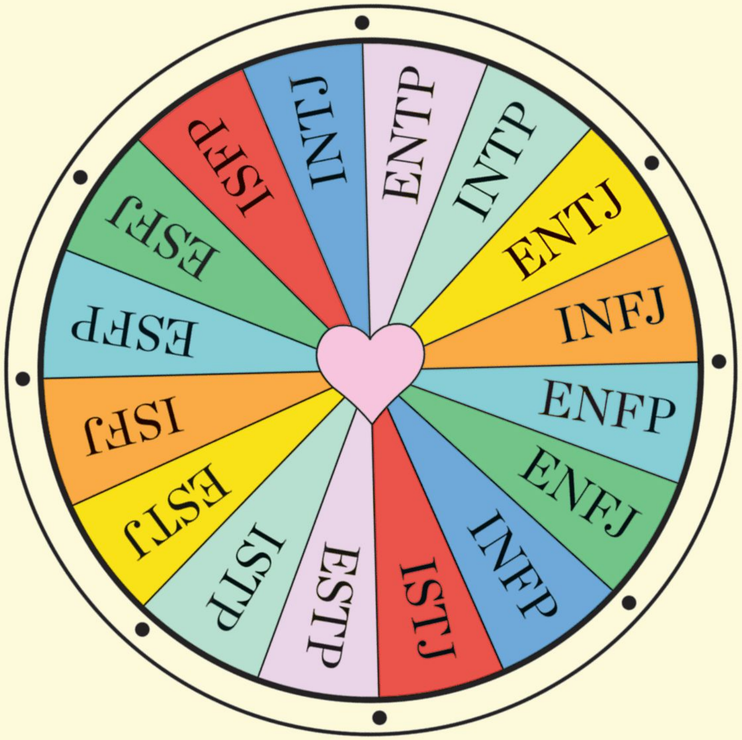


Personality Analysis

Myers-Briggs Type Indicator

Participants
Christopher Guilcapi
Eric Wyluda
Jose Monagas
Katiusca Quijada

01

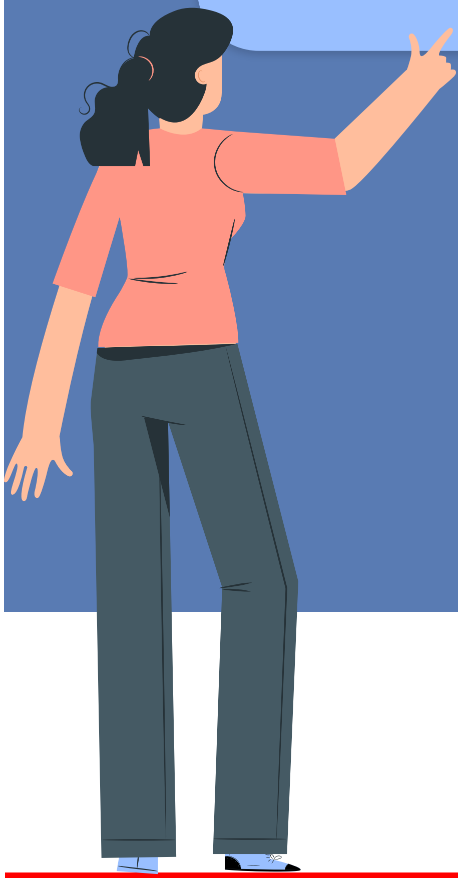
INTRODUCTION

02

Data ETL Approach

03

ML Approach



04

Dashboard

05

Analysis

INTRODUCTION TO MYERS-BRIGGS PERSONALITY TYPE PROBLEM SET

Kaggle Twitter dataset of 200 users with the last 50 tweets from each account collected along with their Myers-Briggs Personality Type.

Purpose of Myers-Briggs Type Indicator is to make the theory of psychological types understandable and useful in people's lives. Seemingly random variation in behavior is actually quite consistent and there are differences in the ways individuals prefer to use their perception and judgment.

Examine provided data

In [3]:
```python
df.head(5)
```

Out[3]:

<table>
<tr><th></th><th>type</th><th>posts</th></tr>
<tr><td>0</td><td>INFJ</td><td>'http://www.youtube.com/watch?v=qsXHcwe3krw|||http://41.media.tumblr.com/tumblr_lfouy03PMA1qa1rooo1_500.jpg|||enfp and intj moments https://www.youtube.com/watc...</td></tr>
<tr><td>1</td><td>ENTP</td><td>'I'm finding the lack of me in these posts very alarming.|||Sex can be boring if it's in the same position often. For example me and my girlfriend are currently ...</td></tr>
<tr><td>2</td><td>INTP</td><td>'Good one _____ https://www.youtube.com/watch?v=fHiGbolFFGw|||Of course, to which I say I know; that's my blessing and my curse.|||Does being absolutely posit...</td></tr>
<tr><td>3</td><td>INTJ</td><td>'Dear INTP, I enjoyed our conversation the other day. Esoteric gabbing about the nature of the universe and the idea that every rule and social code being arb...</td></tr>
<tr><td>4</td><td>ENTJ</td><td>'You're fired.|||That's another silly misconception. That approaching is logically is going to be the key to unlocking whatever it is you think you are entitled ...</td></tr>
</table>

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8675 entries, 0 to 8674
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   type    8675 non-null   object
 1   posts   8675 non-null   object
dtypes: object(2)
memory usage: 135.7+ KB
```

STEP 1: SPLIT POSTS INTO ROWS

```python
# Split grouped posts into indiv. rows
def extract(posts, new_posts):
    for post in posts[1].split("|||"):
        new_posts.append((posts[0], post))

posts = []
df.apply(lambda x: extract(x, posts), axis=1)
print("Number of users", len(df))
print("Number of posts", len(posts))

df = pd.DataFrame(posts, columns=["type", "posts"])
```

```
Number of users 8675
Number of posts 422845
```

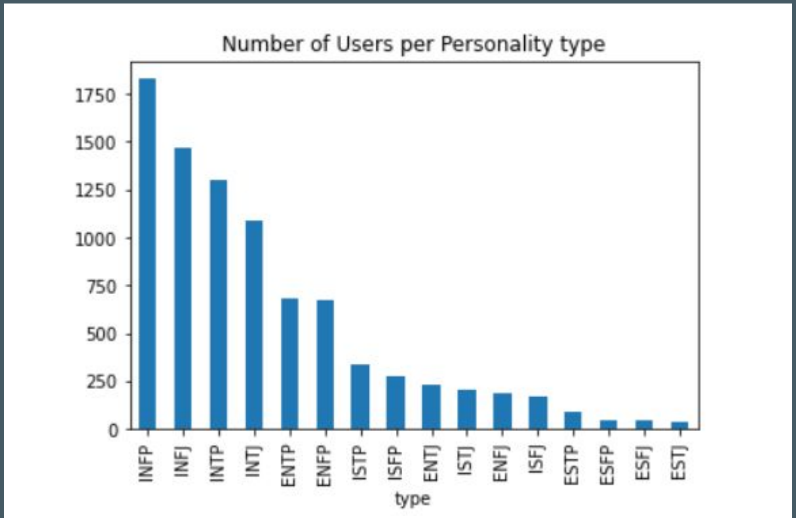
	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw
1	INFJ	http://41.media.tumblr.com/tumblr_lfouy03PMA1qa1rooo1_500.jpg
2	INFJ	enfp and intj moments https://www.youtube.com/watch?v=iz7lE1g4XM4 sportscenter not top ten plays https://www.youtube.com/watch?v=uCdfze1etec pranks
3	INFJ	What has been the most life-changing experience in your life?
4	INFJ	http://www.youtube.com/watch?v=vXZeYwwRDw8 http://www.youtube.com/watch?v=u8ejam5DP3E On repeat for most of today.
...	...	...
422840	INFP	I was going to close my facebook a few months back, but as well as wanting to be able to message my family in ausse and school friends i found that i had connect...
422841	INFP	30 Seconds to Mars - All of my collections. It seems to be fitting my mood right now.
422842	INFP	I have seen it, and i agree. I did actually think that the first time I watched the movie, and from the beginning (or when they got their powers) I kinda thought...
422843	INFP	Ok so i have just watched Underworld 4 (Awakening) and must say it was a really good film, Compared to the other films out in the last few months anyway. I don't...
422844	INFP	I would never want to turn off my emotions. sometimes I hide them from the world, but I still need them for me.'

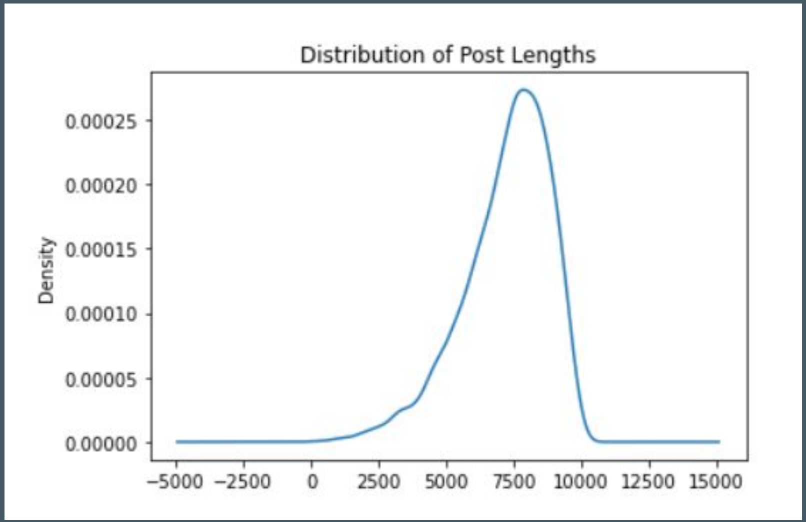
Data cleaning function

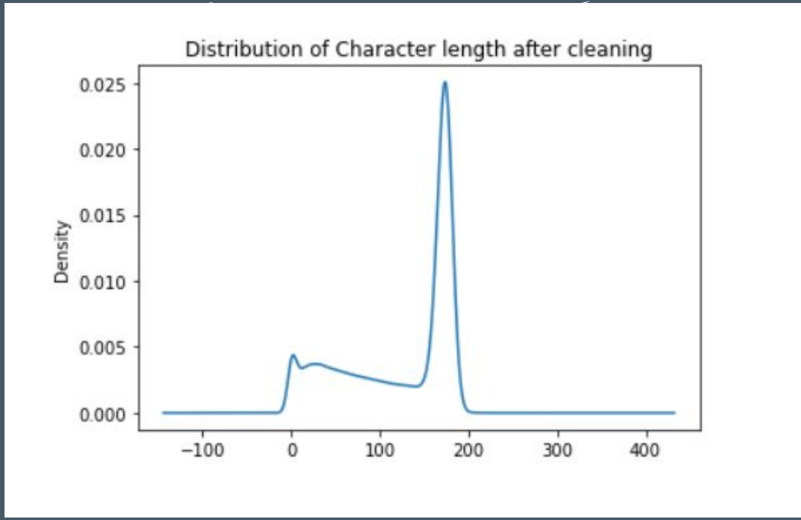
```python
def preprocess_text(df, mbpt_token=True):
    # Remove url links
    df["posts"] = df["posts"].apply(lambda x: re.sub(r'https?:\/\/.*?[\s+]', '', x.replace("|"," ") + " "))

    # Strip misc punctuation
    df["posts"] = df["posts"].apply(lambda x: re.sub(r'[^\w\s]','',x))

    # Remove Non-words
    df["posts"] = df["posts"].apply(lambda x: re.sub(r'[^a-zA-Z\s]','',x))

    # Remove multiple letter repeating words
    df["posts"] = df["posts"].apply(lambda x: re.sub(r'([a-z])\1{2,}[\s|\w]*','',x))

    # Remove short/long words
    df["posts"] = df["posts"].apply(lambda x: re.sub(r'(\b\w{0,2})?\b','',x))
    df["posts"] = df["posts"].apply(lambda x: re.sub(r'(\b\w{30,1000})?\b','',x))

    # Remove Personality Type identifiers/tokens from posts
    # MBPT identifier is subsituted with 'PtypeToken' to avoid bias when training model
    if mbpt_token:
        pers_types = ['INFP' ,'INFJ', 'INTP', 'INTJ', 'ENTP', 'ENFP', 'ISTP' ,'ISFP' ,'ENTJ', 'ISTJ','ENFJ', 'ISFJ' ,'ES
        pers_types = [p.lower() for p in pers_types]
        p = re.compile("(" + "|".join(pers_types) + ")")

    df["posts"] = df["posts"].apply(lambda x: p.sub(' PtypeToken ',x))
    return df
```

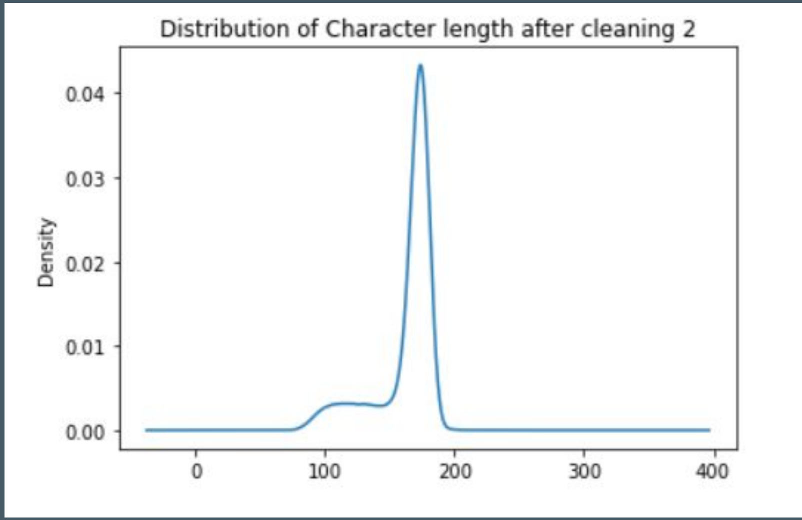
	type	posts
0	INFJ	
1	INFJ	
2	INFJ	PtypeToken and PtypeToken moments sportscenter not top ten plays pranks
3	INFJ	What has been the most lifechanging experience your life
4	INFJ	repeat for most today
...	...	...
422840	INFP	was going close facebook few months back but well wanting able message family ausse and school friends found that had connected few other websites
422841	INFP	Seconds Mars All collections seems fitting mood right now
422842	INFP	have seen and agree did actually think that the first time watched the movie and from the beginning when they got their powers kinda thought Andrew would ...
422843	INFP	have just watched Underworld Awakening and must say was really good film Compared the other films out the last few months anyway dont think was good ...
422844	INFP	would never want turn off emotions sometimes hide them from the world but still need them for

Distribution of data

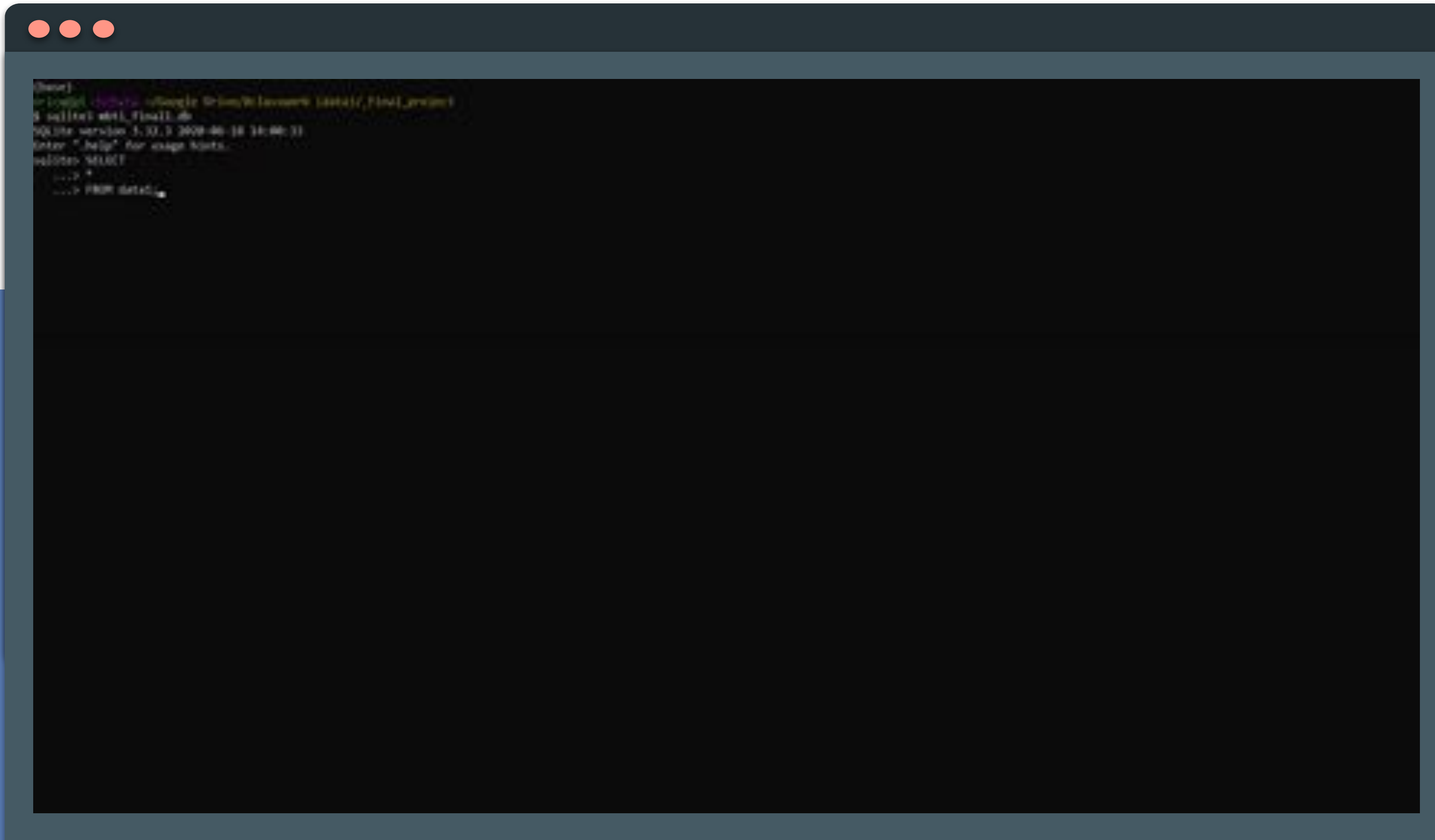






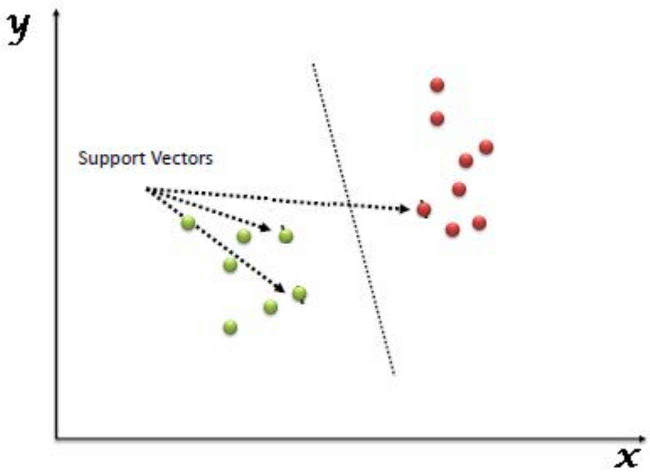


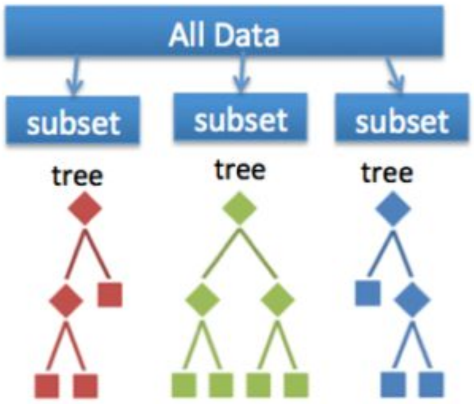
SQLite Database

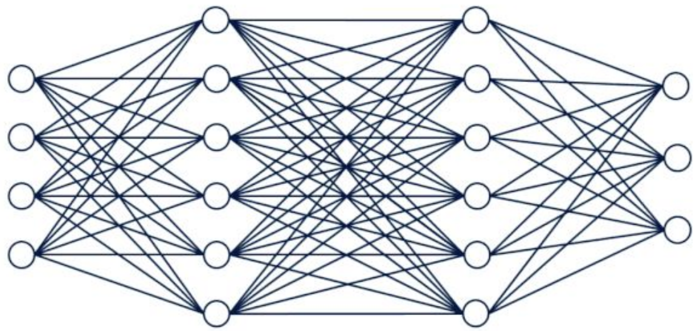


ML APPROACH

We explored three modeling approaches based on their suitability for NLP problems.







SVM: Supervised learning models with associated learning algorithms that analyze data for classification.

Random Forest: Large number of individual decision trees that operate together. Each tree has a class prediction and the one with most votes becomes our model's prediction.

Neural Network: Multitude of simple processing nodes that are highly interconnected and send data through these network connections to estimate a target variable.

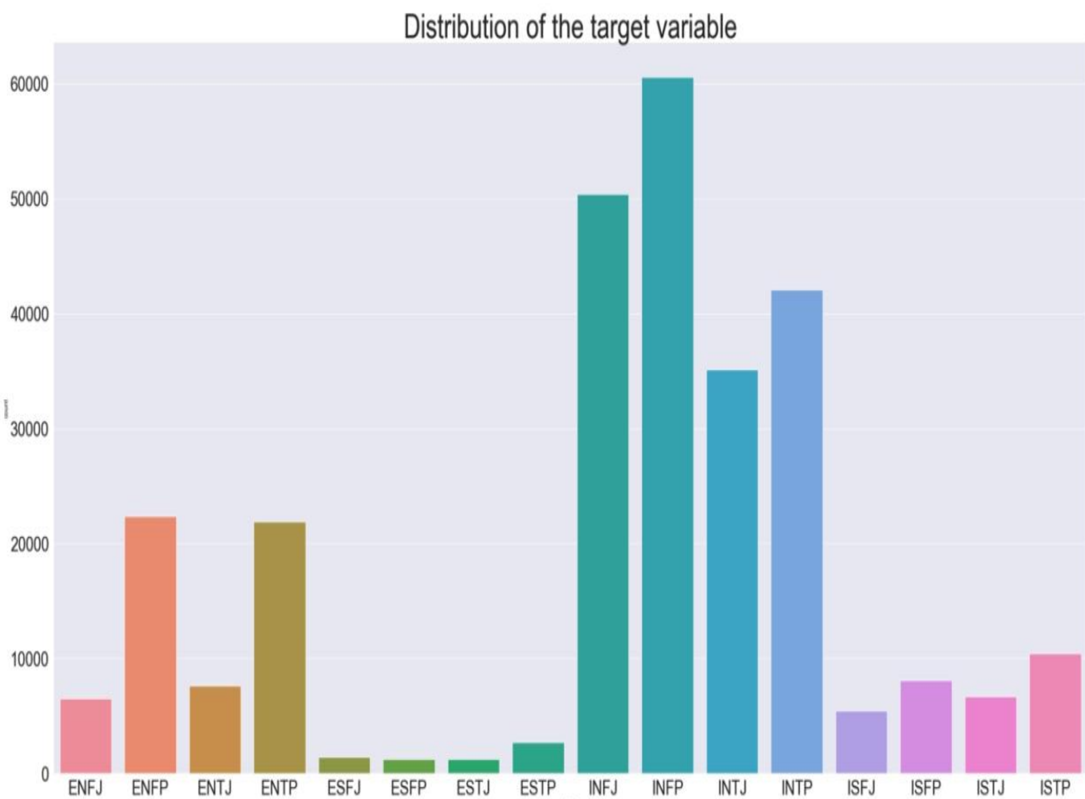
SVM

We applied SVM because of its reputation for success in text classification.

1) We split the data into training and testing sets
2) Then created a data pipeline with SGDClassifier.
3) Trained the model and made predictions.
4) Generated confusion matrix, classification report and heatmaps for better visualization

SVM APPROACH

Distribution of the data



INFP	60583
INFJ	50417
INTP	42071
INTJ	35109
ENFP	22361
ENTP	21922
ISTP	10404
ISFP	8089
ENTJ	7597
ISTJ	6679
ENFJ	6481
ISFJ	5435
ESTP	2697
ESFJ	1441
ESFP	1256
ESTJ	1250

The data is unbalanced

SVM RESULTS

Classification Report

	precision	recall	f1-score	support
ENFJ	0.11	0.09	0.10	1620
ENFP	0.20	0.15	0.17	5590
ENTJ	0.09	0.08	0.09	1899
ENTP	0.19	0.17	0.18	5481
ESFJ	0.03	0.04	0.04	360
ESFP	0.02	0.02	0.02	314
ESTJ	0.05	0.05	0.05	313
ESTP	0.09	0.05	0.07	674
INFJ	0.29	0.28	0.28	12604
INFP	0.31	0.43	0.36	15146
INTJ	0.23	0.22	0.22	8777
INTP	0.26	0.26	0.26	10518
ISFJ	0.15	0.10	0.12	1359
ISFP	0.11	0.08	0.09	2022
ISTJ	0.12	0.07	0.09	1670
ISTP	0.13	0.12	0.12	2601
accuracy			0.25	70948
macro avg	0.15	0.14	0.14	70948
weighted avg	0.24	0.25	0.24	70948

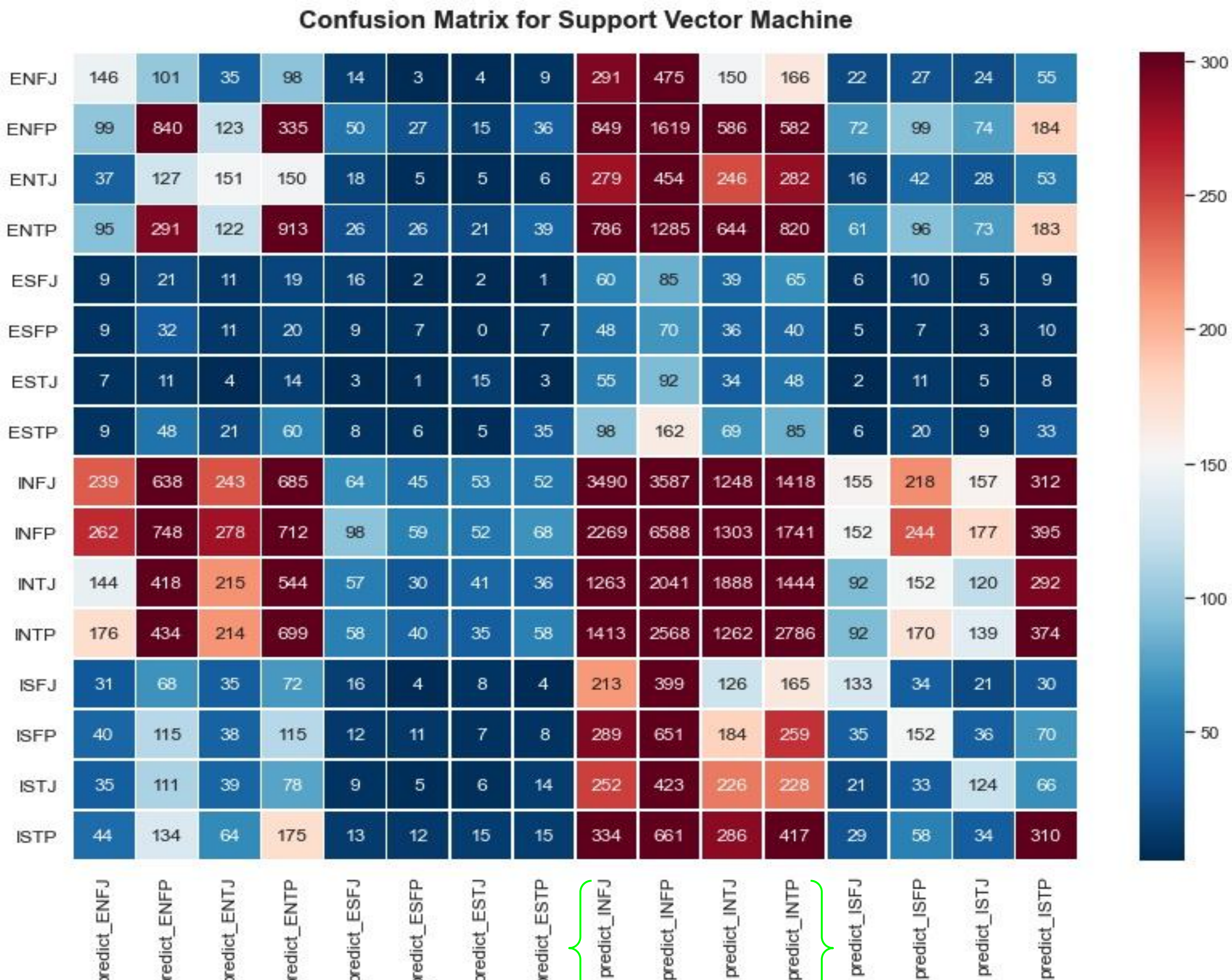
- Accuracy: 25%
- Confusion Matrix: Best performance shown in subset ‘INFP’
- Accuracy is higher for those metrics that are overrepresented

Stats for SVM

```
predicted_svm = text_clf_svm.predict(X_test)
print("Training set score: %f" % text_clf_svm.score(X_train, y_train))
print("Test set score: %f" % text_clf_svm.score(X_test, y_test))
print("Test error rate: %f" % (1 - text_clf_svm.score(X_test, y_test)))
print("Number of mislabeled points out of a total %d points for the Linear SVM algorithm: %d"
% (X_test.shape[0],(y_test != predicted_svm).sum()))
```

```
Training set score: 0.538653
Test set score: 0.249239
Test error rate: 0.750761
Number of mislabeled points out of a total 70948 points for the Linear SVM algorithm: 53265
```

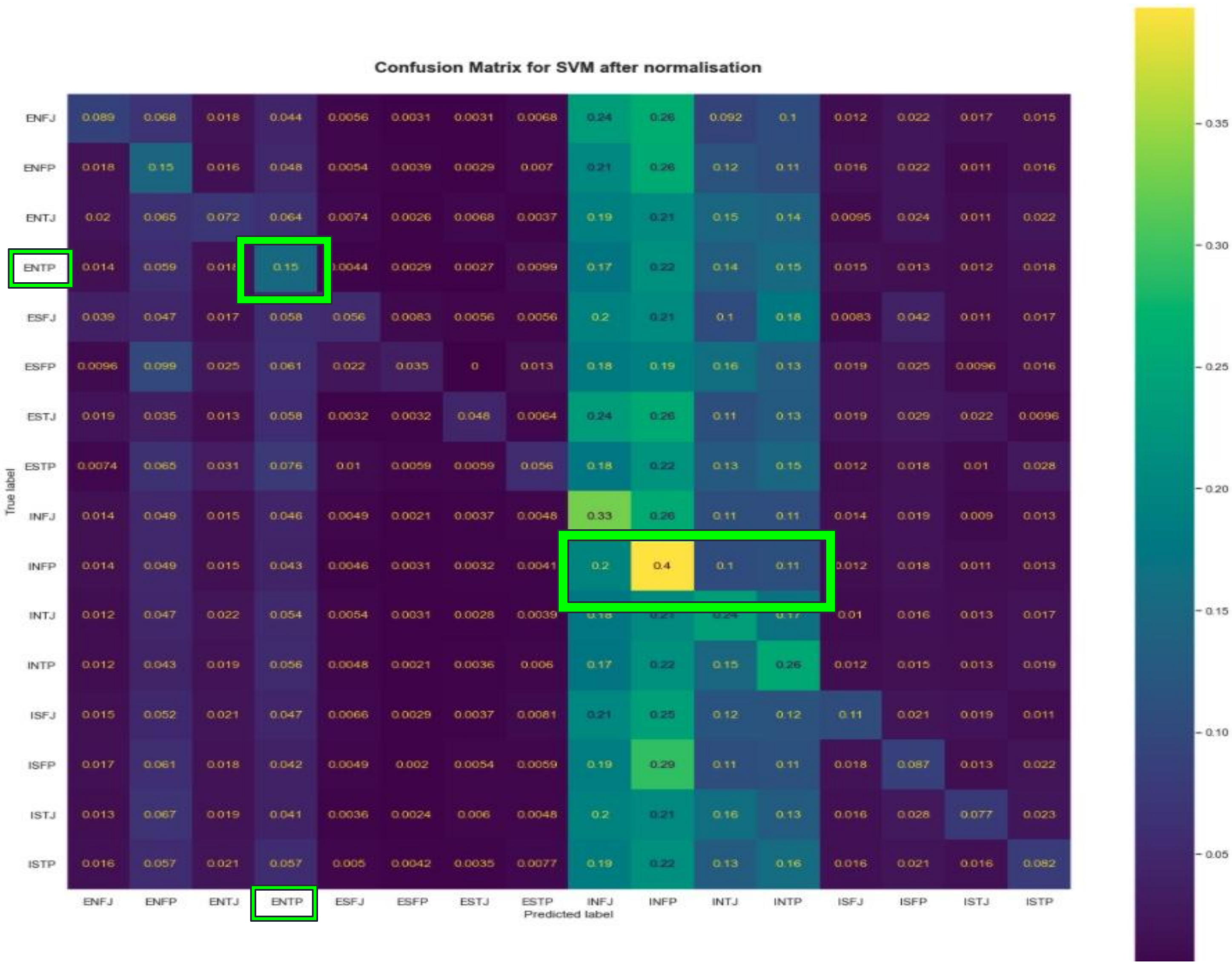
SVM Heatmap



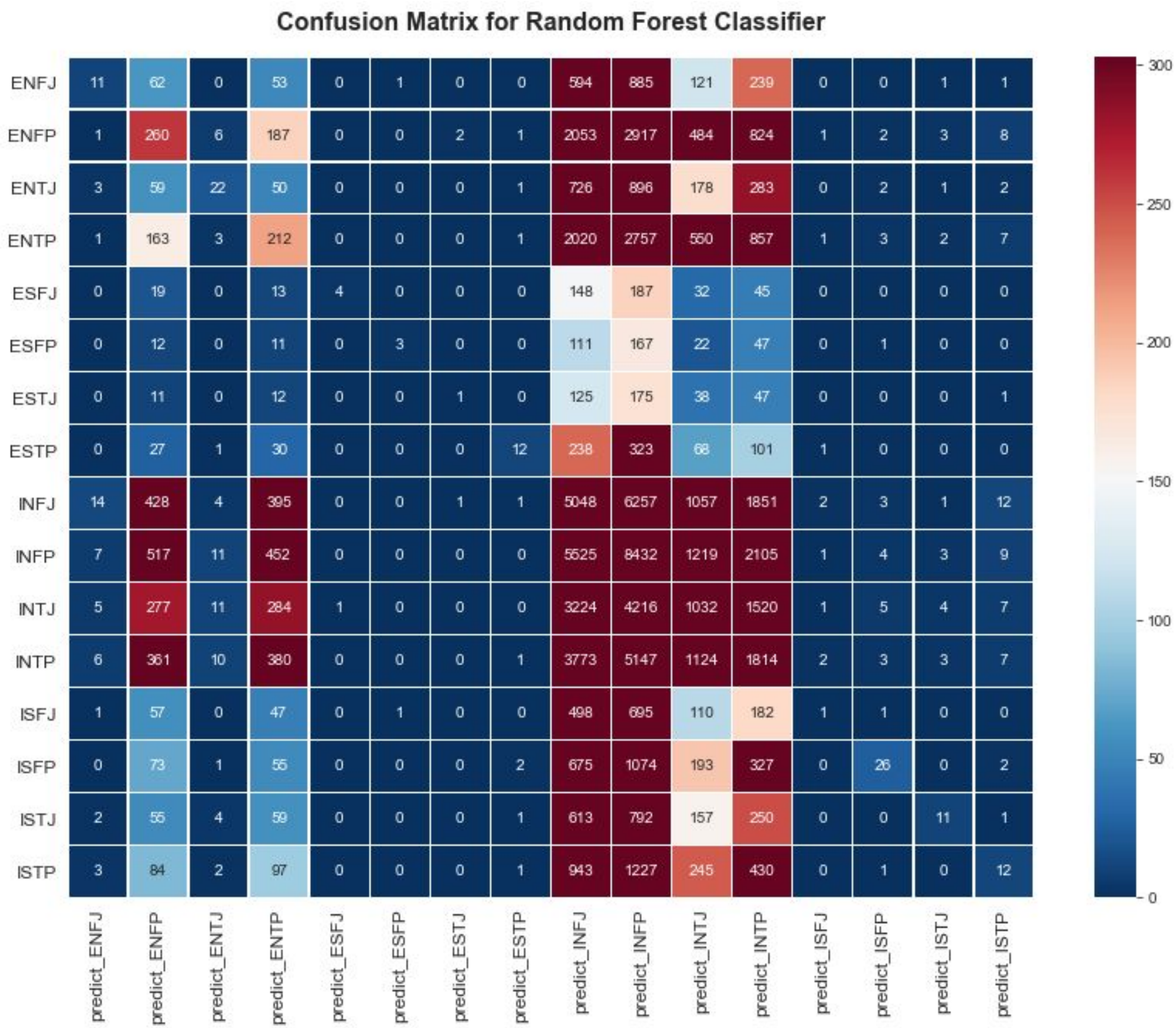
Overrepresented

It’s hard to understand this visualization so we normalized it.

SVM heatmap normalized



RANDOM FOREST RESULTS



- Accuracy: 19.9%
- Confusion Matrix: Best performance shown in subset ‘IN’ and ‘EN’ personalities
- Conclusion: Slightly lower accuracy than baseline so not suited to predicting classes outside of the ones with the highest frequency.

NEURAL NETWORK RESULTS

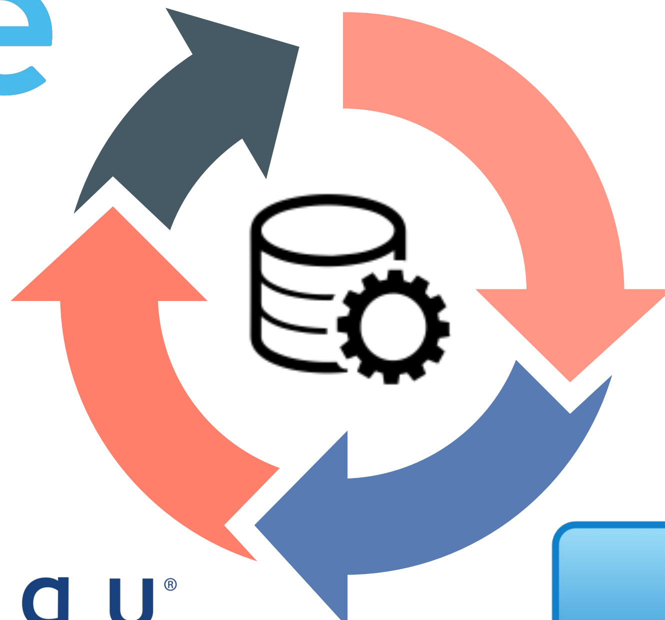
```
# Define the model - deep neural net, i.e., the number of input features and hidden nodes for each layer.
number_input_features = X_train_scaled.shape[1]
hidden_nodes_layer1 =  80
hidden_nodes_layer2 = 30


nn = tf.keras.models.Sequential()

# First hidden layer
nn.add(
    tf.keras.layers.Dense(units=hidden_nodes_layer1, input_dim=number_input_features, activation="relu")
)

# Second hidden layer
nn.add(tf.keras.layers.Dense(units=hidden_nodes_layer2, activation="relu"))



# Output layer
nn.add(tf.keras.layers.Dense(units=16, activation="sigmoid"))

# Compile the Sequential model together and customize metrics
nn.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])


# Define the checkpoint path and filenames
os.makedirs("Optimization_Checkpoints/checkpoints3/",exist_ok=True)
checkpoint_path = "Optimization_Checkpoints/checkpoints3/weights.{epoch:02d}.hdf5"

# Create a callback that saves the model's weights
cp_callback = ModelCheckpoint(
    filepath=checkpoint_path,
    verbose=1,
    save_weights_only=True,
    save_freq=4020)

# Train the model
fit_model = nn.fit(X_train_scaled, encoded_y_train, epochs=100, callbacks=[cp_callback])
```

- Accuracy: 27%
- Loss: 18.9%
- Conclusion: Best accuracy than baseline model at 21%.

DASHBOARD CONNECTION

kaggle

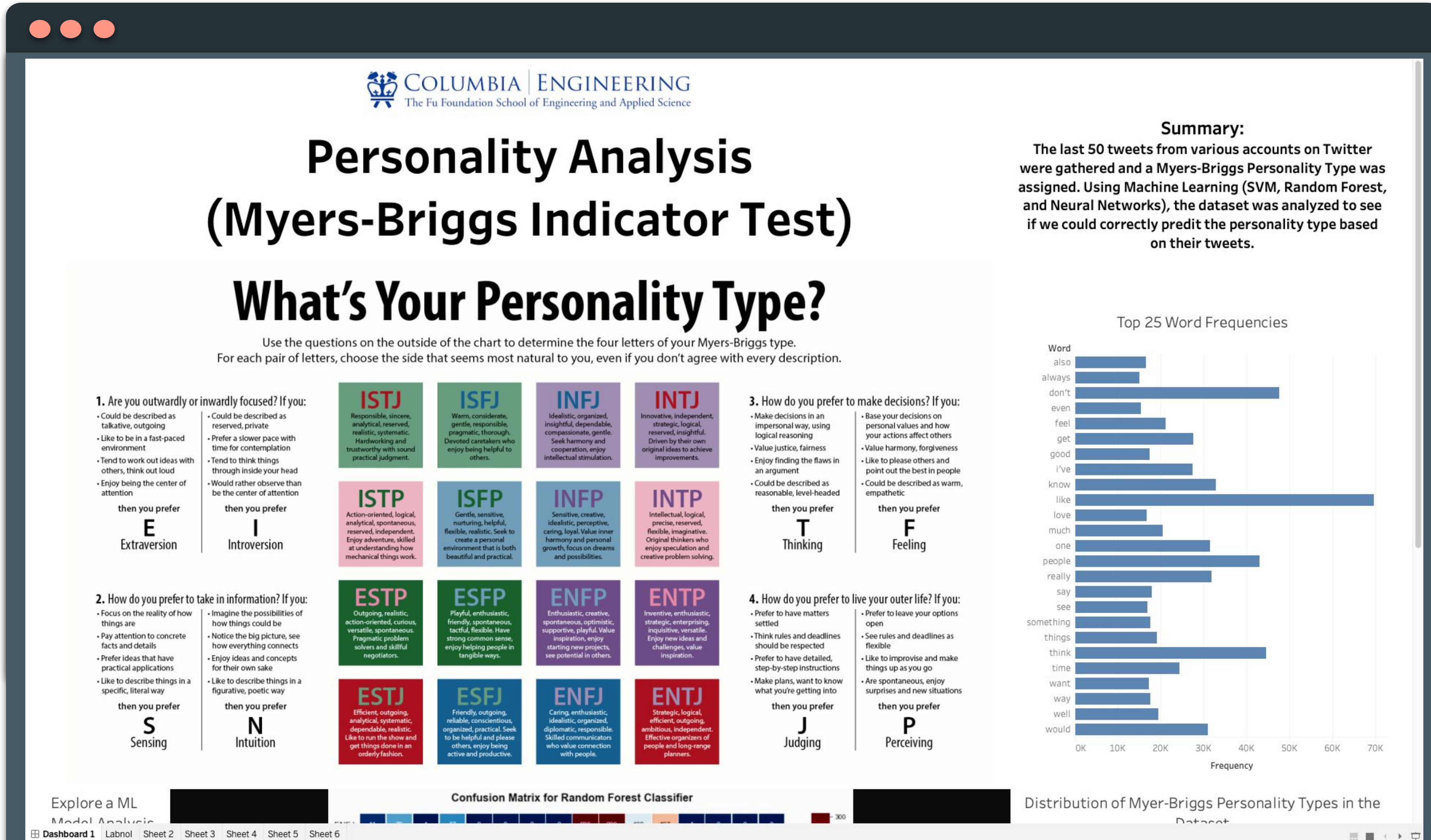
jupyter



tableau

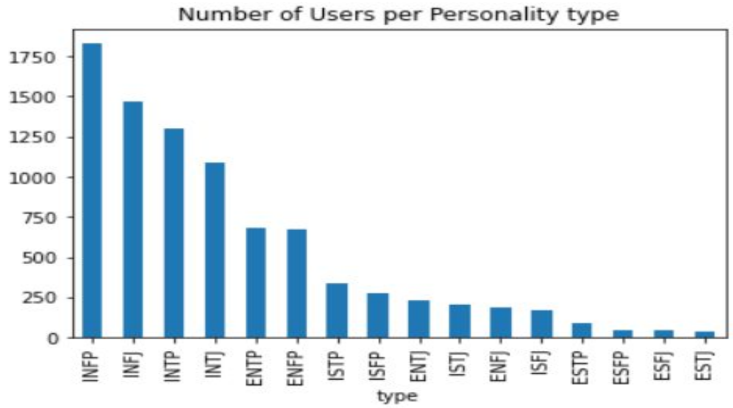
SQLite

DASHBOARD (CLICK HERE)



ANALYSIS

- Larger (n) increases accuracy, our dataset included tweets from 8,675 users with over 50% belonging to 3 personality types.
- Sentiment analysis would be explored in further analysis, by which certain words would be positively or negatively weighted based on their correlation to certain personality types.
- The models misclassify a lot of data by predicting the personality types that are overrepresented. One way to improve this model is by regularizing and penalizing the data points so that the dataset is not so skewed.



THANKS!

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, infographics & images by Freepik and illustrations by Stories