CSCI946 – BIG DATA ANALYTICS

ASSIGNMENT 2

TEAM MEMBERS

AKSHIT SOOD (7409254)

VIGNESH MURUGESAN (7087597)

SAI PRIYANKA VOORADA (7040222)

SRINIVAS REDDY OBILI (6917215)

NAME	CONTRIBUTION PERCENTAGE
AKSHIT SOOD	100%
VIGNESH MURUGESAN	100%
SAI PRIYANKA VOORADA	100%
SRINIVAS REDDY OBILI	100%

Task 1

There are 6 phases in a typical Big Data Analytics Life cycle - Discovery, Data Preparation, Model Planning, Model Building, Communicate Results and Operationalize.
Following the life cycle helps us to increase the chances of our project success, and helps us avoid/ ignore common pitfalls at the right time.
Below we have detailed how the different phases have been used in our project implementation.

Phase 1 - Discovery

In the Discovery phase, we have to ensure that we have clear Domain and technical knowledge about how we will execute the project.

To frame the problem, it is better to have more context around the domain. For this, we will first understand the data provided-We have been provided a dataset of users from Twitter. The dataset consists of 20,000 rows, each of which includes a user name, a random tweet, an account profile and image, a location, and even the color of a link or sidebar among some other information. Data is provided at the user ID level, where each row represents one user. The objective of this assignment is to analyze this Dataset, and distinguish the profile between Human and Non-Human.

The dataset has been provided in an Excel Sheet, and we have processed and analyzed this data using a Jupyter notebook.

After understanding the dataset, we have analyzed the tools and techniques which we will be using for this project-

We made use of Python libraries like Pandas and Numpy for Data pre-processing, and have made use of Matplotlib and Seaborn for visualization of our analysis. Apart from this, for Model building, we have used the Scikit library from python.

Phase 2- Data preparation

In this stage, we need to understand the data in detail. We need to pre-process the columns into a suitable format, before we use them in any model. As there are many columns(Total 26), to reduce computational processing time and to avoid complexity of modeling, we need to only consider columns that are relevant to our use case. We need to process them into the correct format if required for ML algorithms. The decision variable is gender. This variable has the following values present- Male, female, Brand, Unknown and Null.

For our use case, as we need to distinguish between Humans and Non-Humans, we will assign Male and Female as the Human category and Brand as Non-Human.
Unknown and Null values need to be assigned into the Human or Non-Human bucket as well. For the assignment of Unknown and Null values, we have replaced them with the mode of the gender dataset, which is Female.

If we see below, we can see median and different percentile values for different variables.

```
df.describe()
```

	_unit_id	_trusted_judgments	gender:confidence	profile_yn:confidence	fav_number	retweet_count	tweet_count	tweet_id
count	2.005000e+04	20050.000000	20024.000000	20050.000000	20050.000000	20050.000000	2.005000e+04	2.005000e+04
mean	8.157294e+08	3.615711	0.882756	0.993221	4382.201646	0.079401	3.892469e+04	6.587350e+17
std	6.000801e+03	12.331890	0.191403	0.047168	12518.575919	2.649751	1.168371e+05	5.000124e+12
min	8.157192e+08	3.000000	0.000000	0.627200	0.000000	0.000000	1.000000e+00	6.587300e+17
25%	8.157243e+08	3.000000	0.677800	1.000000	11.000000	0.000000	2.398000e+03	6.587300e+17
50%	8.157294e+08	3.000000	1.000000	1.000000	456.000000	0.000000	1.144150e+04	6.587300e+17
75%	8.157345e+08	3.000000	1.000000	1.000000	3315.500000	0.000000	4.002750e+04	6.587400e+17
max	8.157580e+08	274.000000	1.000000	1.000000	341621.000000	330.000000	2.680199e+06	6.587400e+17

Fig 1: median and different percentile values

Interestingly, from the table above, we can see that retweet_count is 0 for the median value, and it's even 0 for 75th percentile of the data, which means that 75% of users have not done retweets. We have also made use of RFE(Recursive Feature Elimination) for feature selection. By using the model's "coef" or "feature importances" attributes, RFE ranks features. It then removes any dependencies and collinearities present in the model by recursively deleting a small number of features per loop.

Apart from this, we have made use of visualizations as well to explore the dataset. We have made use of scatter plots, bar graphs as well as a heat maps for our analysis. Some of the visualizations which we explored have been provided below-

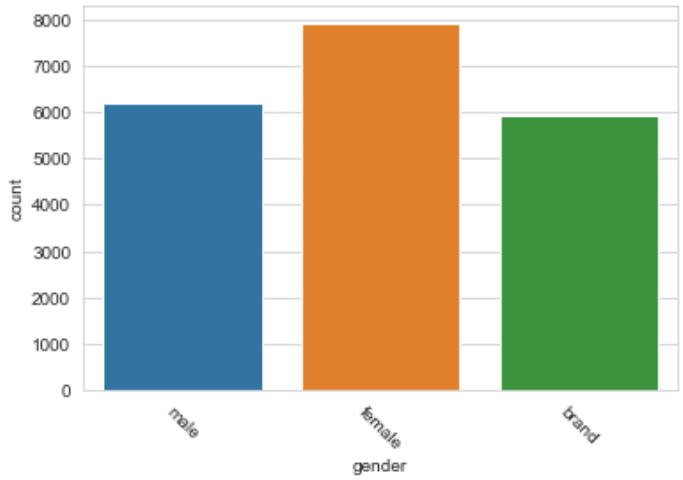

Fig 2 - Bar graph showing number of users belonging to different genders

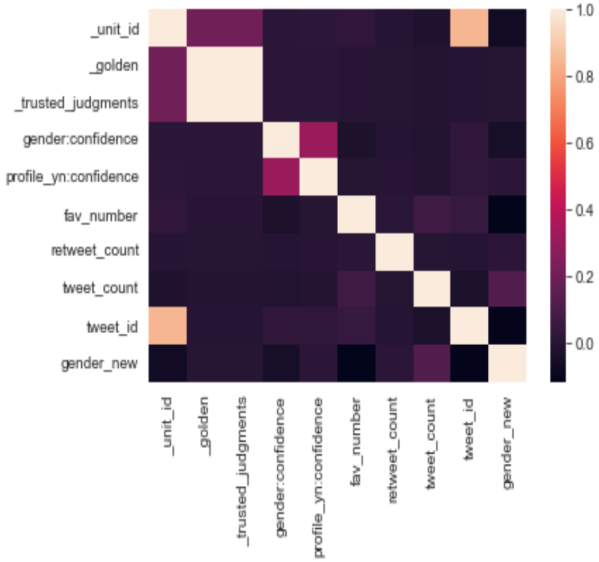

Fig 3- Heat Map showing correlation between different variables

Correlation is a good way to measure how one variable is related to another variable. However, We need to use our own judgment about the domain here as well.

Even though tweet_id (Random ID assigned to a tweet) and unit_id (Unique User Identifier) are showing high correlation in the Fig 2, it doesn't make much sense that they would be highly

correlated in practical sense. This might be a random case where both have numeric values which show correlation.

Phase 3- Model Planning

This phase is very important as well, as in this phase we need to select candidate models, and decide the model or set of multiple models which we would need to use for getting answers from the data. We first try to analyze the data, and do **feature selection** for the models. We have taken **two approaches** in this project for model selection.

For the First approach, we have selected the following features - _trusted_judgments , profile_yn:confidence, fav_number, tweet_count and retweet_count and used these variables as our independent variables for classification and regression. Our decision variable remains gender.

We have chosen Random Forest Classifier as classification algorithm and Decision Tree Regressor for Regression. As we need to distinguish whether an account is human or non-Human, this is a suitable problem for Classification. We have also made use of Clustering to uncover any useful patterns in the data.

For the second approach, we have made use of Text Processing and Association Ruleset to find interesting patterns in the data.

For text processing, we have chosen to implement features like removal of stop words, tokenization amongst others. We have chosen the variable 'text' which is a random tweet by user for our decision variable gender. After doing text processing, we designed a classifier and measured the accuracy of the model. We have made use of nltk, re modules and countVectorizer to implement text processing.

For Association rule mining, we have used Apriori from the mlxtend.frequent_patterns module to find support and confidence between different ruleset variables.

Phase 4- Model Building

After selecting the set of models, which we would like to build, we have implemented the algorithms using different Python libraries like Pandas, Numpy, Scikit Learn, nltk, mlxtend.frequent_patterns amongst others.

Below table gives the brief technical libraries used for the implementation of our models in Python.

Algorithm/ Model	Implementation
Classification	Decision Tree Classifier, Logistic Regression, RandomForest classifier
Regression	Decision Tree Regressor
Clustering	K- Means Clustering
Text Processing	Use of Stop words, Lemmatisation, Regular Expressions and Countvectorizers
Association Rule Mining	Apriori Algorithm

Table: Technical Libraries for Implementation

Phase 5 - Communicate Results

For the Decision Tree Classifier, we observed accuracy of **44.21%**.

For the Decision Tree Regressor, performance was not very ideal, as we observed a high MAE of **0.83**

When using Text Processing, we performed various operations on text columns like Regular Expression to handle special characters, avoid stop words, Lemmatization, Tokenization. After doing all these operations, and then using classification, we received high accuracy from Logistic Regression of about **76.75%**

We also used Association Rule Mining to find useful association rules. We did find support for certain ruleset, however we were not able to mine any rules, due to data size limitations.

Phase 6 - Operationalise

The python notebook provided can be used to extract Load and transform the data(Pre-Processing part). They also contain the code for running the models.

This document can be used for documentation purposes which details the assumptions and results of the various steps we have undertaken in this project.

Task 2

For Task 2, we have pre-processed data in different forms, and accordingly have applied various types of algorithms. Below we have detailed how we have used each of these algorithms in our problem statement.

Classification:

Classification is a supervised learning algorithm, which helps in predicting a class Label, based on a set of Input Data. For our use case, classification is essential, as we need to distinguish between Humans and Non Humans(Based on Gender column).

In the pre-processing part, all the null values are removed and the unknown gender class as nan value and all nan valued columns are removed. The gender column is encoded using the label encoder function. '_trusted_judgements', 'profile_yn:confidence', 'fav_number', 'retweet_count', 'tweet_count' columns are considered as independent variables for classification. Recursive feature elimination function is used to find the best 3 features among the 5 to give best result to the classification.

Random forest classifier is used to predict the gender class and obtained an accuracy of 44.4 %

Classification has again been used in the Text Processing section, where we have classified gender again, after performing Text Processing on the **'Text'** column.

Regression:

Regression is another commonly used Supervised Learning Algorithm, which is used to predict continuous outcomes, given a set of input.

In the pre-processing part, all the null values are removed and the unknown gender class as nan value and all nan valued columns are removed. The gender column is encoded using the label encoder function. '_trusted_judgements', 'profile_yn:confidence', 'fav_number', 'retweet_count', 'tweet_count' columns are considered as independent variables for classification. Recursive feature elimination function is used to find the best 3 features among the 5 to give best result to the classification.

Decision tree regression is used to predict the gender using the 3 best independent variables resulting from recursive feature elimination. The mean absolute error is quite high which is 0.831

Clustering:

Clustering is an algorithm where we group the dataset points into meaningful groups. Because this dataset is unlabelled, it is an unsupervised algorithm.

The preprocessing of the data is the same as mentioned above in classification and regression models. K-means clustering algorithm is used to create clusters and cluster centers with the number of clusters 3.

Below are the few graphs created using K-means clustering.

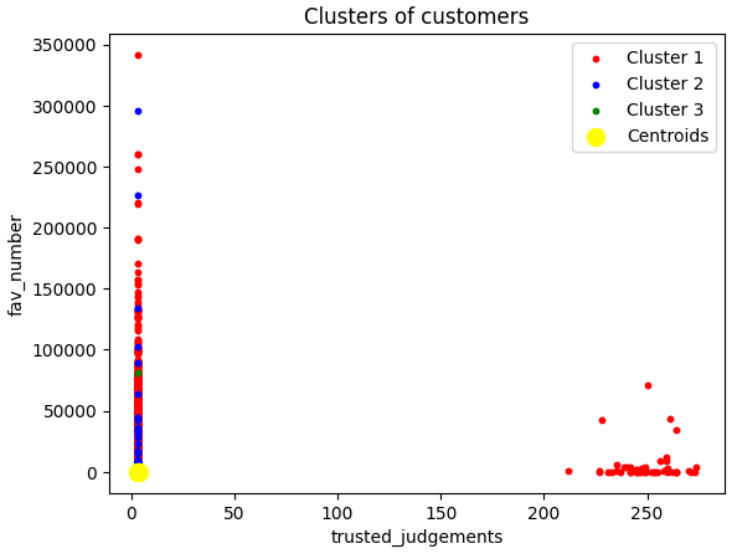

Fig 4: Cluster between trusted_judgements and fav_number

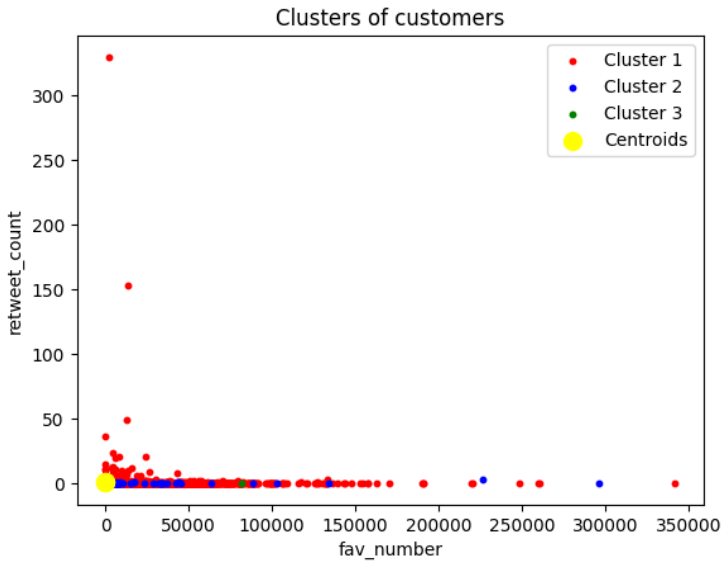

Fig 5: Cluster between fav_number and retweet_count

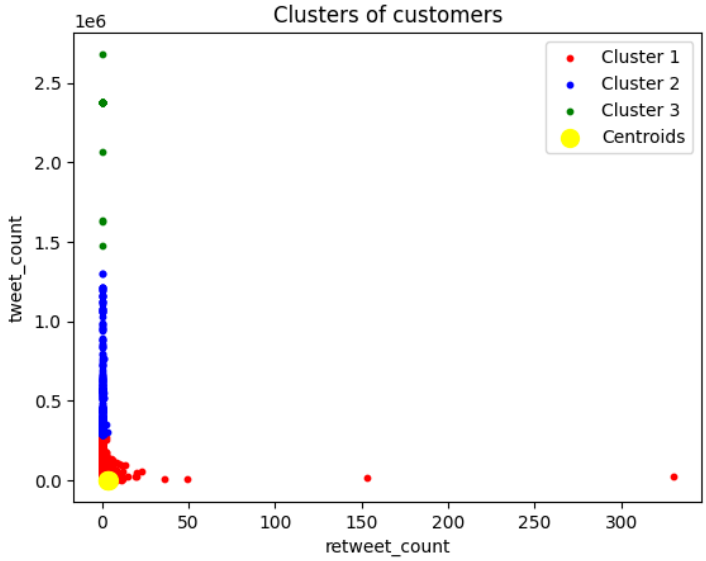

Fig 6: Cluster between retweet_count and tweet_count

As observed from the above plots, the clusters are not well separated and no decisive insights are drawn due to the large scale of data.

Text Processing:

For Text Processing, we have made use of the "Text" column in our dataset. The column gives useful information, of what a particular user ID has tweeted.

We have made use of text processing along with classification to build a classifier which will help to classify a particular user into a Human/ Non-Human profile.

For text- processing, we have imported nltk modules and RE modules for text processing.

Firstly, with the use of Regular Expressions, we have made sure all the characters other than alphabets(A-Z and a-z) are replaced by space. This has been done to make the data cleaner.

We also made use of stop - words removal, where words which are NOT important have been removed from our dataset. We have split the data using Tokenization.
Lemmatization has also been applied to the text format, where words are converted to their root values. For example ‘drive’, ‘drives’, ‘drove’, ‘driven’ will be converted to its root form ‘Drive’.

Below code snippet explains the operations performed-

```
text_cleaned = []           # empty list
for each in df['text']:
    each = re.sub("[^a-zA-Z]"," ", str(each))                                  # regex(regular expressions) for cleaning
    each = each.lower()                                                        # MAking sure everything is lowercase
    each = nlp.word_tokenize(each)                                             # Use of tokenisation for split
    each = [word for word in each if not word in set(stopwords.words("english"))]   # Removal of stop words
    lemma = nlp.WordNetLemmatizer()
    each = [lemma.lemmatize(i) for i in each]                                 # Words will be transformed to their roots
    each = " ".join(each)
    text_cleaned.append(each)
```

Fig 7: Text Processing code snippet

Once these operations were performed, we vectorised the data, used it as our independent variable in a classifier.

Text Vectorization is the process of converting text into numerical representation.¶We have also displayed the features, which were used in our code.

```
print("{} features used are {}".format(max_features,vectorizer.get_feature_names()))
```

```
600 features used are ['able', 'account', 'act', 'actually', 'add', 'age', 'ago', 'agree', 'air', 'album', 'amas', 'amazing',
'american', 'amp', 'answer', 'anymore', 'app', 'apple', 'apply', 'art', 'article', 'artist', 'artistoftheyear', 'ask', 'asked',
'available', 'average', 'award', 'away', 'awesome', 'babe', 'baby', 'bacon', 'bad', 'ball', 'bc', 'beat', 'beautiful', 'beaut
y', 'bed', 'believe', 'best', 'better', 'big', 'biggest', 'birthday', 'bit', 'bitch', 'black', 'blog', 'blood', 'blue', 'body',
'bond', 'book', 'bought', 'box', 'boy', 'break', 'bring', 'brother', 'budget', 'build', 'building', 'bus', 'business', 'buy',
'called', 'came', 'cancer', 'candy', 'car', 'card', 'care', 'case', 'cat', 'catch', 'cause', 'chance', 'change', 'channel', 'ch
aracter', 'check', 'checked', 'child', 'chill', 'christmas', 'city', 'class', 'click', 'close', 'club', 'coach', 'coffee', 'col
d', 'college', 'come', 'coming', 'comment', 'company', 'cool', 'costume', 'country', 'couple', 'course', 'cover', 'crazy', 'cre
dit', 'cup', 'cut', 'cute', 'dad', 'daily', 'damn', 'dance', 'dark', 'data', 'date', 'day', 'dead', 'deal', 'death', 'deserve',
'desk', 'die', 'difference', 'different', 'digital', 'doctor', 'dog', 'dont', 'door', 'dream', 'dress', 'drink', 'drive', 'dro
p', 'dude', 'early', 'earth', 'easy', 'eat', 'eating', 'em', 'email', 'end', 'enjoy', 'episode', 'event', 'everyday', 'everyday
iloveyou', 'excited', 'experience', 'eye', 'fa', 'face', 'facebook', 'fact', 'fall', 'family', 'fan', 'far', 'favorite', 'fee
l', 'feeling', 'fight', 'film', 'final', 'finally', 'fine', 'follow', 'followed', 'follower', 'following', 'food', 'football',
'force', 'forever', 'forevermore', 'forget', 'forward', 'free', 'friday', 'friend', 'fruit', 'fuck', 'fucking', 'fun', 'funny',
'future', 'game', 'gave', 'getting', 'gift', 'girl', 'giveaway', 'goal', 'god', 'going', 'gon', 'gone', 'good', 'got', 'great',
'greatest', 'green', 'group', 'growing', 'gt', 'guess', 'guy', 'haha', 'hair', 'half', 'halloween', 'hand', 'happen', 'happene
d', 'happiness', 'happy', 'hard', 'harry', 'hate', 'head', 'health', 'hear', 'heard', 'heart', 'hell', 'hello', 'help', 'hey',
'hi', 'high', 'history', 'hit', 'hold', 'holiday', 'home', 'hope', 'hot', 'hour', 'house', 'http', 'human', 'hurt', 'idea', 'i
m', 'important', 'incredible', 'inside', 'internet', 'iphone', 'issue', 'james', 'job', 'john', 'join', 'justinbieber', 'kid',
'kill', 'kind', 'know', 'la', 'lady', 'largest', 'late', 'later', 'latest', 'le', 'lead', 'learn', 'leave', 'left', 'let', 'li
e', 'life', 'light', 'like', 'liked', 'line', 'link', 'list', 'listen', 'listening', 'literally', 'little', 'live', 'living',
'lmao', 'lol', 'long', 'look', 'looked', 'looking', 'lord', 'lose', 'lost', 'lot', 'love', 'loved', 'low', 'lt', 'mad', 'make',
'making', 'man', 'matter', 'maybe', 'mean', 'meat', 'medium', 'meet', 'men', 'met', 'million', 'min', 'mind', 'minute', 'miss',
'mom', 'moment', 'monday', 'money', 'month', 'morning', 'movie', 'mr', 'music', 'na', 'need', 'netflix', 'new', 'news', 'nice',
'nigga', 'night', 'november', 'number', 'october', 'office', 'oh', 'ok', 'okay', 'old', 'omg', 'onedirection', 'online', 'ope
n', 'order', 'outside', 'page', 'paid', 'pain', 'paper', 'parent', 'park', 'party', 'past', 'pay', 'people', 'perfect', 'perso
n', 'phone', 'photo', 'pic', 'pick', 'picture', 'piece', 'pink', 'place', 'plan', 'play', 'played', 'player', 'playing', 'pm',
'point', 'pop', 'post', 'power', 'ppl', 'premiere', 'pretty', 'price', 'probably', 'problem', 'produce', 'product', 'project',
'proud', 'public', 'pumpkin', 'pushawardslizquens', 'question', 'quote', 'read', 'reading', 'ready', 'real', 'really', 'reaso
n', 'red', 'relationship', 'remember', 'report', 'rest', 'result', 'return', 'review', 'ride', 'right', 'road', 'rock', 'room',
'round', 'run', 'running', 'sad', 'said', 'sale', 'saturday', 'save', 'saw', 'say', 'saying', 'school', 'season', 'second', 'se
cret', 'seeing', 'seen', 'self', 'send', 'series', 'service', 'set', 'sex', 'sexual', 'share', 'shirt', 'shit', 'shoe', 'shor
t', 'shot', 'sick', 'sign', 'single', 'sister', 'sit', 'site', 'sleep', 'small', 'smile', 'snail', 'social', 'song', 'soon', 's
orry', 'sound', 'space', 'special', 'spectre', 'st', 'stand', 'star', 'start', 'started', 'state', 'stats', 'stay', 'step', 'st
op', 'storage', 'store', 'story', 'street', 'strong', 'struggle', 'student', 'study', 'stuff', 'stupid', 'style', 'suck', 'sund
ay', 'super', 'support', 'sure', 'sweet', 'ta', 'tag', 'taken', 'taking', 'talk', 'talking', 'tax', 'teacher', 'team', 'tech',
'tell', 'test', 'th', 'thank', 'thanks', 'thats', 'thing', 'think', 'thinking', 'tho', 'thought', 'ticket', 'till', 'time', 'ti
p', 'tired', 'today', 'told', 'tomorrow', 'tonight', 'took', 'tour', 'track', 'transforming', 'transponder', 'trecru', 'tried',
'trip', 'true', 'trump', 'trust', 'truth', 'try', 'trying', 'turn', 'tv', 'tweet', 'twitter', 'uk', 'unfollowed', 'unfollower
s', 'unit', 'update', 'ur', 'use', 'used', 'using', 'video', 'view', 'visit', 'voice', 'vote', 'voted', 'wait', 'waiting', 'wal
k', 'walking', 'wall', 'wan', 'want', 'wanted', 'war', 'watch', 'watched', 'watching', 'water', 'way', 'wear', 'wearing', 'weat
her', 'wednesday', 'week', 'weekend', 'welcome', 'went', 'white', 'win', 'winner', 'winter', 'wish', 'woman', 'wonder', 'word',
'work', 'workbench', 'working', 'world', 'worst', 'worth', 'wow', 'written', 'wrong', 'ya', 'yeah', 'year', 'yes', 'yesterday',
'yo', 'young', 'youtube']
```

Fig 8- Features computed after Vectorization

After the processing of text, we have used **Logistic Classification** to classify the genders into accurate buckets.

Association Rules

Association Rules help us to identify how different variables may have certain relationships.

{Eggs} -> {Milk} here means that typically if a person buys Eggs, they are likely to buy milk as well. In this equation, Eggs are antecedent and Milk is Consequent.

Support helps us identify the frequency that a rule occurs. Low support may also be useful, in some instances, to find hidden patterns.

Confidence (implies reliability of the rule) and Lift are some other interesting measures, which are seen in Association Rule Mining.

We have used the mlxtend library from which Apriori algorithm is readily available.
For Pre-processing, we have used the column, 'description', which is a textual field representing the user profile. We have processed the data in required form
We also had to pivot the data, which has been done with the use of Pandas, so that individual description becomes the column, and user_Id is the rows, with cells having tweet_count as value, after which we have one Hot Encoded the dataset.

Once this is done, we created frequent itemsets by applying Apriori function, by setting min_support as 0.001, so as to compute as much data as possible.

	support	itemsets
0	0.001105	('Cos even an old girl's best friend is still ...
1	0.001964	(Subscribe to her Inspirational channel here h...
2	0.001473	(The Map Game is a free geography quiz based o...
3	0.002026	(You can be spiritually empowered, financially...
4	0.001657	(secret little rendezvous)

Fig 9- Itemsets, alongside their support value

After computing itemsets, with their support values, we have tried to compute association rules, with metrics like Confidence, Lift, Leverage etc.

However, we were not able to observe any important rulesets, due to limitations in the meaningful ruleset in the column..

Task 3: Visualization

The method used to depict the relationships within the data is known as data visualization. To give the data a proper relationship, we used the matplot and seaborn libraries. The scatterplot, Implot, box, kde, count, pair, pairgrid, violin, dist, and joint graphical plots are used to display the visualization.

Scatterplot:

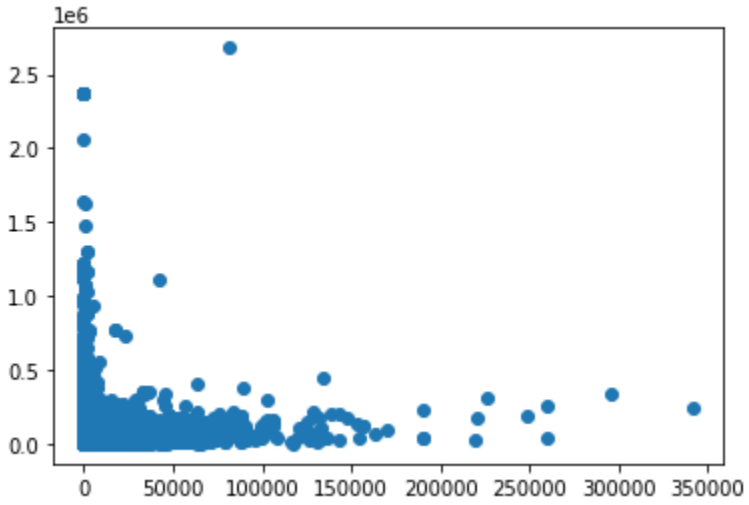

Fig 10: Scatter plot for fav_number and tweet_count

The scatterplot is used to display how the two variables are related to one another (x,y). A correlation in the scatter plot depicts the relationships in the scatterplots. The x-coordinate consists of fav_numbert and the y-coordinate consists of tweet_count. When the y variable rises relative to the x variable, the correlation is positive; otherwise, it is negative.

Implot:

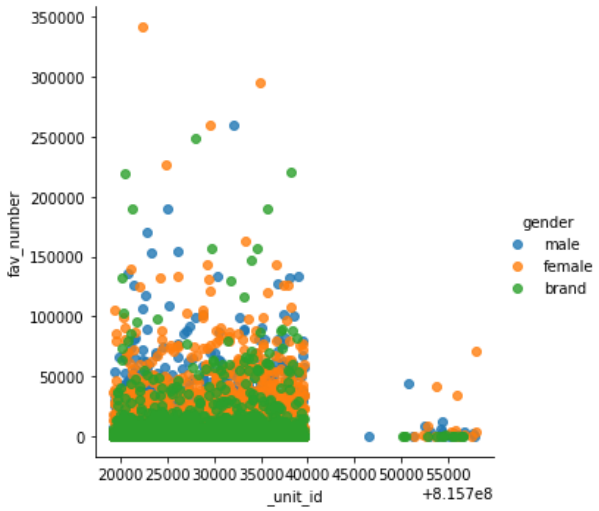

Fig 11: Implot for _unit_id and fav_number

Above graph shows the number of times a tweet has been favorited, by different user Ids. This has been further divided by gender. If we see, mostly brands have lower number of retweets, while male and female have higher value.

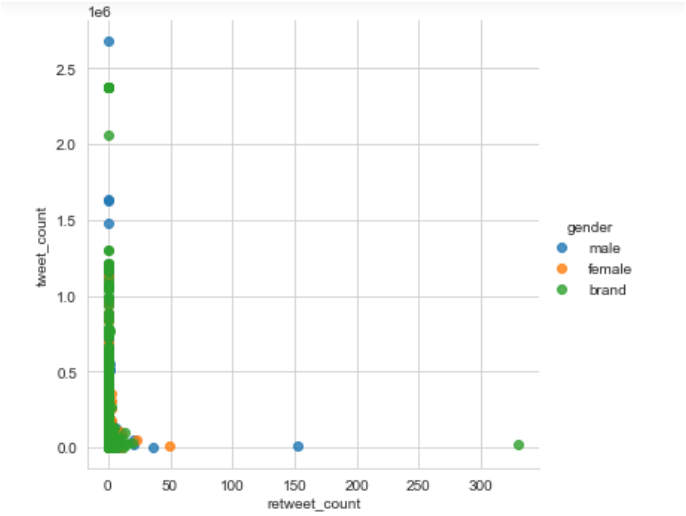

Fig 12: Implot for retweet_count and tweet_count

The seaborn library uses the implot, which is used to plot a scatter plot on a FacetGrid.

Boxplot:

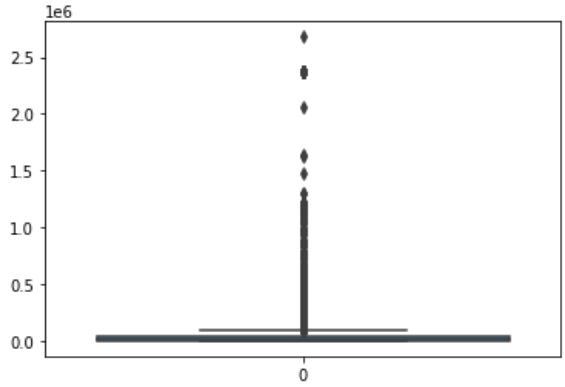

Fig 13: Box-plot for tweet_count

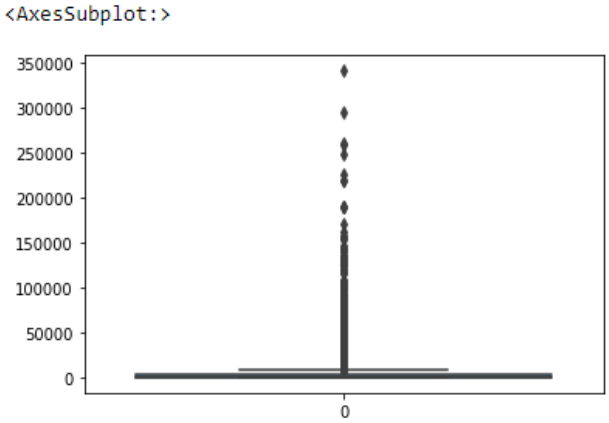

Fig 14: Box-plot for fav_number

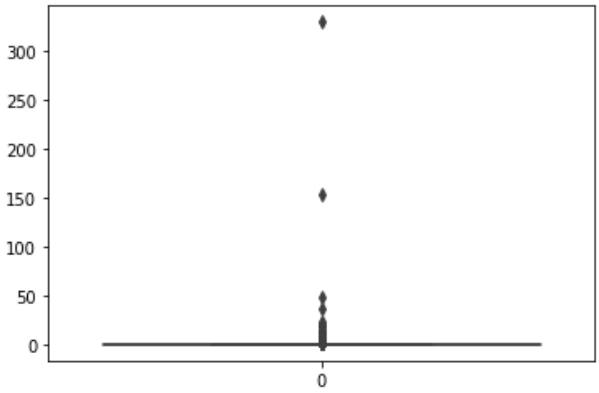

Fig 15: Box-plot for retweet_count

A box plot displays the distribution of numerical data in a way that makes comparisons across variables or between categorical variable levels easier.

Kdeplot:

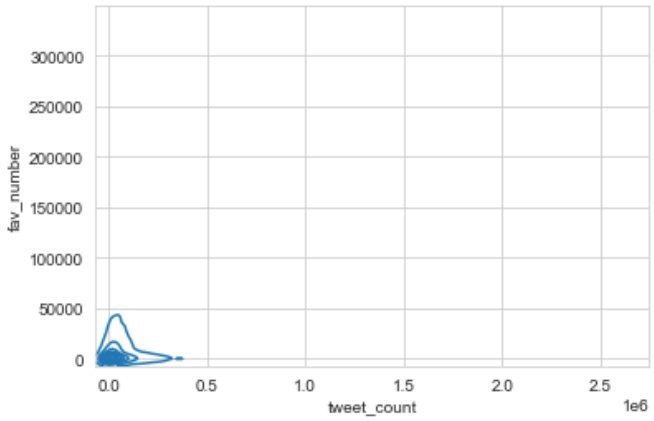

Fig 16: kdeplot for tweet_count and fav_number

Similar to a histogram, a kernel density estimate (KDE) plot is a technique for displaying the distribution of observations in a dataset. A continuous probability density curve in one or more dimensions is used by KDE to describe the data.

Countplot:

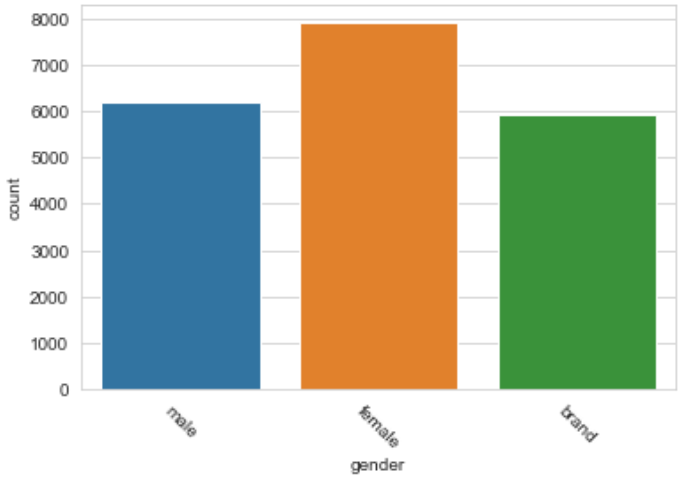

Fig 17: Count plot for each gender

```
Out[61]: (array([0, 1]), [Text(0, 0, '0'), Text(1, 0, '1')])
```

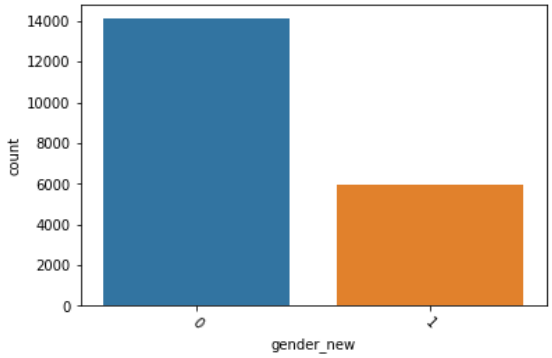

Fig 18: Count plot for gender_new and count

As we can see, 0 here represents Human and 1 represents Machine. In our Dataset, humans are more than double the number of Non-Humans.

A count plot resembles a histogram over a categorical variable as opposed to a quantitative one. You can compare counts across nested variables because the fundamental API and settings are the same as those for barplot().

Pairplot:

```
Out[92]: <seaborn.axisgrid.PairGrid at 0x1f7970af7f0>
```

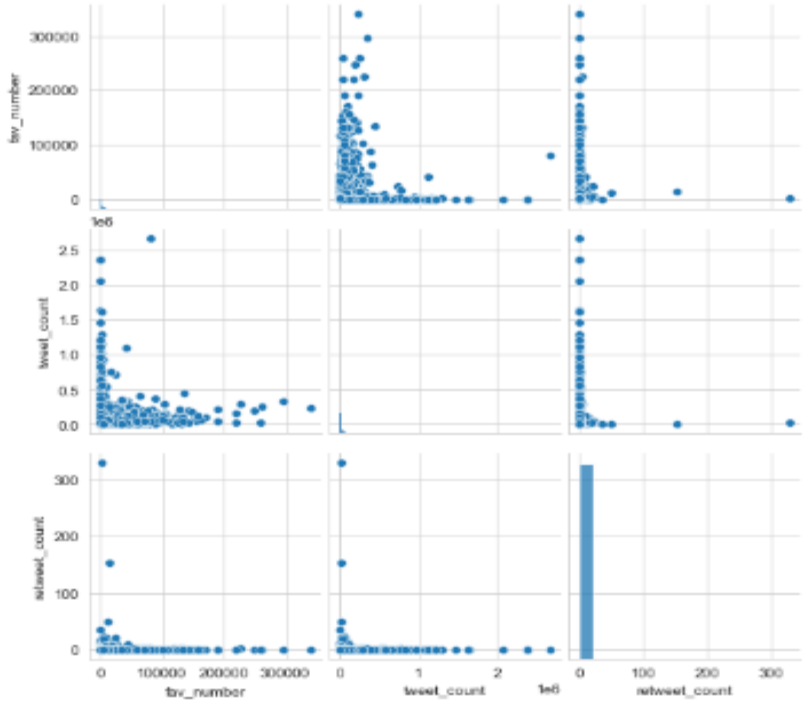

Fig 19 : Pair plot for fav_number, tweet_count, retweet_count

The relationships between the dataset were plotted using a paiplot. It is employed to produce a visualization and show it in a big data environment.

Pairgrid:

```
Out[93]: <seaborn.axisgrid.PairGrid at 0x1f79a3cc460>
```

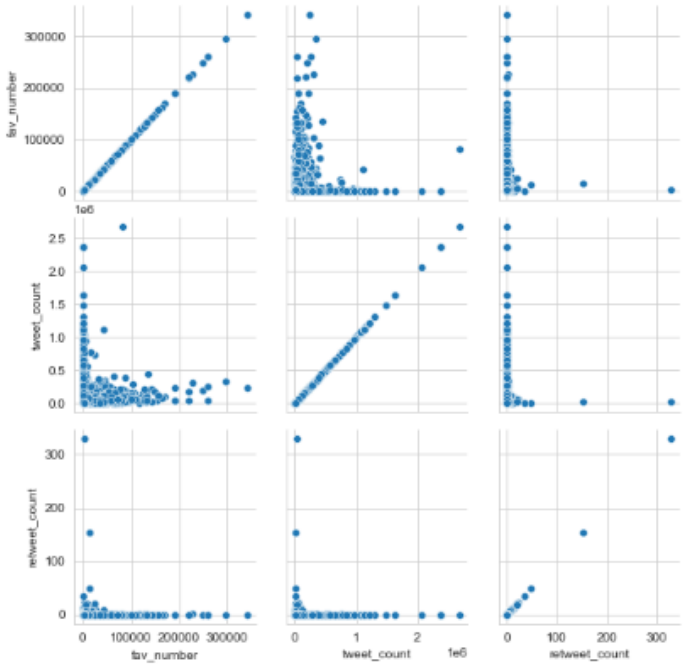

Fig 20: Pairgrid for fav_number, tweet_count, retweet_count

Each variable in the dataset is mapped using it to the columns and rows of several axes. In order to distinguish between distinct colors in the plot, we can also employ different levels of conditionalization known as hue.

Violinplot:

```
Out[85]: <AxesSubplot:xlabel='fav_number', ylabel='tweet_count'>
```

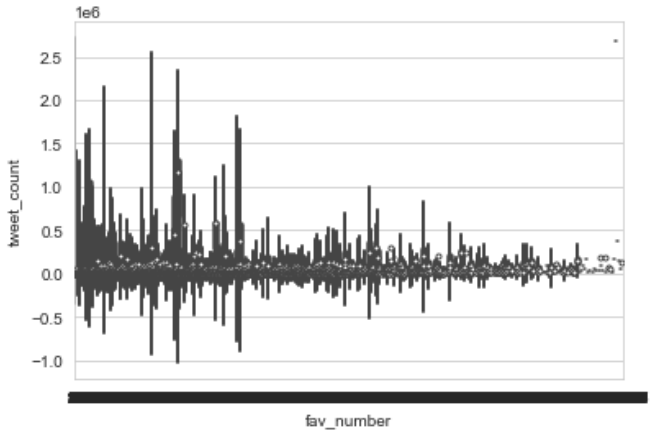

Fig 21: Violin plot for fav_number and tweet_count

```
Out[86]: <AxesSubplot:xlabel='fav_number', ylabel='gender_new'>
```

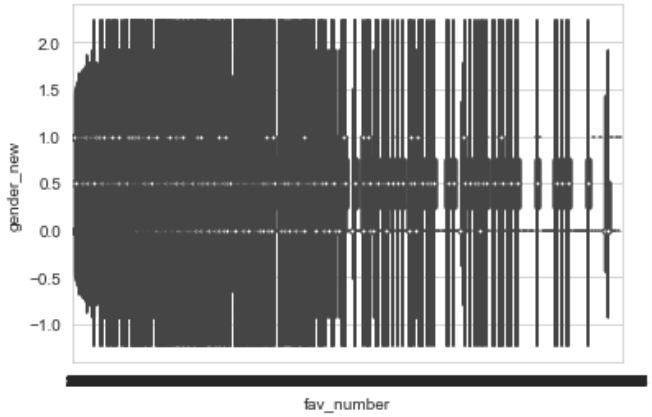

Fig 22: Violin plot for fav_number and gender_new

The box plot and the violin plot are both used to present a variety of facts in a visually appealing way. It is used to maintain each column of numerical data.

Distplot:

```
Out[71]: <AxesSubplot:xlabel='fav_number', ylabel='Density'>
```

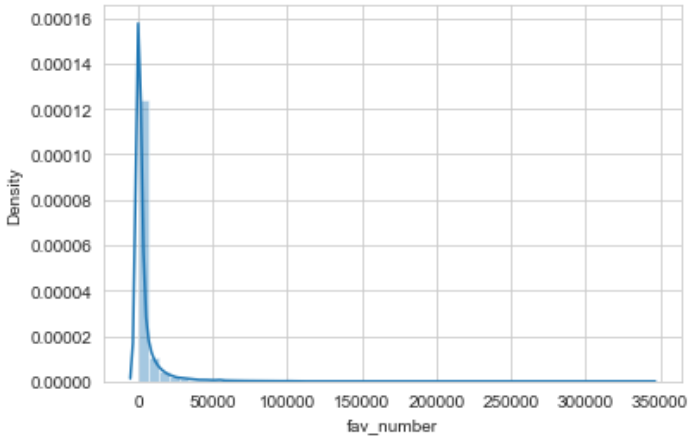

Fig 23: Distplot for fav_number

As

```
Out[72]: <AxesSubplot:xlabel='tweet_count', ylabel='Density'>
```

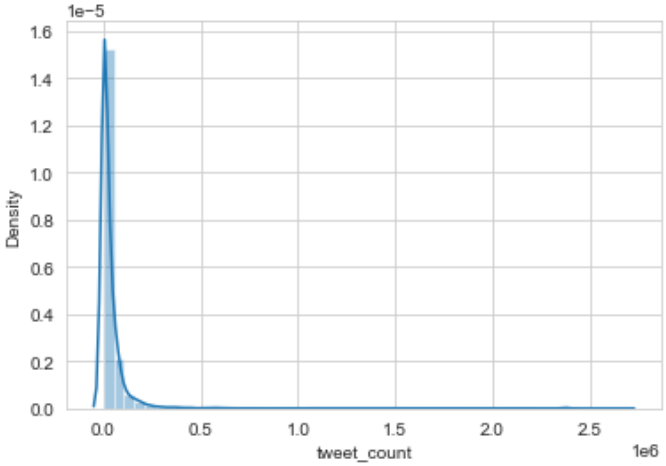

Fig 24 : Distplot for tweet_count

```
Out[73]: <AxesSubplot:xlabel='retweet_count', ylabel='Density'>
```

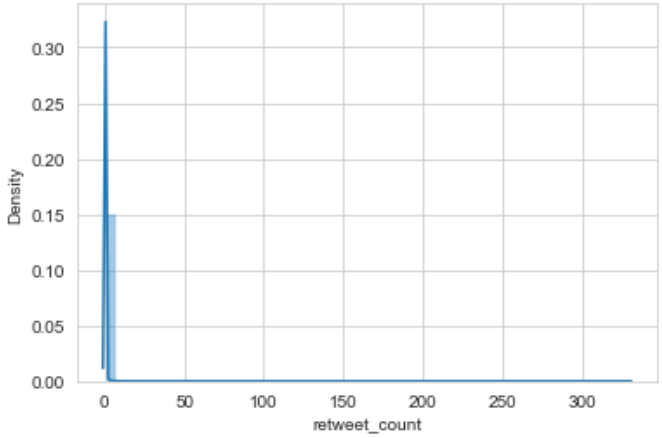

Fig 25 : Distplot for retweet_count

It uses a histogram to graphically represent the observations. It depicts the general distribution of variables in continuous data. As we can see above, we can observe where the most of the data is concentrated. It can also be a visual way of detecting the outliers, although not very accurate.

Jointplot:

```
Out[80]: <seaborn.axisgrid.JointGrid at 0x1f71dadb400>
```

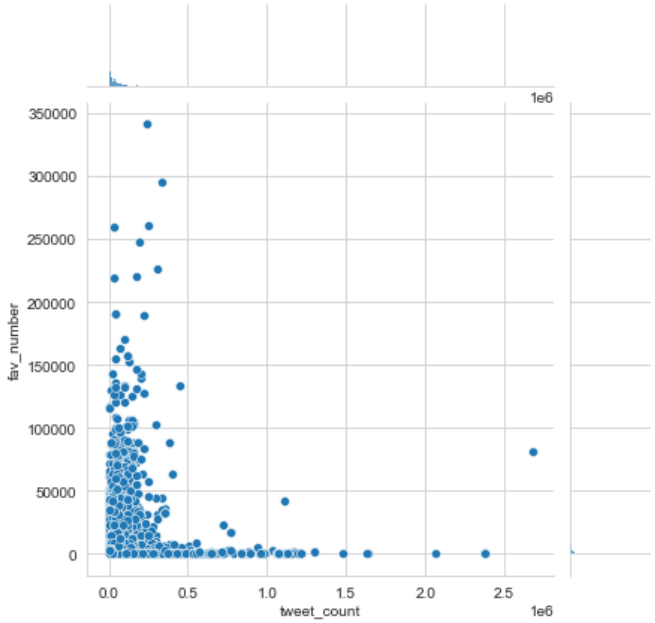

Fig 26: Joint plot for tweet_count and fav_number

```
Out[94]: <seaborn.axisgrid.JointGrid at 0x1f7a56033a0>
```

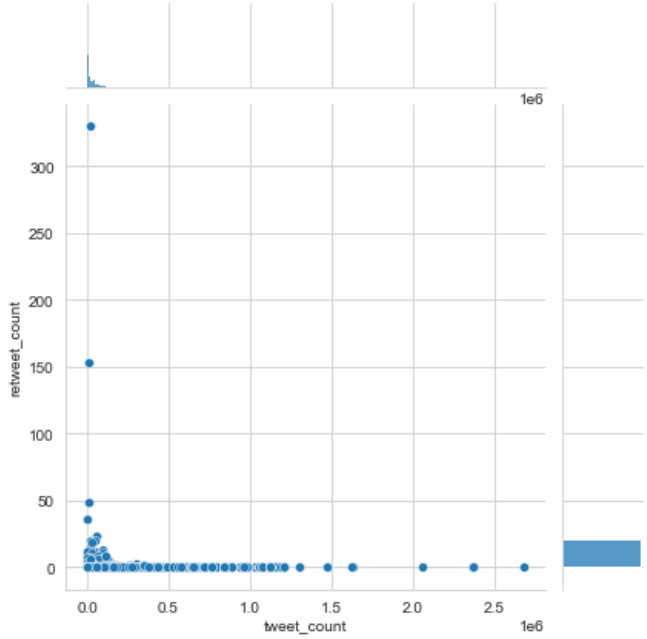

Fig 27: Joint plot for tweet_count and retweet_count

```
Out[81]: <seaborn.axisgrid.JointGrid at 0x1f71e6a4640>
```

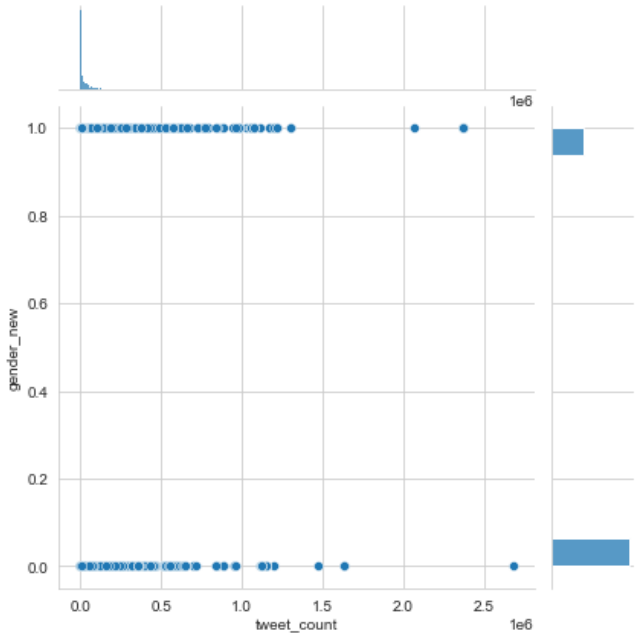

Fig 28: Joint plot for tweet_count and gender_new

It is used to plot the graph with respect to two variables, resulting in a bivariate or univariate graph.

Task 4

gender_new	Tweet_avg	retweet_avg
Humans	29986.434151	0.063935
Non-Humans	60146.667452	0.116123

Tweet_avg and retweet_avg shows the average number of times retweets have been done by Humans and Non-Humans.

As we can see from the above table, non-Human accounts usually have a higher number of tweets and retweets in comparison to Human accounts.

gender_new	sidebar_color_avg
Humans	320.849164
Non-Humans	327.842646

After we label encoded Color, we took the average of the color by each gender class.

Since color after label encoding is in a similar range for both gender classes, color wouldn't be a good decision variable for distinguishing them.