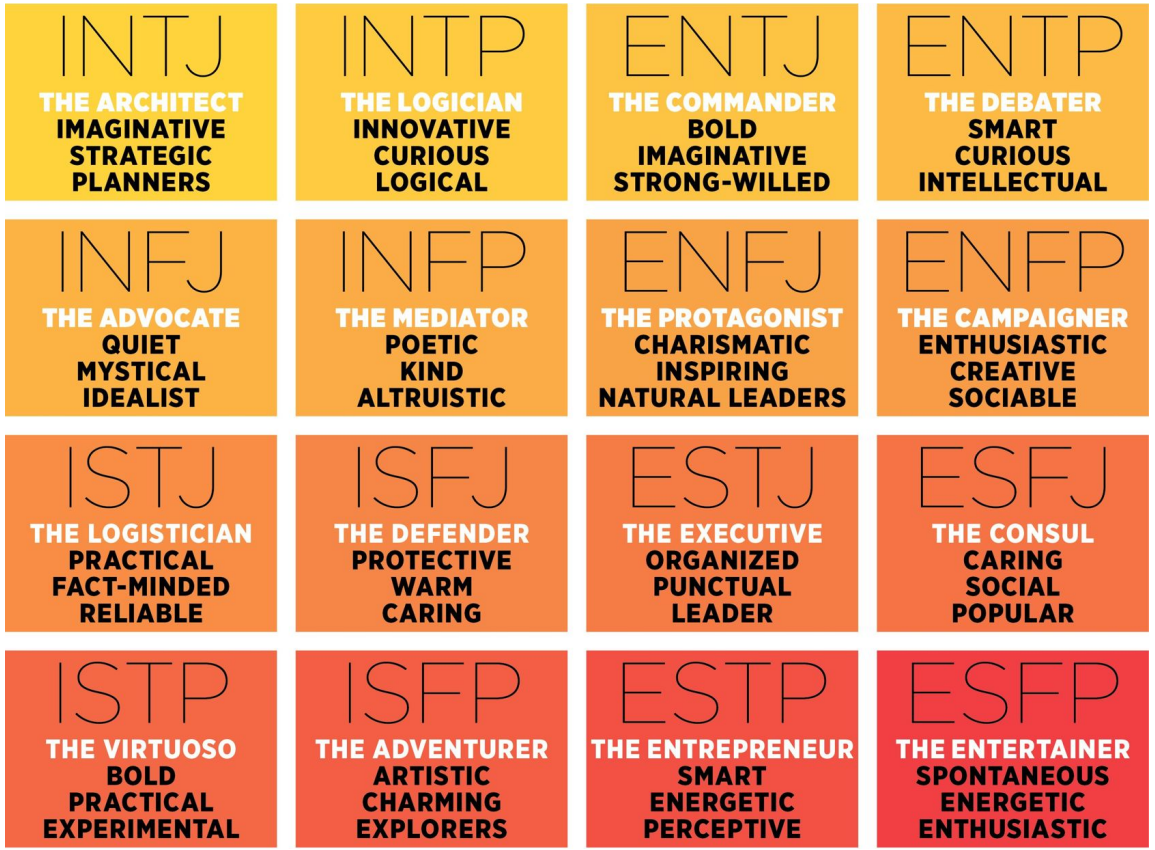
Myers-Briggs Personality Predictor

Devinaa Mangal, Troy Wilson and Brian Putman



Aim

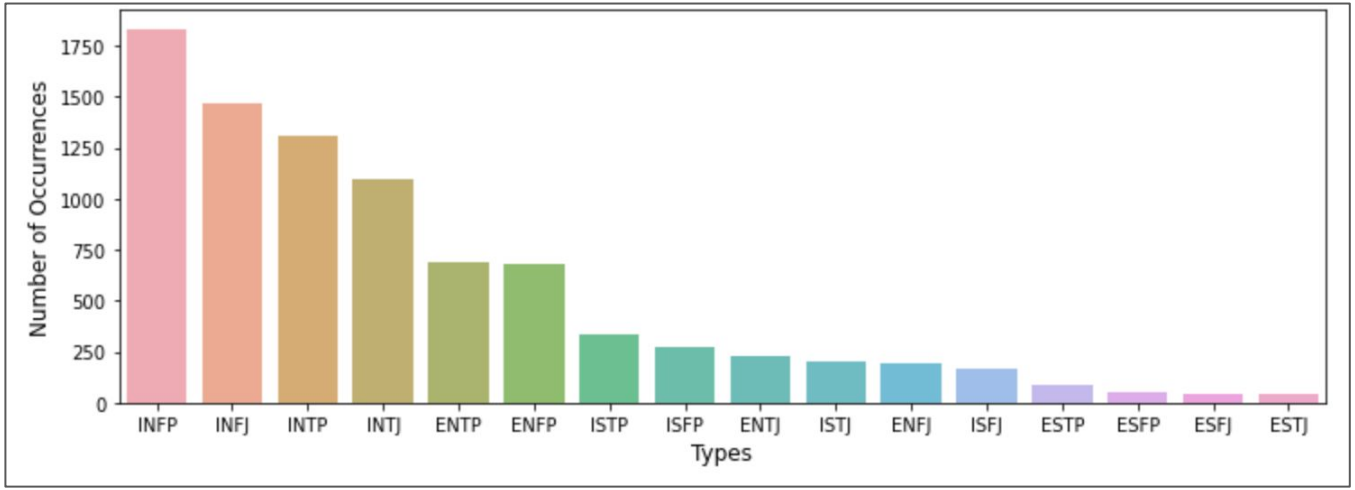
- Goal
  - The overall goal for the outcome is to be able to predict a person's personality type based on individual's social media presence, as well as evaluate the validity of the MBTI's test and its ability to predict the converse — using MBTIs to predict language styles and behavior.
- Questions
  - Can you accurately predict an individual's Meyer-Briggs personality type based on their social media activity?
  - Which model is most accurate?
  - Which features are most important to an accurate prediction?
  - What other information can we learn from data set?

Data

The dataset contains 8,675 observations (people), where each observation gives a person's:

- Myers-Briggs personality type (as a 4-letter code)
- An excerpt containing the last 50 posts on their PersonalityCafe forum (each entry separated by "|||")

	type	posts
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...
1	ENTP	'I'm finding the lack of me in these posts ver...
2	INTP	'Good one _____ https://www.youtube.com/wat...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...
4	ENTJ	'You're fired.|||That's another silly misconce...



Cleaning and Tokenization

1. Removed:
   a. Stopwords
   b. Punctuation
   c. Personality Types from the posts
2. Lower the words
3. WordNetLemmatizer()
4. Tokenized

	type	posts	tokens
0	INFJ	'http://www.youtube.com/watch?v=qsXHcwe3krw|||...	[moment, sportscenter, top, ten, play, prank, ...
1	ENTP	'I'm finding the lack of me in these posts ver...	[finding, lack, post, alarming, sex, boring, p...
2	INTP	'Good one _____ https://www.youtube.com/wat...	[good, one, course, say, know, blessing, curse...
3	INTJ	'Dear INTP, I enjoyed our conversation the o...	[dear, enjoyed, conversation, day, esoteric, g...
4	ENTJ	'You're fired.|||That's another silly misconce...	[fired, another, silly, misconception, approac...

	type	tokens	text
0	INFJ	[moment, sportscenter, top, ten, play, prank, ...	moment sportscenter top ten play prank life ch...
1	ENTP	[finding, lack, post, alarming, sex, boring, p...	finding lack post alarming sex boring position...
2	INTP	[good, one, course, say, know, blessing, curse...	good one course say know blessing curse absolu...
3	INTJ	[dear, enjoyed, conversation, day, esoteric, g...	dear enjoyed conversation day esoteric gabbing...
4	ENTJ	[fired, another, silly, misconception, approac...	fired another silly misconception approaching ...

Joined the tokens to make a continuous text

Word Clouds


Most common words
people
think
know
youtube
watch
really
make
even
thing
feel
mean
seem
want
love
well
though
INFP
INTP
INFJ
INTJ
time
type
something
good
look
life
Will
going
much
actually
thought
right
person
said
need
work
alway
find
sure
mind
feeling
might
agree
tell
guys
read
take
ENFP
Yeah
understand
never
believe
happy
point
question
experience
first
around
used
still
sound
reason
help
probably
sometime
great
better
everyone
least
especially
either
someone
anything
often
talk
idea
come
post
thread
year
little
https
youtube
kind
best
thinking
everything
friend
relationship
usually
give
girl
made
others
guess
maybe
definitely
problem
Thank
ENTP
part

INFJ

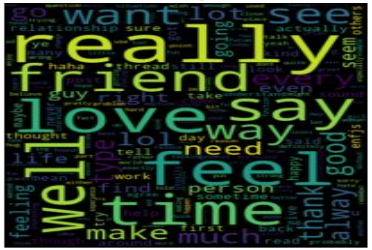
ENTP

INTP

INTJ

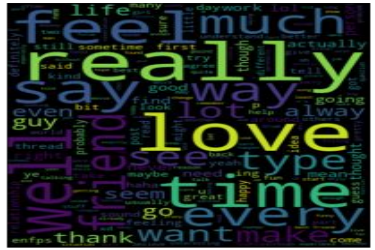
ENTJ

ENFJ



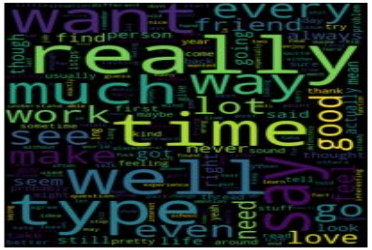
INFP

ENFP



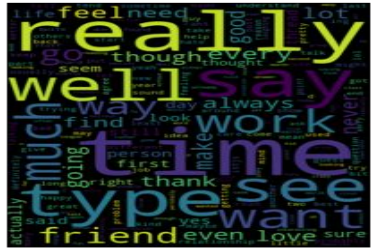
ISFP

ISTP



ISFJ

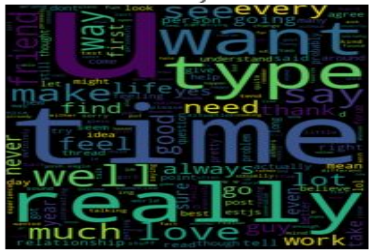
ISTJ



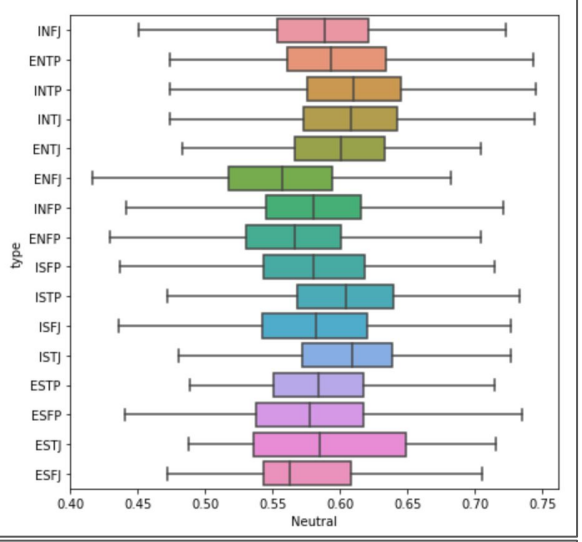
ESTP

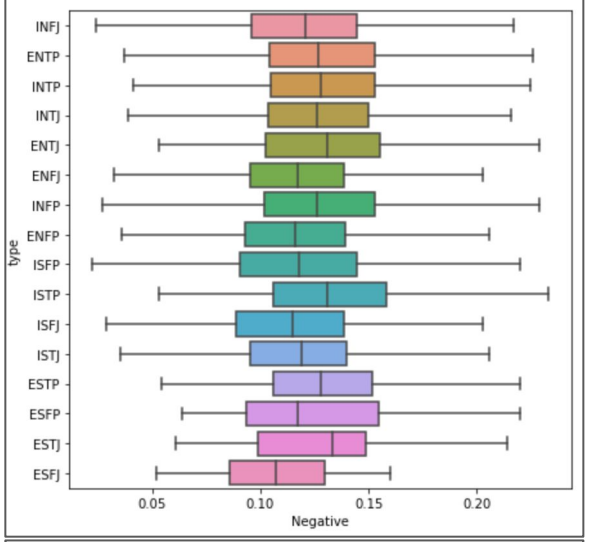
ESFP

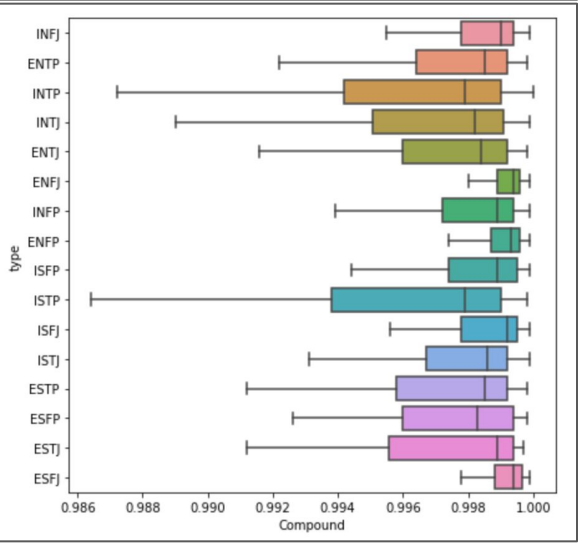
ESTJ

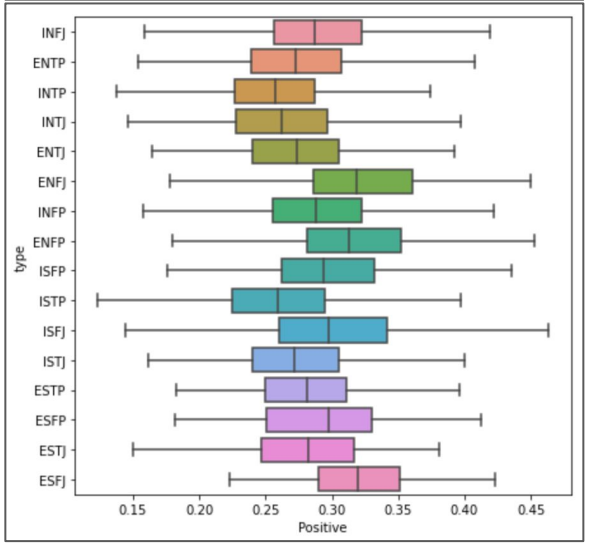


ESFJ









Sentiment Analysis

Vadar Analyzer to get positive, negative, neutral and compound sentiments

FEATURE EXTRACTION

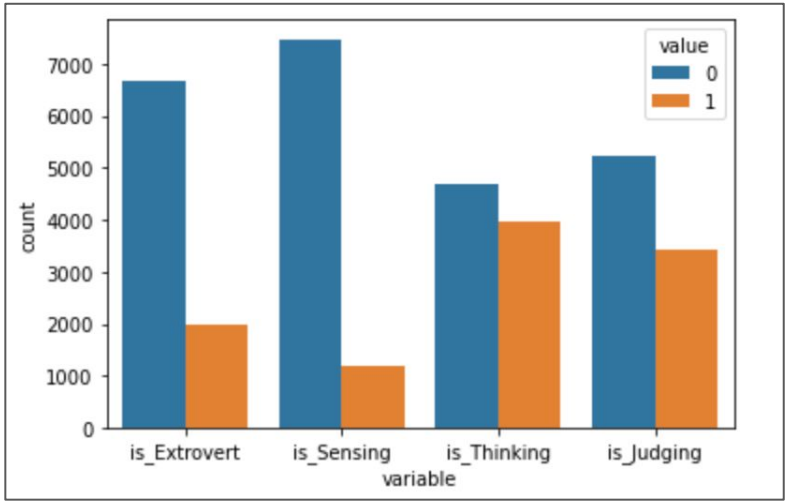
Words per comment	Variance of words per comment	Links per comments	Images per comment
Question marks per comments	Exclamation marks per comment	Ellipsis per comments	Nouns per comment
Verbs per count	Adjectives per comment	Prepositions per comments	Interjections per comment
Determiners per comment	Sentiment Analysis	TF IDF	Countvectorizer

Target Variables

	type	is_Extrovert	is_Sensing	is_Thinking	is_Judging	text
0	INFJ	0	0	0	1	moment sportscenter top ten play prank life ch...
1	ENTP	1	0	1	0	finding lack post alarming sex boring position...
2	INTP	0	0	1	0	good one course say know blessing curse absolu...
3	INTJ	0	0	1	1	dear enjoyed conversation day esoteric gabbing...
4	ENTJ	1	0	1	1	fired another silly misconception approaching ...

```
Introvert Count: 6676
Extrovert Count: 1999
Intuition Count: 7478
Sensing Count: 1197
Feeling Count: 4694
Thinking Count: 3981
Perceiving Count: 5241
Judging Count: 3434
```

High imbalance between **E-I** and **N-S**

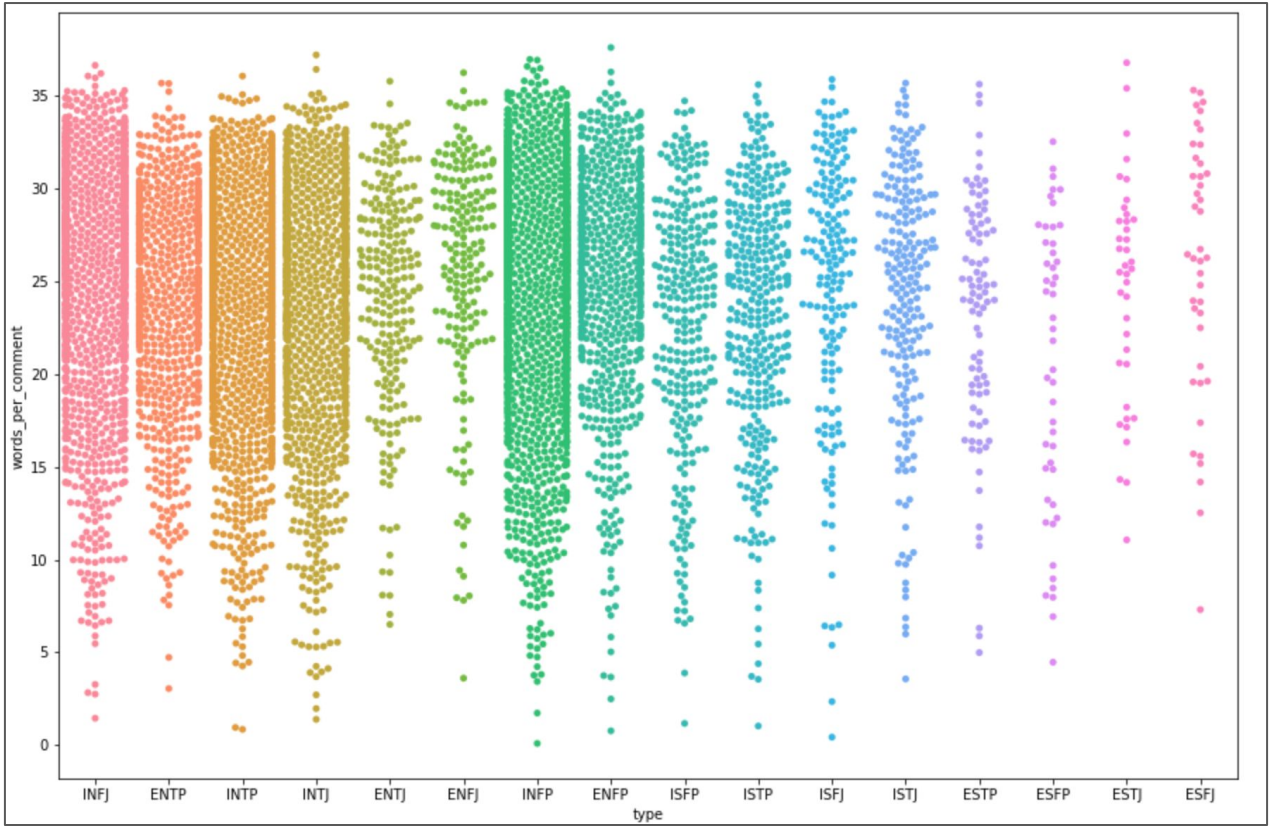


Parts of Speech

nouns_per_comment	adjs_per_comment	verbs_per_comment	prepositions_per_comment	interjections_per_comment	determiners_per_comment
5.18	1.62	1.72	0.54	0.00	1.06
3.98	2.12	4.68	2.72	0.04	1.84
2.70	1.68	3.10	1.32	0.08	1.10
3.28	1.80	4.14	1.88	0.04	1.86
3.58	1.80	4.42	1.78	0.06	1.60

words_per_comment	variance_of_word_counts	http_per_comment	img_per_comment	qm_per_comment	excl_per_comment	ellipsis_per_comment
11.12	135.2900	0.48	0.12	0.36	0.06	0.30
23.40	187.4756	0.20	0.02	0.10	0.00	0.38
16.72	180.6900	0.10	0.00	0.24	0.08	0.26
21.28	181.8324	0.04	0.00	0.22	0.06	0.52
19.34	196.4576	0.12	0.04	0.20	0.02	0.42

Data Imbalance



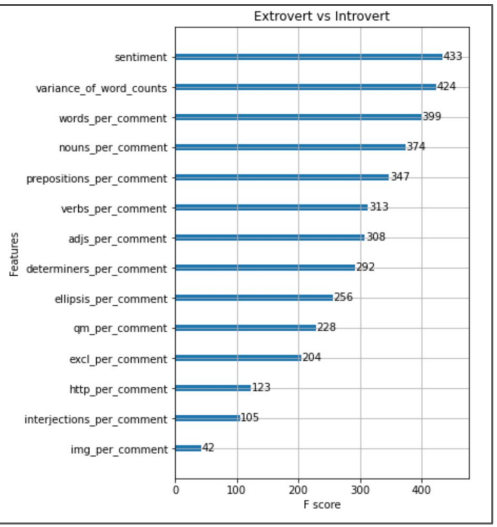
Feature Importance

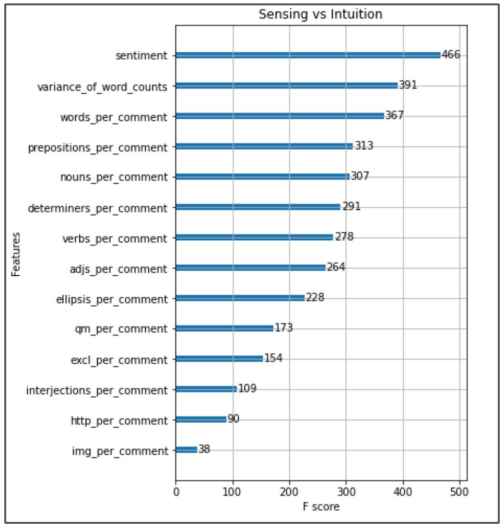
	model	accuracy	precision	recall	f1score	specificity
0	XGBoost	0.216282	0.173088	0.215994	0.181250	0.941301
0	RandomForest	0.215994	0.171780	0.214841	0.174131	0.941650
0	GaussianNaiveBayes	0.178242	0.169792	0.176225	0.152649	0.941225
0	kNN	0.154035	0.143637	0.154035	0.146515	0.937493
0	DecisionTree	0.139193	0.154690	0.147262	0.145332	0.937794
0	Logistic	0.210086	0.081250	0.211527	0.101408	0.937408

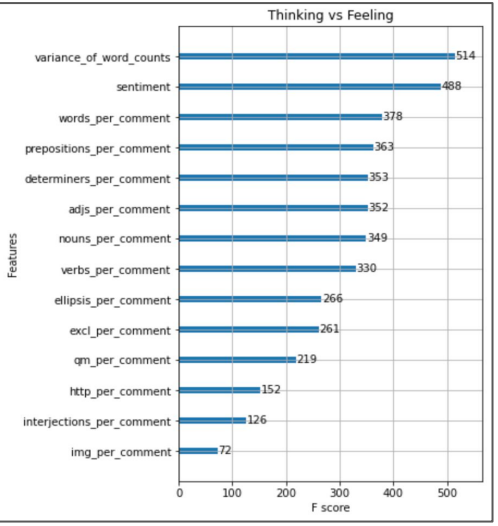
```
X = df_train[
    [
        "sentiment",
        "nouns_per_comment",
        "adjs_per_comment",
        "verbs_per_comment",
        "prepositions_per_comment",
        "interjections_per_comment",
        "determiners_per_comment",
        "words_per_comment",
        "variance_of_word_counts",
        "http_per_comment",
        "img_per_comment",
        "qm_per_comment",
        "excl_per_comment",
        "ellipsis_per_comment",
    ]
]
```

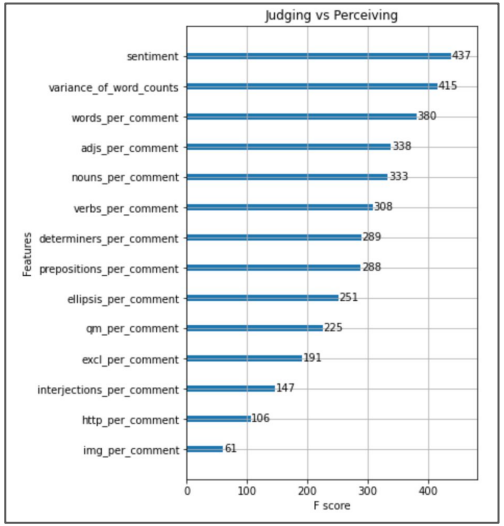
Y = is_Extrovert,
is_Sensing,
is_Thinking,
is_Judging

- Ran a couple models with the extracted features to be able to see which model does best.
- This was sorted based on f1 score which gives accuracy and precision equal importance.
- Here XG Boost did the best
- This model was used to extract the most important features









XG Boost

- Data was divided in training and testing data set based for each personality type.
- XGBClassifier() was used to extract the most important features

- XG Boost was used to see the importance of each feature for that particular characteristic
- All of the characteristics except Thinking vs Feeling shows that sentiments are very important
- All of them have the same top three features

MACHINE LEARNING MODELS

Set Up

```
# setting X to clean_posts, compound sentiment score, pos tags and various other counts
X = combined_df[
    [
        "text",
        "sentiment",
        "nouns_per_comment",
        "adjs_per_comment",
        "verbs_per_comment",
        "prepositions_per_comment",
        "interjections_per_comment",
        "determiners_per_comment",
        "words_per_comment",
        "variance_of_word_counts",
        "http_per_comment",
        "img_per_comment",
        "qm_per_comment",
        "excl_per_comment",
        "ellipsis_per_comment",
    ]
]

# setting y to four target classes -> is_Extrovert, is_Sensing, is_Thinking, is_Judging
y = combined_df.iloc[:, 24:28]
```

1. Divided the Vectorized data into training and testing sets
2. Fit the model
3. Defined predicted model
4. Set the model to give
    a. Geometric Mean Score
    b. ROC-AUC Score
    c. Average Precision-Recall Score
5. Ready to run multiple models

```
# for selecting k best features from features other than words
best_k_features = make_pipeline(MinMaxScaler(), SelectKBest(f_classif, k=10))

# setting up preprocessing for TF-IDF vectorizer
preprocesser_tf = ColumnTransformer(
    transformers=[
        (
            "tfidf",
            TfidfVectorizer(min_df=25, max_df=0.85, stop_words=additional_stopwords),
            "text",
        ),
        ("selectbest", best_k_features, counts_n_scores),
    ],
    remainder="passthrough",
)
```

```
# setting up preprocessing for COUNT vectorizer
preprocesser_ct = ColumnTransformer(
    transformers=[
        (
            "ct_vect",
            CountVectorizer(min_df=25, max_df=0.85, stop_words=additional_stopwords),
            "text",
        ),
        ("selectbest", best_k_features, counts_n_scores),
    ],
    remainder="passthrough",
)
```

- Preprocessing step was designed to vectorize the clean posts and to select k best features out of the other (non-word) features using column transformer.
- Since the data was imbalanced Imbalance Learn Pipeline was used to create the models. The pipeline was composed of preprocess (from the above step), random under sampler (to handle class imbalance) and the classification model.

Running Different Models

1. TF-IDF Logistic Regression
2. Count Vectorized Logistic Regression
3. TF-IDF Logistic Lasso
4. Count Vectorized Logistic Lasso
5. TF-IDF Logistic Ridge
6. Count Vectorized Logistic Ridge
7. TF-IDF Support Vector Classifier
8. Count Vectorized Support Vector Classifier
9. TF-IDF Naive Bayes
10. Count Vectorized Naive Bayes
11. TF-IDF Random Forest
12. Count Vectorized Random Forest

ROC-AUC Score

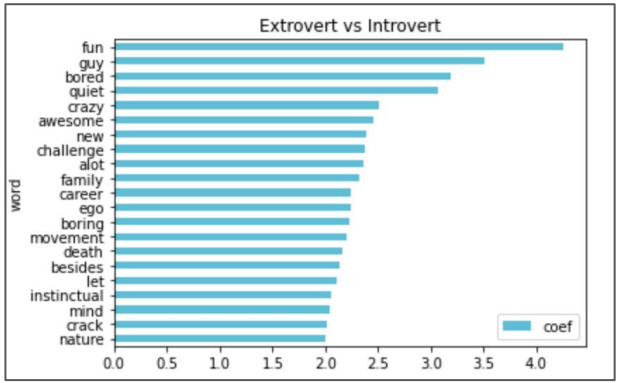
Model	E vs I	S vs N	F vs T	J vs P
TF-IDF Logistic Regression	0.76	0.73	0.87	0.68
Count Vectorized Logistic Regression	0.73	0.71	0.86	0.67
TF-IDF Logistic Lasso	0.71	0.7	0.86	0.66
Count Vectorized Logistic Lasso	0.71	0.67	0.85	0.66
TF-IDF Logistic Ridge	0.74	0.74	0.87	0.68
Count Vectorized Logistic Ridge	0.73	0.7	0.86	0.67
TF-IDF Support Vector Classifier	0.74	0.73	0.87	0.67
Count Vectorized Support Vector Classifier	0.67	0.68	0.81	0.62
TF-IDF Naive Bayes	0.75	0.73	0.85	0.67
Count Vectorized Naive Bayes	0.75	0.74	0.85	0.67
TF-IDF Random Forest	0.7	0.65	0.81	0.63
Count Vectorized Random Forest	0.7	0.68	0.81	0.6

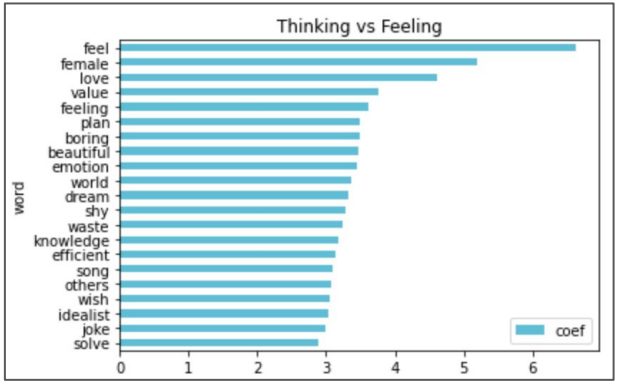
Average Precision-Recall Score

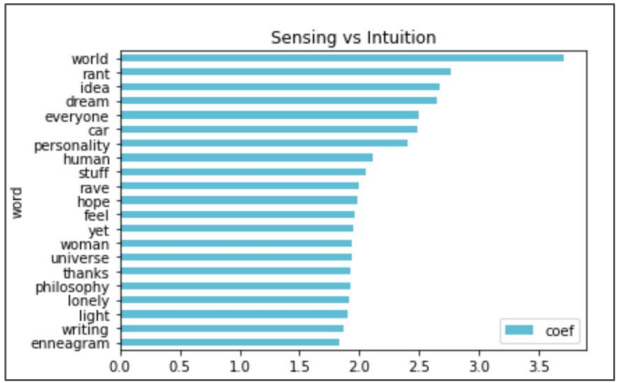
Model	E vs I	S vs N	F vs T	J vs P
TF-IDF Logistic Regression	0.5	0.33	0.84	0.58
Count Vectorized Logistic Regression	0.45	0.29	0.83	0.55
TF-IDF Logistic Lasso	0.45	0.3	0.83	0.55
Count Vectorized Logistic Lasso	0.44	0.24	0.82	0.56
TF-IDF Logistic Ridge	0.49	0.32	0.84	0.56
Count Vectorized Logistic Ridge	0.45	0.28	0.83	0.55
TF-IDF Support Vector Classifier	0.48	0.33	0.84	0.56
Count Vectorized Support Vector Classifier	0.4	0.26	0.77	0.5
TF-IDF Naive Bayes	0.48	0.32	0.8	0.56
Count Vectorized Naive Bayes	0.46	0.32	0.78	0.55
TF-IDF Random Forest	0.41	0.24	0.76	0.51
Count Vectorized Random Forest	0.41	0.26	0.75	0.47

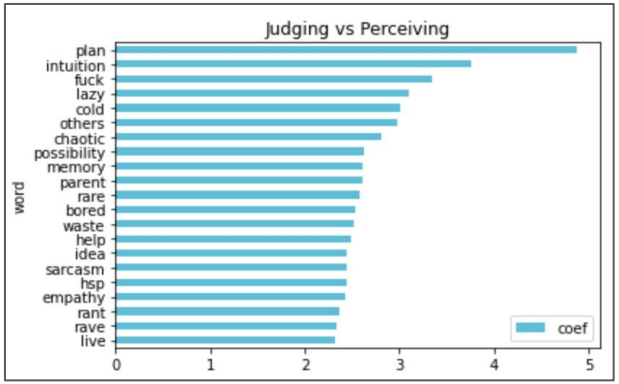
Final Model: Logistic Regression with TF-IDF Vectorization

Selected TF-IDF Logistic Regression as our final model as it returned the highest scores for all metrics - accuracy, precision, recall, roc-auc, avg_precision_recall as compared to other models.









Tweepy

- Python library for accessing the Twitter API
- Easily pull tweet ID, text and time data for users, topics, etc.
  - Option to include/exclude RTs, user replies
-

```
username = 'jack'
count = 50
tweets = tweepy.Cursor(api.user_timeline,id=username).items(count)
tweets_list = [tweet.text for tweet in tweets]
```

```
['RT @AmritaAhuja: 🌎💚 commited to doing our part',
 '@aismallard @GiveDirectly No',
 'RT @Square: Today we published our 2020 Corporate Social Responsibility Report. \n\nWe’re committed
to operating a responsible and sustainabl…',
 'Ending this March 21st\n\nWill immediately convert proceeds to #Bitcoin\n\nAnd send to @GiveDirectl
y Africa Response',
 '@ConorOkus 💯',
 'RT @ConorOkus: Timechain &gt; Blockchain',
 'This bag is ♾ 🧡 https://t.co/wT8NXL5tqt',
 'Matt is incredible 🧡 https://t.co/OImdEaKnXE',
 'https://t.co/xCnWG9EqgD',
```

Postmortem

- The dataset was **heavily imbalanced** with most people identifying as introverted (I) and Intuitive (N) rather than extroverted (E) and Sensitive (S).
- This caused all our models to have a **hard time classifying** Extroversion vs. Introversion and Sensitivity vs. Intuition.
- We were able to overcome this problem to some extent by using **Imbalance Learn's Random Under Sampling method**. This improved the scores but not significantly.
- We also added **extra words** that didn't seem to be very helpful in classifying a personality trait to the stop words list to make the model predict more efficiently.
- Our model is able to predict 2 to 4 traits out of a total of 4 traits and we consider this as success because even for **human readers it is difficult to accurately predict a person's Myers Briggs personality type**.
- **Information lost** during data clean-up
    - E.g., emojis, images, links
- How would outcome change for different forms of social media?
    - I.e., LinkedIn vs. Twitter vs. Facebook

For Future

- Add more data specifically for the types that have **low counts** in our current dataset to balance the classes.
- Implement a **neural network** based model in an attempt to see if that can outperform our current model in accurately predicting all 4 traits of the Myers Briggs personality type.

Questions?