# MAJOR PROJECT

ENSEMBLE DATA MODELING

BATCH – ML052B12

WORK FLOW:


STEP-1
• DATA CLEANING
STEP-2
• DATA VISUALISATION
STEP-3
• QUERYING THE DATASET ALONG WITH ANSWERS
STEP-4
• FINDING ACCURACY THROUGH RANDOM FOREST CLASSIFICATION ALGORITHM.
STEP-5
• FINDING ACCURACY THROUGH SVM CLASSIFICATION ALGORITHM.
STEP-6
• FINDING ACCURACY THROUGH NAIVE BAYES CLASSIFICATION ALGORITHM.
STEP-7
• DECLARING THE ALGORITHM WITH BEST ACCURACY.

## DATASET USED:

## Twitter User Gender Classification

This data set was used to train a CrowdFlower AI gender predictor. Contributors were asked to simply view a Twitter profile and judge whether the user was a male, a female, or a brand (non-individual). The dataset contains 20,000 rows, each with a user name, a random tweet, account profile and image, location, and even link and sidebar color.

The dataset contains the following fields:

- **_unit_id**: a unique id for user
- **_golden**: whether the user was included in the gold standard for the model; TRUE or FALSE
- **_unit_state**: state of the observation; one of finalized (for contributor-judged) or golden (for gold standard observations)
- **_trusted_judgments**: number of trusted judgments (int); always 3 for non-golden, and what may be a unique id for gold standard observations
- **_last_judgment_at**: date and time of last contributor judgment; blank for gold standard observations
- **gender**: one of male, female, or brand (for non-human profiles)
- **gender:confidence**: a float representing confidence in the provided gender
- **profile_yn**: "no" here seems to mean that the profile was meant to be part of the
- dataset but was not available when contributors went to judge it
- **profile_yn:confidence**: confidence in the existence/non-existence of the profile
- **created**: date and time when the profile was created
- **description**: the user's profile description
- **fav_number**: number of tweets the user has favorited
- **gender_gold**: if the profile is golden, what is the gender?
- **link_color**: the link color on the profile, as a hex value
- **name**: the user's name
- **profile_yn_gold**: whether the profile y/n value is golden
- **profileimage**: a link to the profile image
- **retweet_count**: number of times the user has retweeted (or possibly, been retweeted)
- **sidebar_color**: color of the profile sidebar, as a hex value
- **text**: text of a random one of the user's tweets
- **tweet_coord**: if the user has location turned on, the coordinates as a string with the format "[latitude, longitude]"
- **tweet_count**: number of tweets that the user has posted
- **tweet_created**: when the random tweet (in the text column) was created
- **tweet_id**: the tweet id of the random tweet
- **tweet_location**: location of the tweet; seems to not be particularly normalized
- **user_timezone**: the timezone of the user

# Sample data in the dataset:

| _unit_id | _golden | _unit_state | _trusted_judgments | gender | gender:confidence | profile_yn | profile_yn:confidence | created | description | fav_number |
|---|---|---|---|---|---|---|---|---|---|---|
| 815719226 | False | finalized | 3 | male | 1.0000 | yes | 1.0 | 12/5/13 1:48 | i sing my own rhythm. | 0 |
| 815719227 | False | finalized | 3 | male | 1.0000 | yes | 1.0 | 10/1/12 13:51 | I'm the author of novels filled with family dr... | 68 |
| 815719228 | False | finalized | 3 | male | 0.6625 | yes | 1.0 | 11/28/14 11:30 | louis whining and squealing and all | 7696 |
| 815719229 | False | finalized | 3 | male | 1.0000 | yes | 1.0 | 6/11/09 22:39 | Mobile guy. 49ers, Shazam, Google, Kleiner Pe... | 202 |
| 815719230 | False | finalized | 3 | female | 1.0000 | yes | 1.0 | 4/16/14 13:23 | Ricky Wilson The Best FRONTMAN/Kaiser Chiefs T... | 37318 |

| link_color | name | profileimage | retweet_count | sidebar_color | text | tweet_count | tweet_created | tweet_id |
|---|---|---|---|---|---|---|---|---|
| 08C2C2 | sheezy0 | https://pbs.twimg.com/profile_images/414342229... | 0 | FFFFFF | Robbie E Responds To Critics After Win Against... | 110964 | 10/26/15 12:40 | 6.587300e+17 |
| 0084B4 | DavdBurnett | https://pbs.twimg.com/profile_images/539604221... | 0 | C0DEED | □ÛÏIt felt like they were my friends and I was... | 7471 | 10/26/15 12:40 | 6.587300e+17 |
| ABB8C2 | lwtprettylaugh | https://pbs.twimg.com/profile_images/657330418... | 1 | C0DEED | i absolutely adore when louis starts the songs... | 5617 | 10/26/15 12:40 | 6.587300e+17 |
| 0084B4 | douggarland | https://pbs.twimg.com/profile_images/259703936... | 0 | C0DEED | Hi JordanSpieth Looking at the url do you... | 1693 | 10/26/15 12:40 | 6.587300e+17 |
| 3B94D9 | WilfordGemma | https://pbs.twimg.com/profile_images/564094871... | 0 | 0 | Watching Neighbours on Sky catching up with t... | 31462 | 10/26/15 12:40 | 6.587300e+17 |

# Data Cleaning:

Data Cleaning means the process of identifying the incorrect, incomplete, inaccurate, irrelevant or missing part of the data and then modifying, replacing or deleting them according to the necessity.

As much as you make your data clean, as much as you can make a better model. So, we need to process or clean the data before using it.

To proceed with the application of classification algorithms data must be filtered, keeping only the necessary rows and columns.

In our project, we did the following functions as a part of cleaning the data.

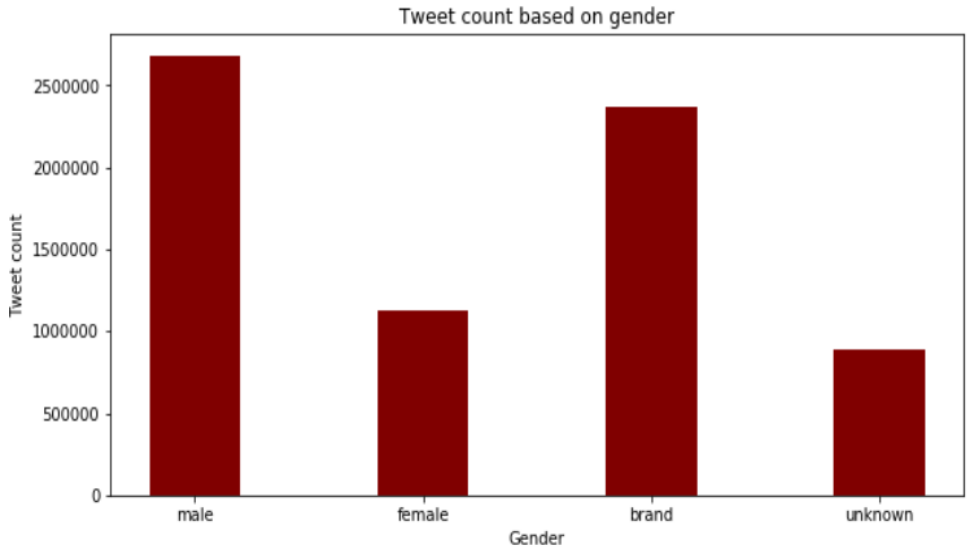
- **Dropping the columns which are of less significance like "profile_yn_gold", "gender_gold", "tweet_location","user_timezone","tweet_coord", "_last_judgment_at".**
- **Dropping duplicate values**
- **Dropping null values in gender and gender:confidence columns as they are of no significance with gender being the dependant variable**
- **Replacing null values in “text” column with empty string**
- **Replacing null values in “description” column with empty string**
- **Removing the special characters and hyperlinks in text column using regular Expressions**
- **Converting Categorical data present in columns like '_golden', '_unit_state', 'gender', 'profile_yn', 'created', 'description', 'link_color', 'name', 'profileimage', 'sidebar_color', 'text', 'tweet_created' to Numerical data using label encoder.**

# Data visualization:

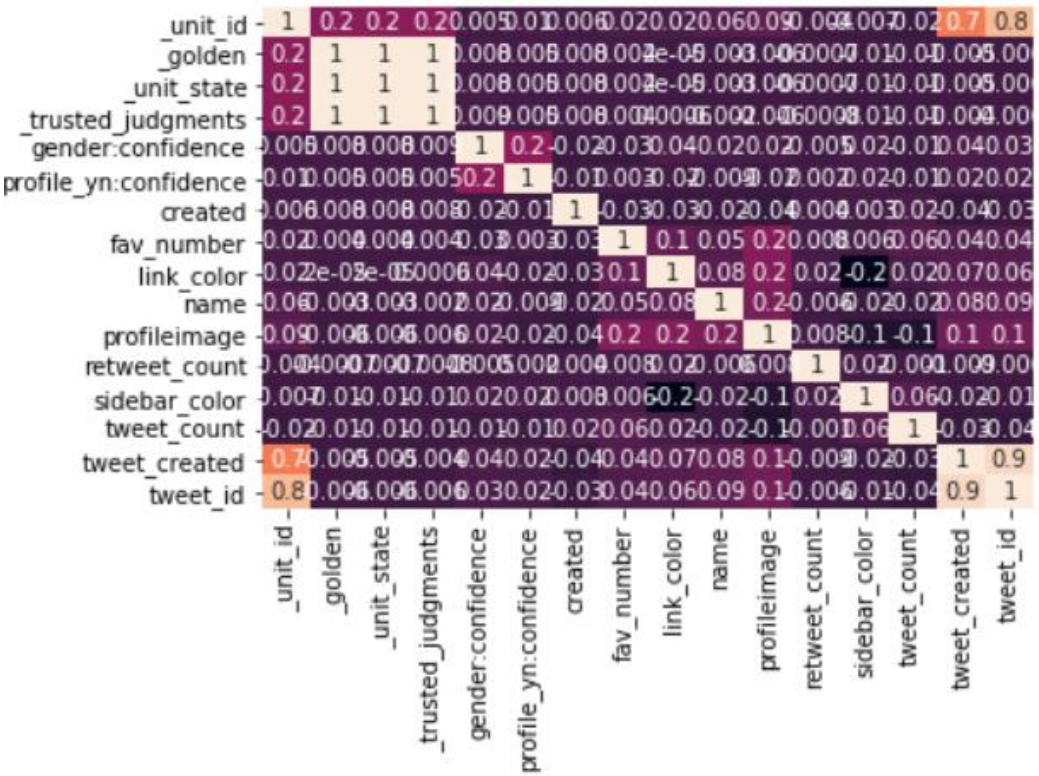
Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

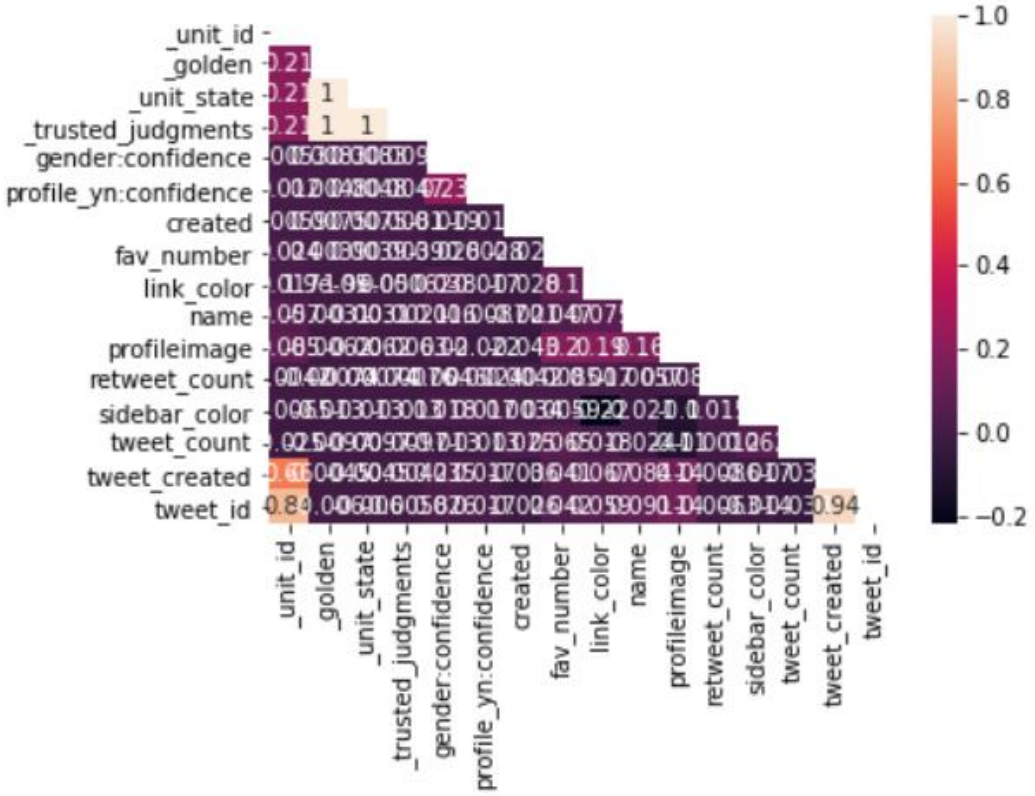
In our project, we used bar plots, heatmap as part of data visualization.

## 1) Bar plot against gender and tweet count:



## 2) Seaborn heatmap for a correlation matrix:



# Querying the dataset along with answers:

We selected two questions for querying and answered using the dataset.

### 1) Which gender has more gender confidence?

| gender | mean_con |
|---|---|
| female | 0.92636 |

After analysing the dataset, it can be concluded that gender=female has more gender confidence.

### 2) Which gender has more average tweet count?

| gender | max_tweet |
|---|---|
| brand | 60146.667452 |

After analysing the dataset, it can be concluded that gender=brand has more average tweet count.

# Classification algorithms opted to find accuracy on this dataset :

# 1) Random Forest

# 2) Support Vector Machine

# 3) Naive Bayes

# Accuracy using Random Forest Algorithm:

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

In the project, to find the accuracy using Random Forest Algorithm, we followed the below steps.

1. Imported random forest model.
2. Selected the
   Dependent variable - ('gender')
   independent variables - ('unit_id', '_golden', 'unit_state', 'trusted_judgments', 'gender_confidence', 'created', 'fav_number', 'name', 'profileimage', 'tweet_created', 'tweet_id', 'sidebar_color', 'link_color', 'profile_yn:confidence', 'retweet_count', 'tweet_count')
   of the model.
3. Divided the data into training and test set through train_test_split function from sklearn library.
4. Test_size=0.3 means that 30% data will be test data and 70% data will be training data. Set the random state so that different rows dont go to training and testing data.
5. Set n_estimators and max depth to 100 and 10 respectively.
6. Fit the model with the training data, and predict on the testing data.
7. Import accuracy_score method from sklearn.metrics and calculate score using suitable arguments to the method.
8. After acquiring an accuracy of 63%, since Random Forest is prone to overfitting , tuning is done through hyperparameters.
9. In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node. (The parameters of a random forest are the variables and thresholds used to split each node learned during training).
10. After hyperparameter tuning , the accuracy is 65 %.

```
grid_predictions = grid.predict(X_test)

print("Accuray is \n", accuracy_score(Y_test, grid_predictions))
```

```
Accuray is
 0.6577927112949082
```

**The Accuracy of Random Forest Model is 65 %**

# Accuracy using Naive Bayes Algorithm:

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

In the project, to find the accuracy using Naïve Bayes Algorithm, we followed the below steps.

1. First we check the text and then remove all the stopwords from text (since stopwords are present in abundance so they provide little to no information) (The basic idea is that stopwords are useless from the perspective of classification and should be ignored; they are just too common to provide much information.)
2. The cleaned text is then put in a separate new column named clean_texts.
3. Then we create a sparse matrix using the clean_text column using CountVectorizer().
4. The **CountVectorizer** provides a simple way to both tokenize a collection of text documents and build a vocabulary of known words, but also to encode new documents using that vocabulary.
5. Then the train_test_split is done on the sparse data and "gender" with "gender" being the dependant variable.
6. Now we fit the model into multinomial naive Bayes technique (we are using multinomial naive Bayes technique as we are dealing with text document).
7. The term **Multinomial Naive Bayes** simply lets us know that each node is a **multinomial** distribution, rather than some other distribution. This works well for data which can easily be turned into counts, such as word counts in text.
8. Import accuracy_score method from sklearn.metrics and calculate score using suitable arguments to the method.
9. Then the accuracy score is checked which turns out to be 61%.

```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
clf.fit(X_train, y_train)
predicted = clf.predict(X_test)
from sklearn.metrics import accuracy_score
accuracy_score(y_test, predicted)
```

```
3]: 0.6166253101736973
```

**Accuracy of Naive Bayes model is 61 %**

## Accuracy using SVM algorithm:

**SVM** is a supervised machine learning algorithm which can be **used for** classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.

Support Vectors are simply the co-ordinates of individual observation.

In the project, to find the accuracy using SVM Algorithm, we followed the below steps.

1. Extracted the required feature alone from data that will more fit the MODEL. The features taken are 'gender:confidence', 'created', 'profileimage', 'tweet_id', 'sidebar_color', 'link_color', 'retweet_count', 'tweet_count'. We consider these are the features required for identifying gender.
2. The dataset is split into a test and train subset with test_size as 0.2 and train_size be 0.8. Whereas random_state is set to 100, the random_state parameter is used for initializing the internal random number generator, which will decide the splitting of data into train and test indices.
3. Choosing between SVC and LinearSVC. Linear Support Vector Classification.
4. Similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples.This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.
5. Once the data is split, we create a LinearSVC model. Before creating a model, it would be better to standardise the data. Standardization can be done in Scikitlearn using StandardScalar:It transforms the data in such a manner that it has mean as 0 and standard deviation as 1. In short, it standardizes the data. Standardization is useful for data which has negative values.
6. Parameters of LinearSVC: penalty was set to l2, since l2 norm of penalization is better for the data, tol( Tolerance for stopping criteria. ) is set to 1e-5 and dual is set False to avoid convergence warning intercept_scaling was set to 43.9. Since the data is not already centered, fit_intercept is not set to false.
7. After creating the model, the train data is fit into the model. Since the batch size of both train data is to be the same, the data with different batch size is taken transposed using ravel.
8. Actual predicted value of the model is found and accuracy score between actual predicted and the output we have is 63.16%.
9. This is the best accuracy we achieved so far. Tuning the model with GridSearchCV and RandomSearchCV produced a accuarcy of 63.16% and 62.1% respectively. When the penalty was set to l1, with changes in other parameters the maximum accuracy reached is 62.04%

```
score = accuracy_score(op_test,actual)
score
```

```
]: 0.6316401706087631
```

**SVM model accuracy is 63.16%.**

## DECLARING THE ALGORITHM WITH BEST ACCURACY:

From the observation done on the accuracy of each algorithm opted, following can be deduced.

| ALGORITHM | ACCURACY |
|---|---|
| **RANDOM FOREST ALGORITHM** | **65.7%** |
| **SUPPORT VECTOR MACHINE ALGORITHM** | **63.16%** |
| **NAÏVE BAYES ALGORITHM** | **61.6%** |

**FROM THIS, WE HAVE CONCLUDED THAT RANDOM FOREST GAVE THE HIGHEST ACCURACY AMONG THE THREE ALGORITHMS FOR THE GIVEN DATASET.**