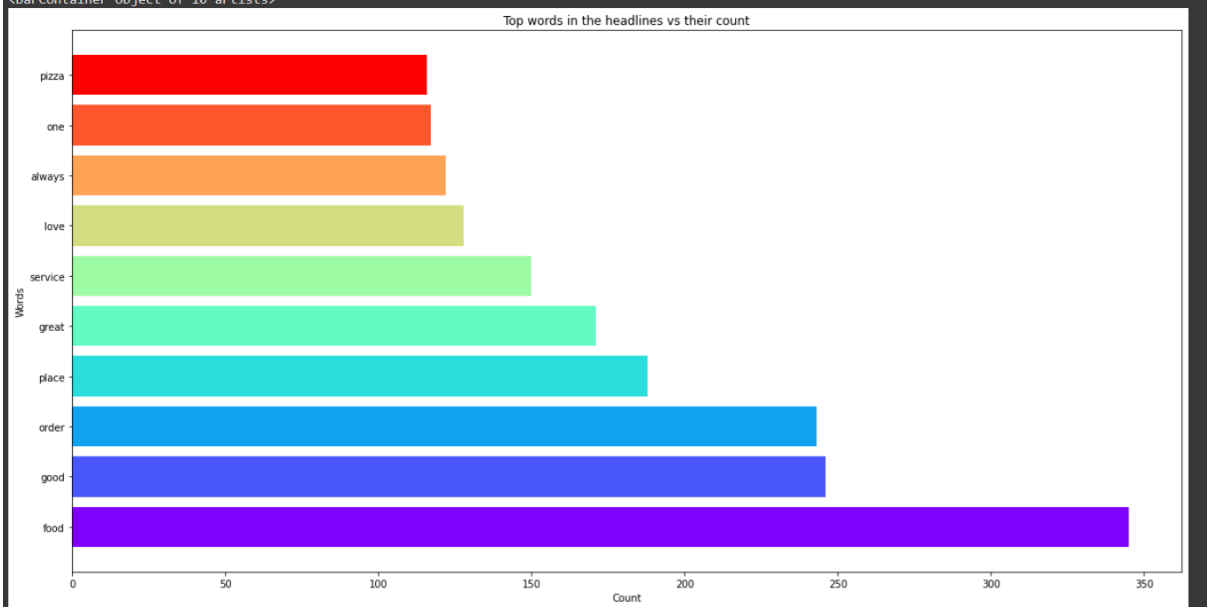
```
Reviews.tail()
```

	Food_Reviews
1138	They have everything you need and the\nThe cos...
1139	This is the place to go for all your BBQ needs...
1140	AMAZING! Hired this crew for my Senior class o...
1141	Clean place and nice staff
1142	Looooove this place. So convenient for gatheri...

-

```
colors = plt.cm.rainbow(np.linspace(0, 1, 10))
plt.rcParams['figure.figsize'] = 20, 10
plt.title('Top words in the headlines vs their count')
plt.xlabel('Count')
plt.ylabel('Words')
plt.barh(words, counts, color=colors)
```

```
<BarContainer object of 10 artists>
```



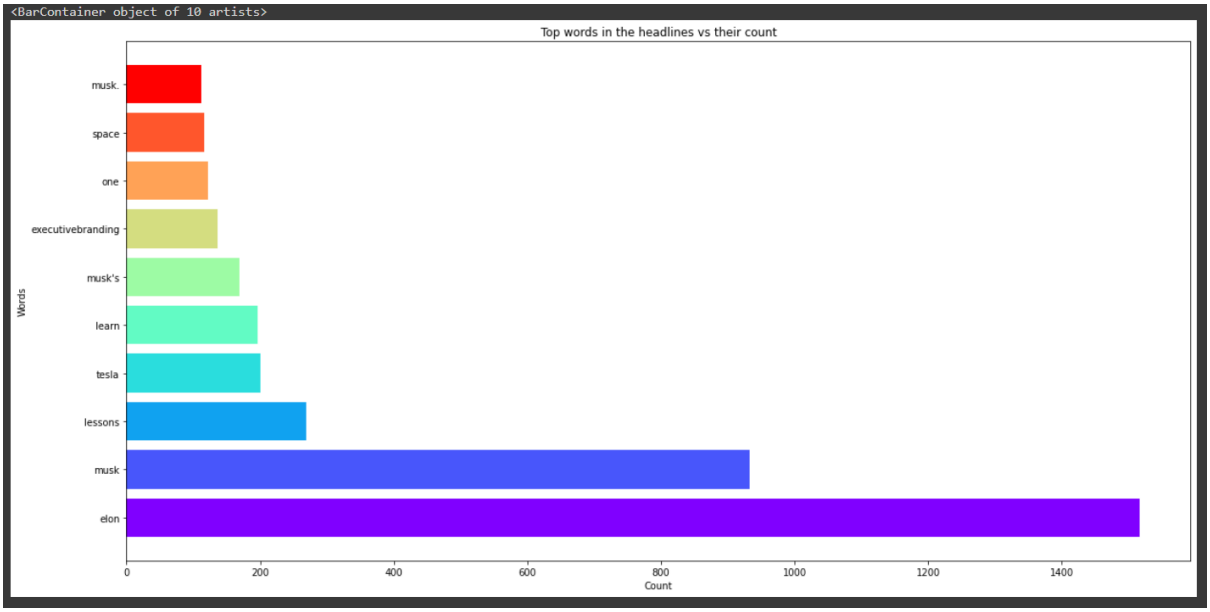
Reviews from Twitter API

```
df.head(5)
```

	Date	User	Tweet
0	2019-12-31 23:45:00+00:00	Candywoodkey	3 Executive-Branding Lessons we can Learn from...
1	2019-12-31 21:20:20+00:00	thirdrowtesla	NEW LEAKED EMAIL FROM ELON MUSK: \n\n"If we ...
2	2019-12-31 21:02:50+00:00	MaraWilson	@1followernodad I think about this article fro...
3	2019-12-31 19:31:34+00:00	Ministraitor	@LargeCardinal Maybe they should get some bitc...
4	2019-12-31 17:20:00+00:00	CHAPINCITO	I like this from Elon Musk, some times you nee...

```
df.tail()
```

	Date	User	Tweet
1495	2019-09-23 08:03:40+00:00	igo2fix	Fake Bitcoin Investment Platform From 'Elon Mu...
1496	2019-09-23 08:00:17+00:00	bitsmart_io	Fake Bitcoin Investment Platform From 'Elon Mu...
1497	2019-09-23 07:57:09+00:00	9figurefortune	Fake Bitcoin Investment Platform From 'Elon Mu...
1498	2019-09-23 07:53:10+00:00	bitlyfool	Fake Bitcoin Investment Platform From 'Elon Mu...
1499	2019-09-23 07:50:50+00:00	bitbrokersinc	Fake Bitcoin Investment Platform From 'Elon Mu...



Reviews from Amazon Api

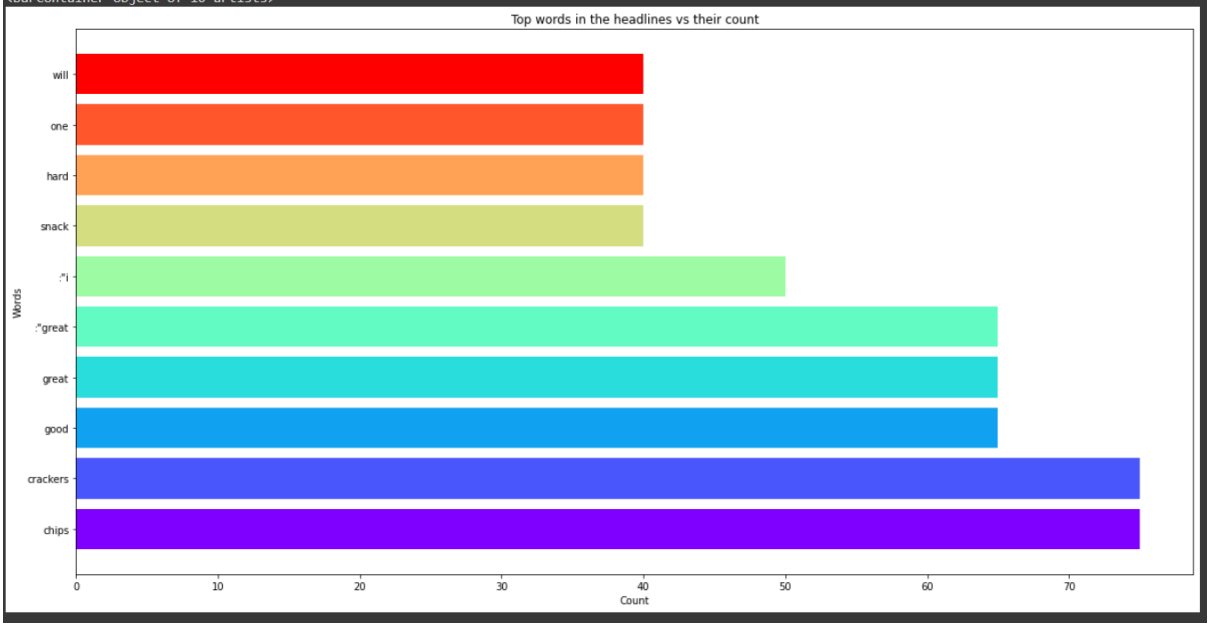
```
pd.DataFrame(Total_Review5[i] for i in range(len(list1)))
```

	0
0	"review":"This is good for kids' lunches and t...
1	"review":"I loved these. The crackers are fres...
2	"review":"Very good deal. Fresh and good tast...
3	"review":"Taste good, but received a lot of cr...
4	"review":"Would take pictures but done gone,v...
5	"review":"Inexpensive but good quality perfect...
6	"review":"Bought these for travel snacks and w...
7	"review":"Needs more cheese crackers","review...
8	"review":"Bought this to send in military care...
9	"review":"Amazon delivered on a Sunday to a M-...
10	"review":"Family loved theses crackers , did n...
11	"review":"I bought these for my granddaughter ...
12	"review":"Best flavors and freshness.","review...
13	"review":"I buy these as on-bike snacks when c...
14	"review":"These are so easy to grab n go!! Th...
15	"review":"Tasted good as I expected, thanks, ...
16	"review":"good snack for lunches at a good pri...
17	"review":"Great selection, super fresh, perfec...
18	"review":"I loves those because of individual pa...

```
comment_words = ''
stopwords = set(STOPWORDS)

# iterate through the csv file
for val in Reviews_f['Reviews']:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    comment_words += " ".join(tokens)+" "

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)
filtered_words = [word for word in comment_words.replace('"review"','').replace('reviews','').split() if word not in stopwords]
counted_words = collections.Counter(filtered_words)

words = []
counts = []
for letter, count in counted_words.most_common(10):
    words.append(letter)
    counts.append(count)
```

```
import numpy as np
colors = plt.cm.rainbow(np.linspace(0, 1, 10))
plt.rcParams['figure.figsize'] = 20, 10

plt.title('Top words in the headlines vs their count')
plt.xlabel('Count')
plt.ylabel('Words')
plt.barh(words, counts, color=colors)
```

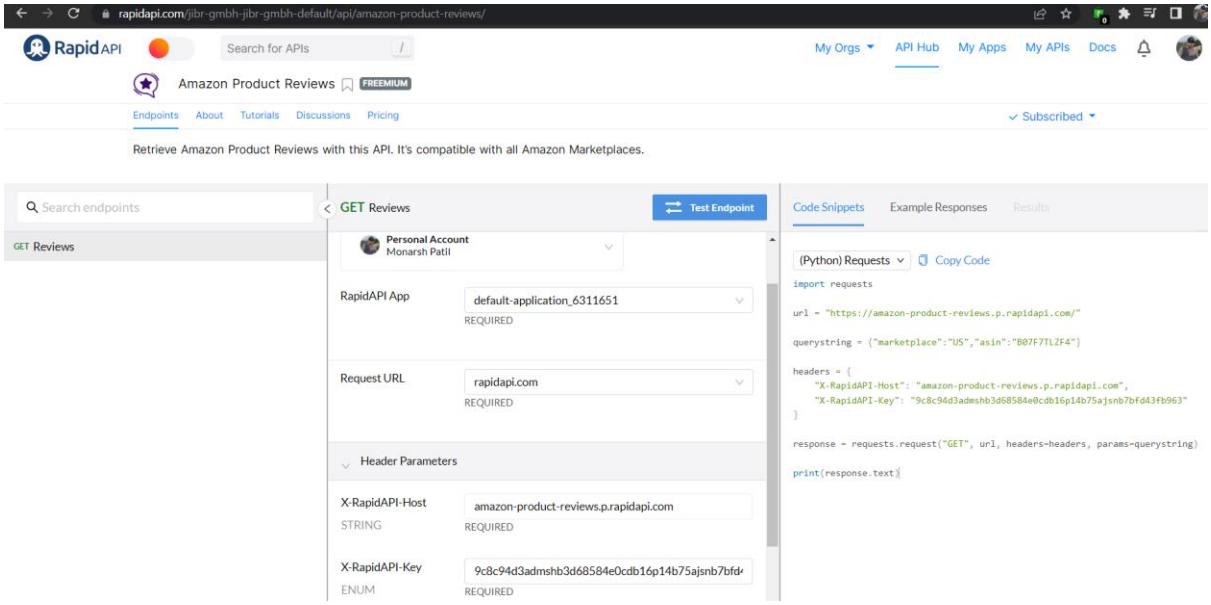
```
<BarContainer object of 10 artists>
```



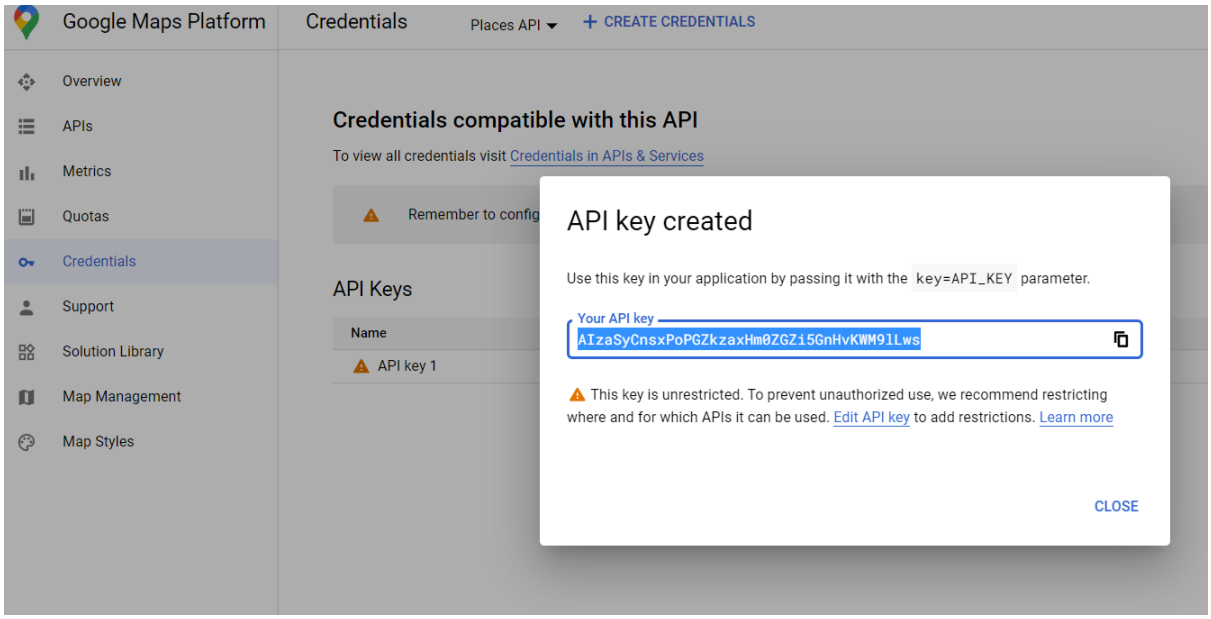
API creation:

We created API's on two different websites one for amazon reviews and other for google places.

AMAZON RAPID API



GOOGLE MAPS API KEY



Results Task1:

Task 1: - Overall all the models are classifying the images on unseen data accurately with a score above 90%.
The support vector machine is doing much better with a score of 98%.

In order to predict an image, we use the predict method as follows

```
1 svc.predict(X_test.iloc[20:32,:])
```

```
array([21, 16, 23, 21,  7, 23,  4,  8,  3,  5, 18,  4])
```

This chunk of code is used to predict rows 21 to 32 signs.

Results Task2:

Task 2: - We used text mining technique to identify top 10 most used words in each company Review(downloaded). Once we have the top 10 most used words we have represented our findings in word cloud Illustration for each company. In this assignment we have chose 3 different companies namely, Google, Twitter, Amazon.
Following are the screenshots of the results we got.

Using text mining technique, identify the top 10 MOST USED WORDS in each company Reviews that you have downloaded. Represent your findings in Word Cloud Illustration for each company like the following example. The size of the words gets bigger as they become more frequently used.

Google:

```
words
```

```
['food',
 'good',
 'order',
 'place',
 'great',
 'service',
 'love',
 'always',
 'one',
 'pizza']
```

```
# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Twitter:

```
words
```

```
['food',
 'good',
 'order',
 'place',
 'great',
 'service',
 'love',
 'always',
 'one',
 'pizza']
```

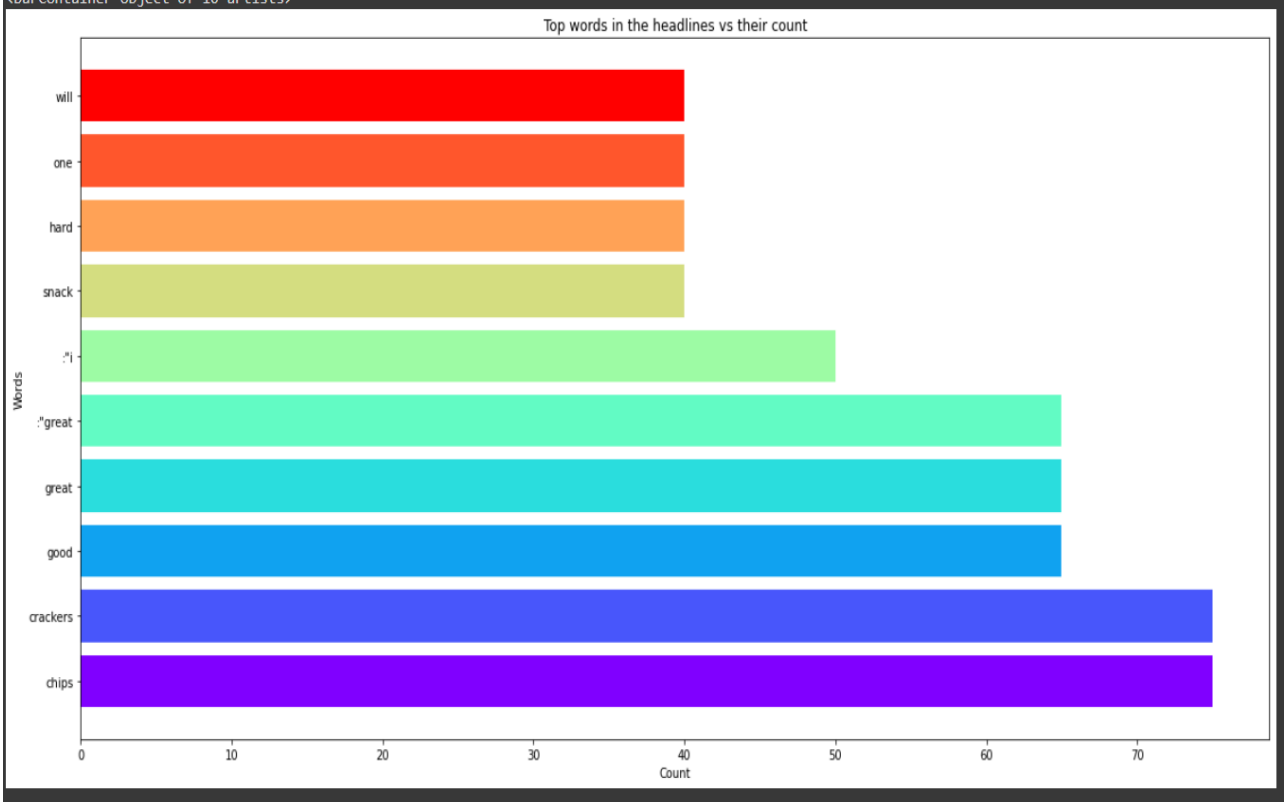
```
for letter, count in counted_words.most_common(10):
    words.append(letter)
    counts.append(count)


# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Amazon:

```
# plot the WordCloud image
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()
```

```
import numpy as np
colors = plt.cm.rainbow(np.linspace(0, 1, 10))
plt.rcParams['figure.figsize'] = 20, 10

plt.title('Top words in the headlines vs their count')
plt.xlabel('Count')
plt.ylabel('Words')
plt.barh(words, counts, color=colors)
```

```
<BarContainer object of 10 artists>
```



Reflection on Learning:

- In this assignment I got a hands-on experience on the Basic concept of machine learning. This assignment helped us understand concepts like Logistic regression, Support vector machine, Random Forest classifier from task 1. Task 2 made us understand concepts like Word cloud, API key generation.

Individual contributions:

Task2

Monarsh Patil

Task 2 - Contributed to separating the target variable that is the sign from independent variables. Applied logistic and parts of support vector machine models on the data. Trained the dataset. After the dataset was filtered & trained used, the svc.predict method to predict an image.

1. import the pandas
2. loaded the csv file

3. After that downloaded the packages for training the dataset

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

4. splitting the dataset

```
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.33, random_state=42)
```

5. Applied logistic regression

```
lr = LogisticRegression()
lr.fit(X_train, y_train)

round(lr.score(X_test, y_test), 2)
```

Fiaz Ali Khan

Task2:- Contributed in creating the technical documentation related to task 1 and applied support vector Machine model on the data.

1. import the libraries
2. After that downloaded the packages for training the dataset
(from sklearn.svm import SVC)
3. Use svc.predict method to predict the images

Esra Tokgoz

Task 2:- Contributed in separating the target variable (i.e. the signs from independent variables). Applied Random Forest Classifier on dataset and trained the dataset. Assisted in Appling support vector machine as well.

1. import the pandas
2. load the csv file
3. After that downloaded the packages for training the dataset

```
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
```

4. Applying random forest classifier

```
rfc = RandomForestClassifier()
rfc.fit(X_train, y_train)

rfc.score(X_test, y_test)
```

TASK 3

MONRASH PATIL

Task 3 – Done the parts of documentation. Created account on google cloud for API key on google maps API by referring the document on DBS Moodle page. Worked on creating WordCloud illustration and Bar-plots for Google and Amazon API's.

Following are steps for creating google API key.

1. Creating account on google cloud
2. After create a new project
3. Go to APIs & Services
4. Go to library and search for Google Place API
5. Go to Credentials and create one
6. Click on API key
7. API created

Following steps are used for creating word cloud (google API).

1. Define class for getting all info about types with specific radius
2. Then use call function for google map API
3. Getting info about coordinates having different types
4. Specifying info we want to get from places
5. Retrieving info from places and combining all reviews.
6. Importing libraries and defining loops to get rid of stopwords.
7. Plotting the word cloud image

FAIZ Ali Khan

Task3: - Contributed in retrieving information from the google places. Done the part of Documentation in task 3. Created rapid API Key on amazon review. Listed the top ten words used in amazon reviews.

Following are steps for creating google API key.

1. Creating account on amazon rapidapi
2. Search for Amazon Product Review
3. Go to pricing and select basic plan
4. Go to endpoint
5. Copy the code in (python)Request format
6. Paste on the python file working on
7. API is created

Following steps are used for creating word cloud (Amazon API)

1. Import request from amazon product review
2. Apply pre-processing and import the libraries
3. Process the data than Iterate through CSV file
4. Split the value
5. Convert each token into lowercase

6. Plot the Wordcloud image
7. Import numpy as np and implement bar_plot

ESRA TOKGOZ

Task3: - Done the parts of documentation in task 3. Worked on Creating cloud illustration and bar plots for twitter API.
Following are the steps to create cloud illustration

1.In that first step install pip snscape.
2.Then getting the tweets from Elon musk for limited time span.
3. Display the data head and data tail. Then typecasting each Val to string.
4. Then splitting the data and converting each token into lowercase.
5. Writing the syntax for word cloud. Plotting the word cloud image

References:

- https://rapidapi.com/jibr-gmbh-jibr-gmbh-default/api/amazon-product-reviews/
- https://console.cloud.google.com/google/maps-apis/credentials?project=machinelearningg-project100522