# Problem statement : Fake social media detection and reporting.

# Problem I'd : PS010005

# College : Madhuben and Bhanubhai Patel Institute of Technology -CVM University

# Introduction

Have you ever purchased a product or service based on the nomination of the owner of a YouTube channel you follow or a blogger on Instagram? If you did... it means that you have fallen under the influence of influencer marketing.

Who is the influencer? An influencer is someone who has the ability to influence the buying decisions of others because of their power, knowledge, position, or relationship with their audience. It is important to note that these individuals are not just marketing tools, they are social networking assets with which brands can collaborate to achieve their marketing objectives.

From here, the concept of influencer marketing is to exploit the best content makers on different platforms, celebrities and influencers in a particular field to spread awareness among people and promote a particular product or service. The challenge that promoters and marketing experts faced at the time is their scepticism about the validity of this influencer depending on a number of factors. Among the challenges - many - that have occurred in the influencer marketing arena since its appearance in 2006, is the emergence of social media platforms little by little in the picture such as Facebook and Twitter, which Instagram joined later in 2010, According to the January 2019 We Are Social Report, 3.484 billion people actively use social media - that's 45% of the world's population. These people inevitably look to social media influencers to guide their decisions.

You can separate different types of social media influencers in multiple ways. Some of the most common methods are through the number of followers. Big influencers are people who have a large number of followers on their social networks, which made some influencers manipulate the numbers of followers and interaction rates in social media. From here came the idea of our project, which is the automated classification of Fake/ Not Fake accounts using machine learning capabilities.

# Question/Need

- What are the features we are interested in?
- Which features have more impact with (on) the target feature?
- What features do we need to activate the tools?
- We need a predictive model for the classification of fake / not fake accounts.
- We need tools to prove the approval of the model.
- Model overview.

# Framework

## Description of the framework proposed

The model is a classification model, which means that it is predicting the class of given data points dependent variable using more than one explanatory variable. There are two essential features (independent variable) that are highly correlated with the Classification of accounts (dependent variable). Features Considered Include. The goal is to find attributes that have a strong association with real accounts.

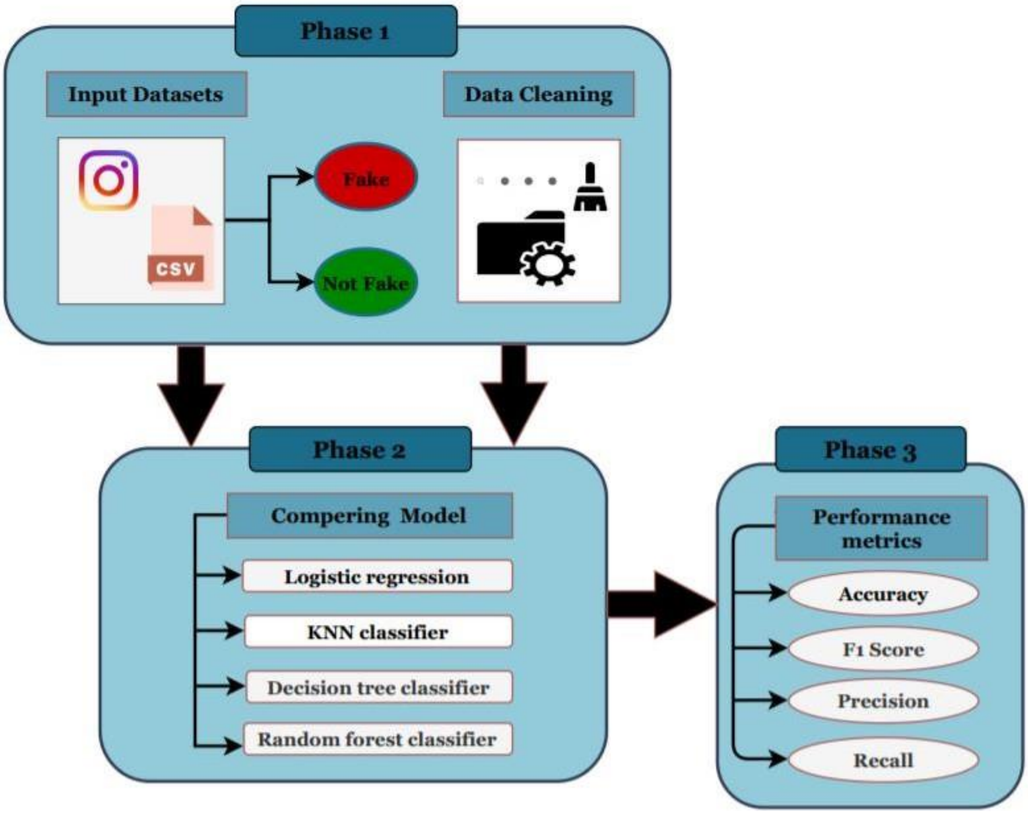


Figure 1: Phases of the proposed framework.

# Phase1:

## step 1: Dataset

- The data set consists of 12 columns.

| Column name | Description |
|---|---|
| 1. Profile pic | Does the user have a profile picture if their profile pic=1, else profile pic=0? |
| 2. Nums/length username | Ratio of numerical to alphabetical characters in the username |
| 3. Fullname words | How many words are in the user's full name? |

| | |
|---|---|
| 4. Nums/length fullname | Ratio of numerical to alphabetical characters in the full name |
| 5. Name==username | Is the user's full name the same as the username? |
| 6. Description length | How many characters is in the user's Instagram bio? |
| 7. External URL | Does the user have an external URL linked to their profile? |
| 8. Private | Is the user private? |
| 9. Posts | Number of posts |
| 10. Followers | Number of people following the user |
| 11. Follows | Number of people the user follows |
| 12. Fake | If the user is fake, fake=1, else fake=0 |

Table1: dataset consists.

## step 2: EDA and data cleaning

Data cleaning is an essential part of the project. It upgrades the data quality and increases the overall productivity. When we clean the data, all outdated or incorrect information is removed. In this phase, we discover the dataset and see if we should fix data or find if there are any incorrect format, duplicate, or incomplete data within a dataset. We imported the data then inspect the dataset and find out:

1. Describe: count, mean, std, etc
2. Information
3. Shape
4. Check the null value
5. Create a correlation matrix for the features in the training data to check for significantly relevant features.

# Phase 2: Comparing Models

Recent Machine Learning techniques are widely available for various purposes. However, it is not yet clear which classifier is best suited to a given set of data to accommodate for this, the work is validated using well-known classifiers. In this work, we suggested using four classifiers and comparing the results to find out the best classifier for our problem.

## 1. Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for classification problems, it is a predictive analysis algorithm based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the 'Sigmoid function' or also known as the 'logistic function' instead of a linear function.

## 2. KNN

The k-nearest neighbours (KNN) algorithm is a data classification method for estimating the likelihood that a data point will become a member of one group or another based on what group the data points nearest to it belong to. The k-nearest neighbour algorithm is a type of supervised machine learning algorithm used to solve classification and regression problems. However, it's mainly used for classification problems.

### 3. Decision Tree

The goal of using a Decision Tree is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

### 4. Random Forest

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and, based on the majority votes of predictions, and it predicts the final output.

## Phase 3: Performance Metrics

Finally, after data cleaning, pre-processing the first step we do is to feed it to an outstanding model and, of course, get output in probabilities. To enhance the effectiveness and the performance. That is where the Confusion matrix comes into the limelight. Confusion matrix is a performance measurement for machine learning classification. So, we used the confusion matrix to calculate the Accuracy, Precision, Recall, and F1-Score. to build up a summary picture of the classification performance.

*Accuracy = TruePositive (TP) + TrueNegative (TN)/ (Total No of Samples) * 100 (1)*

***Precision estimates how many positive labels we had predicted.***

*Precision = TruePositive (TP)/ (TruePositive (TP) + FalsePositive (FP) (2)*

***Recall evaluates how many positive labels we had correctly predicted from our data***

*Recall = TruePositive (TP)/ (TruePositive (TP) + FalseNegative (FN) (3)*

***While F1-Score is the weighted mean of Recall and Precision.***

$F1 - Score = 2 \times (Precision \times Recall)/ (Precision + Recall)$ (4)

# Result

We have now reached the last stage, which is the results, and before we talk about the results, let's look at the description of the data after it has been divided. The data obtained is balanced by plotting the number of results according to their value for each of the Training dataset and the Test dataset.

| | profile pic | nums/length username | fullname words | nums/length fullname | name==username | description length | external URL | private | #posts | #followers | #follows |
|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 576.000000 | 576.000000 | 576.000000 | 576.000000 | 576.000000 | 576.000000 | 576.000000 | 576.000000 | 576.000000 | 5.760000e+02 | 576.000000 |
| mean | 0.701389 | 0.163837 | 1.460069 | 0.036094 | 0.034722 | 22.623264 | 0.116319 | 0.381944 | 107.489583 | 8.530724e+04 | 508.381944 |
| std | 0.458047 | 0.214096 | 1.052601 | 0.125121 | 0.183234 | 37.702987 | 0.320886 | 0.486285 | 402.034431 | 9.101485e+05 | 917.981239 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000e+00 | 0.000000 |
| 25% | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 3.900000e+01 | 57.500000 |
| 50% | 1.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 9.000000 | 1.505000e+02 | 229.500000 |
| 75% | 1.000000 | 0.310000 | 2.000000 | 0.000000 | 0.000000 | 34.000000 | 0.000000 | 1.000000 | 81.500000 | 7.160000e+02 | 589.500000 |
| max | 1.000000 | 0.920000 | 12.000000 | 1.000000 | 1.000000 | 150.000000 | 1.000000 | 1.000000 | 7389.000000 | 1.533854e+07 | 7500.000000 |

Table2: Train description.

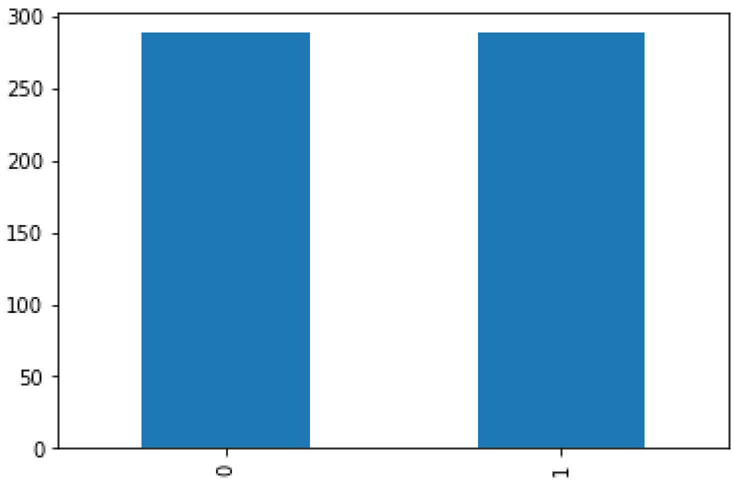

Figure2: Count of 'fake' value counts in Training data.

| | profile pic | nums/length username | fullname words | nums/length fullname | name==username | description length | external URL | private | #posts | #followers | #follows |
|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 120.000000 | 120.000000 | 120.000000 | 120.000000 | 120.000000 | 120.000000 | 120.000000 | 120.000000 | 120.000000 | 1.200000e+02 | 120.000000 |
| mean | 0.758333 | 0.179917 | 1.550000 | 0.071333 | 0.041667 | 27.200000 | 0.100000 | 0.308333 | 82.866667 | 4.959472e+04 | 779.266667 |
| std | 0.429888 | 0.241492 | 1.187116 | 0.209429 | 0.200664 | 42.588632 | 0.301258 | 0.463741 | 230.468136 | 3.816126e+05 | 1409.383558 |
| min | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000e+00 | 1.000000 |
| 25% | 1.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 6.725000e+01 | 119.250000 |
| 50% | 1.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 8.000000 | 2.165000e+02 | 354.500000 |
| 75% | 1.000000 | 0.330000 | 2.000000 | 0.000000 | 0.000000 | 45.250000 | 0.000000 | 1.000000 | 58.250000 | 5.932500e+02 | 668.250000 |
| max | 1.000000 | 0.890000 | 9.000000 | 1.000000 | 1.000000 | 149.000000 | 1.000000 | 1.000000 | 1879.000000 | 4.021842e+06 | 7453.000000 |

Table3: Test description.

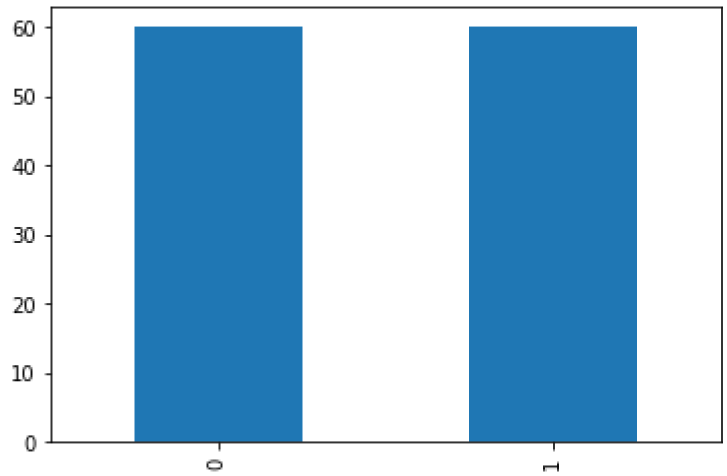


Figure3: Count of 'fake' value counts in Testing data.

To check for significantly relevant features, we need measures that we can use on the data to select the right features. used a correlation matrix for the features in the Training data to find Features with high correlation are more linearly dependent and hence have almost the same effect on the dependent variable. So, when two features have high correlation, we can drop one of the two features.

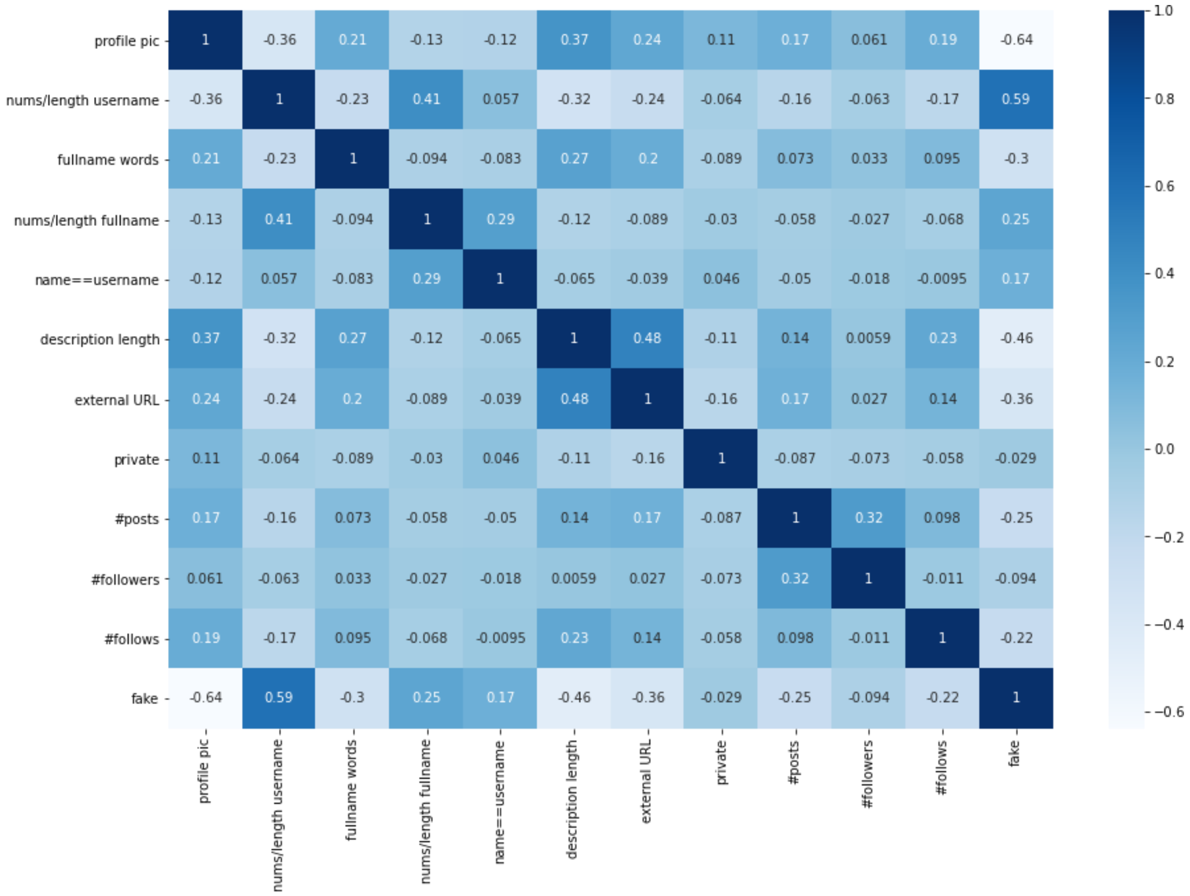


Figure4: features correlation matrix in the training data.

Next, we compare the correlation between features and find that most features have high correlations with the target and the most feature is nums/length fullname.

# Evaluation

At the end of building the model, we obtained the results based on the results of the model tests on several scales, including: a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class. The name stems from the fact that it makes it easy to see whether the system is confusing two classes (fake /not fake account).

# confusion matrix

It was applied to each built model and these are the results for each model separately:

**Logistic Regression**

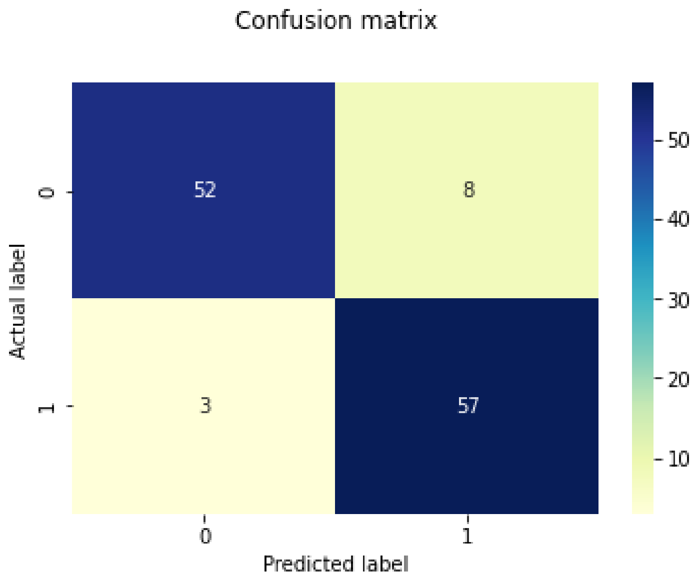


Figure5:LR_confusion matrix.

**KKN Classifier**

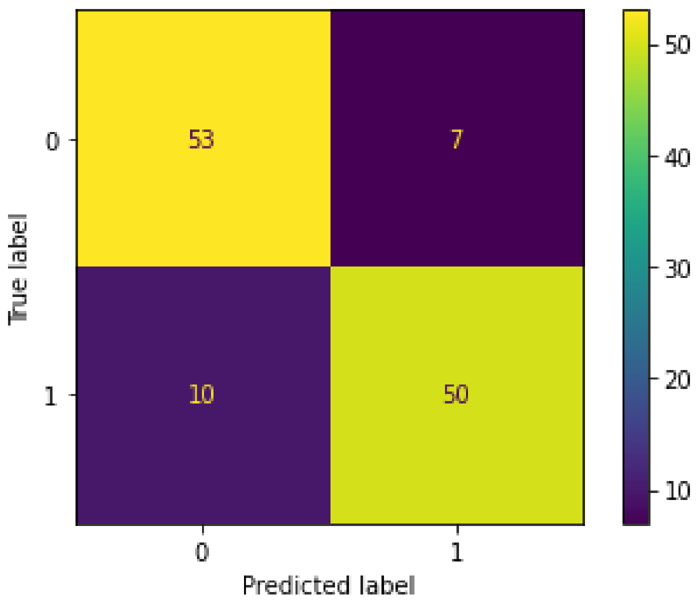


Figure6: KNN_confusion matrix.

### Decision Tree Classifier

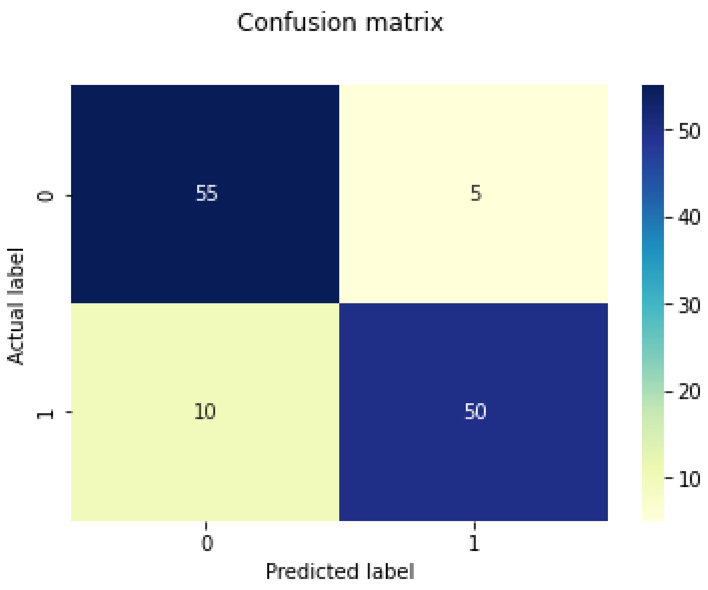


Figure7: DT_confusion matrix.

### Random Forest Classifier

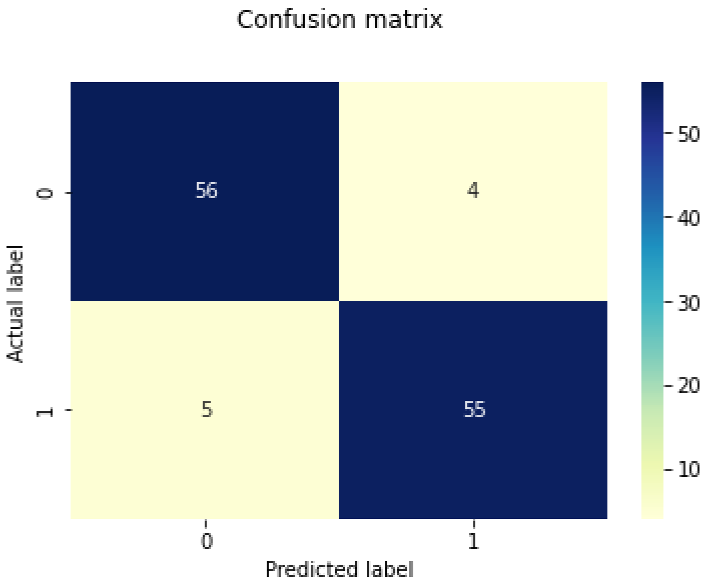


Figure8: RF_confusion matrix.

# Accuracy Comparing

After creating a confusion matrix which is required to compute the accuracy of the machine learning algorithm in classifying the data into its corresponding labels, the next step in the process is to calculate its predictive power. In order to calculate these statistics, we'll need to split our data into a training and validation set across multiple scales of evaluation metric. Here, the table presents a comparison of the results of calculating several evaluation metrics.

| Classifier | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| **Logistic Regression** | 0.908 | 0.95 | 0.876 | 0.912 |
| **KNN Classifier** | 0.866 | 0.833 | 0.877 | 0.854 |
| **Decision Tree Classifier** | 0.875 | 0.833 | 0.909 | 0.869 |
| **Random Forest Classifier** | 0.925 | 0.916 | 0.932 | 0.924 |

Table4: Comparison several evaluation metrics.

And through the comparison from the table above and based on the evaluation metrics. We found that the model **Random Forest Classifier** Better performance and prediction than other models.

# Future Work

One of the future directions that we identified upon completion of the project is to first implement the mechanism on other social media such as Facebook, in order to take advantage of the mechanism for several applications. And other analyses of classification such as classification of followers by geographic region or age group and other classifications. By thinking about advertisers or followers and to make it easier for them to find real influencers, we put in future plans to create a mechanism in application form.