Modeling for Covid-19 misinformation using Machine Learning - Final Report

- ***Team 5***

Abstract

Facebook, Instagram, and Twitter, among other social media sites, have become an inextricable part of our everyday lives. These social media networks are useful for sharing news, images, and other types of information. Apart from the benefits of ease, these platforms are frequently used to spread dangerous material or information. This misinformation has the potential to mislead users and have a negative impact on society's culture, economy, and healthcare. It is tough to combat the spread of such a large volume of disinformation. As a result, the dissemination of misinformation about the COVID-19 pandemic, its treatment, and vaccine may pose serious issues for frontline workers in each country.

Hence, developing an effective machine-learning (ML) misinformation-detection model for detecting COVID-19 misinformation is critical. Through our work, we want to leverage such machine learning models develop such a model that can identify this misinformation, and subsequently prevent this misinformation from spreading.

Problem Statement

We want to develop a classification model that identifies whether a COVID-19 related tweet is fake or not by analyzing the various features of a tweet such as retweets, reply counts, likes counts, user data – followers, verified status, friends count. Account life, total number of tweets, sentiment of tweet

Data Extraction and Data Preprocessing

CoAID (Covid-19 healthcare misinformation Dataset) is a diverse COVID-19 healthcare misinformation dataset, including fake news on websites and social platforms, along with users' social engagement about such news. It includes 5,216 news, 296,752 related user engagements, 958 social platform posts about COVID-19, and ground truth labels.

We have extracted the dataset and preprocessed in the DataFrame. This dataset provides clear differentiation on the Real and Fake in Claims and News on Twitter from the fact-check website.

Steps followed to scrape the dataset:

```
In [ ]: client = tweepy.Client(bearer_token='AAAAAAAAAAAAAAAAAAAAALADbQEAAAAAWuhDzDRixlqbghW2sy7HGbU2wwE%3Dk9wrvfmwRmGfI6hE6
        ClaimFakeCOVID_19 = pd.read_csv("./05-01-2020/ClaimFakeCOVID-19.csv")
        ClaimFakeCOVID_19_tweets = pd.read_csv("./05-01-2020/ClaimFakeCOVID-19_tweets.csv")
        ClaimFakeCOVID_19_tweets_replies = pd.read_csv("./05-01-2020/ClaimFakeCOVID-19_tweets_replies.csv")
        ClaimRealCOVID_19 = pd.read_csv("./05-01-2020/ClaimRealCOVID-19.csv")
        ClaimRealCOVID_19_tweets = pd.read_csv("./05-01-2020/ClaimRealCOVID-19_tweets.csv")
        ClaimRealCOVID_19_tweets_replies = pd.read_csv("./05-01-2020/ClaimRealCOVID-19_tweets_replies.csv")
        NewsFakeCOVID_19 = pd.read_csv("./05-01-2020/NewsFakeCOVID-19.csv")
        NewsFakeCOVID_19_tweets = pd.read_csv("./05-01-2020/NewsFakeCOVID-19_tweets.csv")
        NewsFakeCOVID_19_tweets_replies = pd.read_csv("./05-01-2020/NewsFakeCOVID-19_tweets_replies.csv")
        NewsRealCOVID_19 = pd.read_csv("./05-01-2020/NewsRealCOVID-19.csv")
        NewsRealCOVID_19_tweets = pd.read_csv("./05-01-2020/NewsRealCOVID-19_tweets.csv")
        NewsRealCOVID_19_tweets_replies = pd.read_csv("./05-01-2020/NewsRealCOVID-19_tweets_replies.csv")
```

Step 1: Claim Real Dataset

```
In [4]: Claim_Real_Covid_data = pd.DataFrame (heavy_structure, columns = ['tweet_id', 'tweet_text', 'retweet_count', 'reply_
        Claim_Real_Covid_data["Real_Fake"]=1
        Claim_Real_Covid_data
```

Out[4]:

	tweet_id	tweet_text	retweet_count	reply_count	like_count	Real_Fake
0	1253283636843089920	How large does a meeting or event need to be i...	0	1	0	1
1	1253965710520397828	@Cordobesa2201 @Rumpelstinski6 @sanidadgob Doe...	0	0	0	1
2	1253959004436467713	@Cordobesa2201 @Rumpelstinski6 @sanidadgob Doe...	0	0	0	1
3	1252584911183249409	@AngryDuck91 MASS GATHERING\n- Does WHO recomm...	0	0	0	1
4	1252577361545043968	@AngryDuck91 - Does WHO recommend that all int...	0	0	0	1
...	...	...	...	...	...	...
5491	1222182262130823168	Are there any specific medicines to prevent or...	0	2	2	1
5492	1222091107611353088	Q: Are there any specific medicines to prevent...	0	1	4	1
5493	1221843359305818115	@WHOPhilippines @WHO_Mongolia @takeshi_kasai @...	253	1	437	1
5494	1221841094050230278	Q: Are there any specific medicines to prevent...	0	0	1	1
5495	1221840413700562944	@WHOWPRO @WHOSEARO @WHO_Europe @pahowho @WHOEM...	941	40	1193	1

5496 rows × 6 columns

Step 2: Claim Fake Dataset

```
Claim_Fake_Covid_data = pd.DataFrame (heavy_structure_1, columns = ['tweet_id', 'tweet_text', 'retweet_count', 'repl
#Claim_Fake_Covid_data.to_csv('./05-01-2020/Claim_Fake_Covid_data.csv', index=False)
Claim_Fake_Covid_data["Real_Fake"]=0
Claim_Fake_Covid_data
```

Out[50]:

	tweet_id	tweet_text	retweet_count	reply_count	like_count	Real_Fake
0	1252630938770649089	Can you tell fact from fiction?\n1. The immedi...	0	0	0	0
1	1243968198111789058	2. Only older adults and young people are at r...	0	0	0	0
2	1242474839966765056	2. Only older adults and young people are at r...	0	1	0	0
3	1242461115616866304	#Corona Myth & Reality.\n\nMyth: Only olde...	0	0	0	0
4	1238131052582928385	COVID-19 is just like the flu\n\nOnly older ad...	0	0	0	0
...	...	...	...	...	...	...
371	1243151242194489345	Myth 24: "The virus originated in a laboratory...	0	0	0	0
372	1240247344131493888	@MuraliPriya13 @CarmineSabia @realDonaldTrump ...	0	1	4	0
373	1236627989036781569	16. The virus will die off when temperatures r...	0	1	1	0
374	1245392712607694860	@ActivistaNG @ActionAidNG @Okwuosamj Myth 23:\...	2	0	3	0
375	1236627989036781569	16. The virus will die off when temperatures r...	0	1	1	0

376 rows × 6 columns

Step 3: News Fake Dataset

In [52]:
```
News_Fake_Covid_data = pd.DataFrame (heavy_structure_3, columns = ['tweet_id', 'tweet_text', 'retweet_count', 'reply
#News_Fake_Covid_data.to_csv('./05-01-2020/News_Fake_Covid_data.csv', index=False)
News_Fake_Covid_data["Real_Fake"]=0
News_Fake_Covid_data
```

Out[52]:

	tweet_id	tweet_text	retweet_count	reply_count	like_count	Real_Fake
0	1255263076087185413	Accidents happen.\n \nPentagon Confirms Corona...	0	0	0	0
1	1254390256461365248	Pentagon Confirms Coronavirus Accidently Got I...	0	0	1	0
2	1251308087467802624	@MzMugzzi Pentagon Confirms Coronavirus Accide...	0	0	0	0
3	1251163393790099457	Pentagon Confirms Coronavirus Accidently Got I...	0	0	0	0
4	1250166483570905088	🎵🎺 🥁🎤Blue Skies....smiling at me 😎 \n ...	0	0	0	0
...	...	...	...	...	...	...
5744	1223637588487147521	Coronavirus Contains "HIV Insertions", Stoking...	0	0	0	0
5745	1223637495998550017	Coronavirus Contains "HIV Insertions", Stoking...	0	0	0	0
5746	1223637455783460865	Coronavirus Contains "HIV Insertions", Stoking...	0	0	0	0
5747	1223636583191986177	#Coronavirus Contains 'HIV Insertions', Stokin...	0	0	0	0
5748	1223635628073635843	Coronavirus Contains "HIV Insertions", Stoking...	0	0	0	0

5749 rows × 6 columns

Step 4: News Real Dataset

In [54]:
```
News_Real_Covid_data = pd.DataFrame (heavy_structure_4, columns = ['tweet_id', 'tweet_text', 'retweet_count', 'reply
#News_Fake_Covid_data.to_csv('./05-01-2020/News_Real_Covid_data.csv', index=False)
News_Real_Covid_data["Real_Fake"]=1
News_Real_Covid_data
```

Out[54]:

	tweet_id	tweet_text	retweet_count	reply_count	like_count	Real_Fake
0	1256336927483023360	Coronavirus Outbreak LIVE Updates: CRPF worst-...	0	0	0	1
1	1256328275564982272	Coronavirus live updates: Lockdown extended fo...	0	0	0	1
2	1256325469474963461	Coronavirus China Italy | Coronavirus Outbreak...	0	1	0	1
3	1256322568367116290	Stay Home. Stay Safe. Stay Strong. WATCH LIVE ...	2	0	1	1
4	1256278079745351680	7 PM Bulletin-Coronavirus Outbreak LIVE | Indi...	0	0	1	1
...	...	...	...	...	...	...
835	1249684789562662912	I like numbers 7 & 9! Foods with a Long Sh...	0	0	2	1
836	1250525082499215360	@sciencechick1 Yep. \n\n*Early research indica...	40	4	64	1
837	1250086684478586885	'False Negatives' in COVID-19 Testing: If You ...	0	0	0	1
838	1249945393296629760	'False Negatives' in COVID-19 Testing: If You ...	0	0	0	1
839	1255750122711011334	How To Build Daily and Weekly Routines As Shel...	0	0	0	1

840 rows × 6 columns

Initial analysis – Descriptive Statistics

We performed some exploratory data analysis to look into some descriptive statistics, the correlation between features, assessing data distribution, identifying data anomalies and plotting charts to visualize the data further.

Shape (dimensions) of the DataFrame

```
In [68]: df.shape
Out[68]: (12461, 7)
```

We can see that the dataset has 12,461 observations and 7 features, and one of those features is the target variable.

Data types of the various columns

```
In [69]: df.dtypes
Out[69]: tweet_id          int64
         tweet_text       object
         retweet_count     int64
         reply_count       int64
         like_count        int64
         Real_Fake         int64
         Claim_News       object
         dtype: object
```

We observe that our dataset has a combination of categorical (object) and numeric (float and int) features.

Summary statistics of the numerical features

```
In [85]: df.describe().transpose()
```

Out[85]:

	count	mean	std	min	25%	50%	75%	max
tweet_id	12461.0	1.242201e+18	9.264734e+15	1.207409e+18	1.237568e+18	1.243589e+18	1.249392e+18	1.256366e+18
retweet_count	12461.0	5.860525e+00	1.026739e+02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	7.069000e+03
reply_count	12461.0	8.265789e-01	7.766590e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	4.240000e+02
like_count	12461.0	8.629163e+00	1.183975e+02	0.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00	7.955000e+03
Real_Fake	12461.0	5.084664e-01	4.999484e-01	0.000000e+00	0.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00

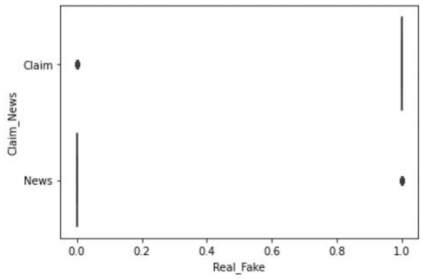
```
In [84]: df.describe(include='object').transpose()
```

Out[84]:

	count	unique	top	freq
tweet_text	12461	11728	1 Deadly Coronavirus What is a coronavirus?\n\...	36
Claim_News	12461	2	News	6589

Segment the target variable by categorical features

In [95]:
```
for column in df.select_dtypes(include="object"):
    if df[column].nunique() <10:
        sns.boxplot(y=column, x="Real_Fake", data=df)
    plt.show()
```



Group numeric features by each categorical feature

In [96]:
```
for column in df.select_dtypes(include='object'):
    if df[column].nunique() < 10:
        display(df.groupby(column).mean())
```

Claim_News	tweet_id	retweet_count	reply_count	like_count	Real_Fake
Claim	1.243301e+18	10.868018	1.087704	15.466792	0.935967
News	1.241220e+18	1.397936	0.593869	2.535590	0.127485

Objectives

We're currently working towards the following research questions:

- Research Question 1 (RQ1): How can we identify if a tweet or a thread of Twitter conversations about COVID-19 is true or not? Which classification model will be the best to identify the validity of tweets with the highest accuracy?
- Research Question 2 (RQ2): Which features of a tweet contribute most significantly to its credibility?

Subsequently, we expect this work can make the following contributions:

1. We develop a prediction model that can predict whether a Tweet or a thread of Twitter conversations is true or not
2. We analyze the various aspects of a tweet to identify the best features that contribute to assessing the credibility of a tweet

Methodology

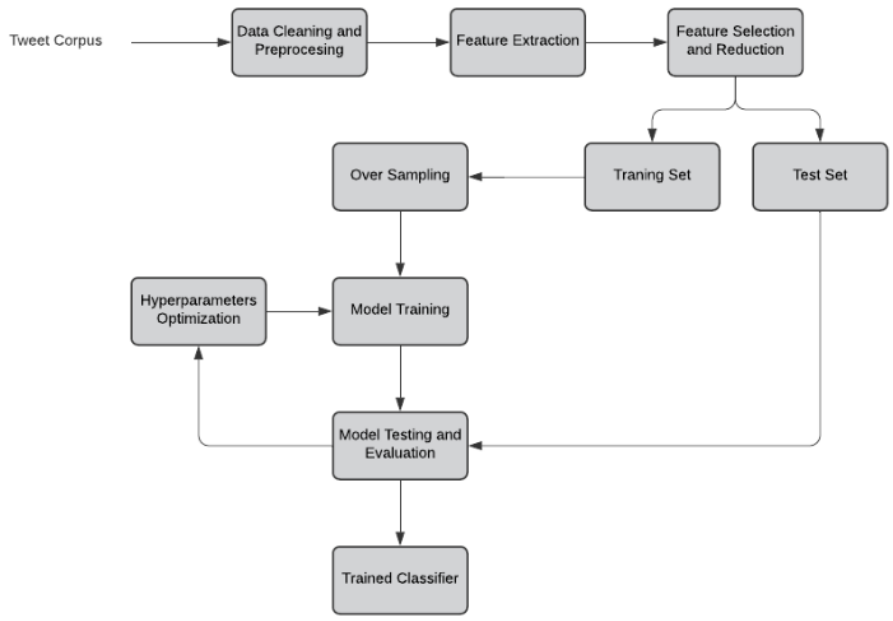


Fig. Algorithm Methodology

Data cleaning and preprocessing:

To eliminate unwanted or irrelevant data or noise in the supplied dataset in order to produce the corpus in a clean and understandable format to improve data accuracy. This step involves the

removal of unwanted symbols such as punctuation, special characters, URLs, hashtags, www, HTTPS, and digits. After the data are cleaned, they are preprocessed, including stop-word removal, stemming, and lemmatization. Here, we only removed the stop words.

Processed_Tweets
tell fact fiction immediate risk becoming seri...
older young people risk like infect people age...
older young people risk like infect people age...
corona myth reality myth older young people ri...
like flu older young people risk infect people...
...
like long shelf life buy avoid via
yep early research common test may produce fal...
false testing assume disease via
false testing assume disease
build daily weekly

Fig. Cleaned Tweet Text

Exploratory Data Analysis:

After performing data cleaning and preprocessing, we performed exploratory data analysis to understand the data further and analyze the available features. We tried to compare features of the fake tweets vs real tweets to get an idea of what features can we optically distinguish on, and further what differences can we tease out by understanding the underlying themes of the data.

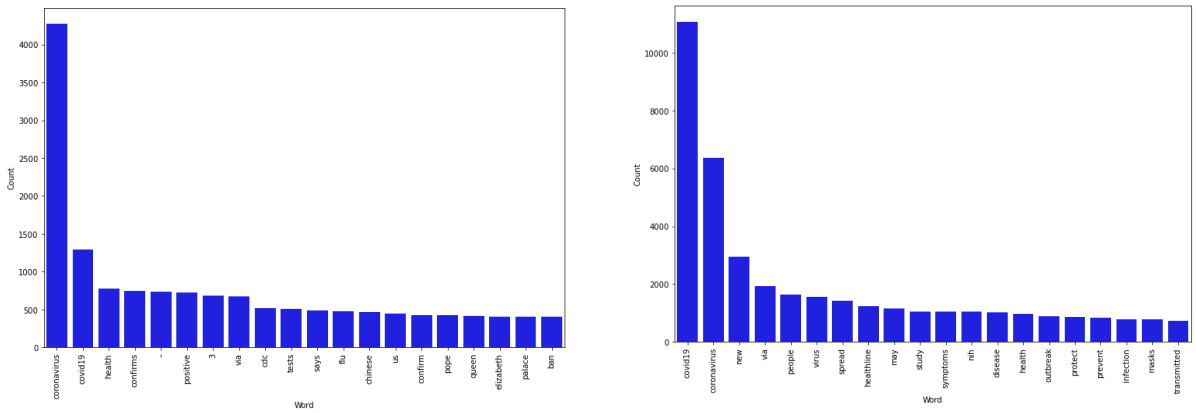


Fig. Fake Tweet v/s Real Tweet Word Count

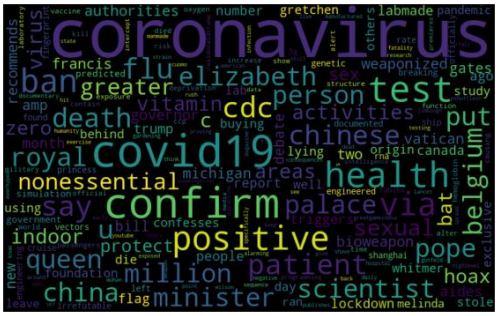


Fig. Fake Tweet Word Cloud

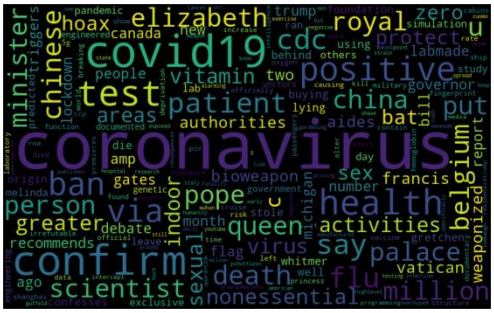


Fig. Real Tweet Word Cloud

Classification Model:

After exploring the features, the entire corpus is divided into training and test sets. We then explored 3 classification models to see which can most accurately predict fake tweets.

Logistic Regression

- This article discusses the basics of Logistic Regression and its implementation in Python.
- Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for a given set of features(or inputs), X.
- Contrary to popular belief, logistic regression IS a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.

  $g(z) = \frac{1}{1+e^{-z}}$

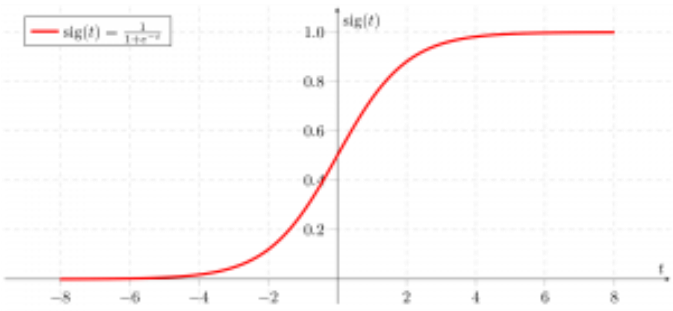


Fig. Sigmoid Function

Decision Tree

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

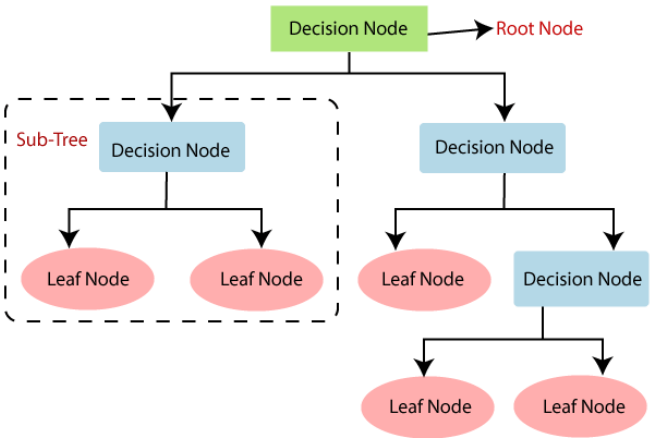


Fig. Decision Tree

Random Forest

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest Algorithm

We can understand the working of Random Forest algorithm with the help of following steps −

- Step 1 − First, start with the selection of random samples from a given dataset.
- Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- Step 3 − In this step, voting will be performed for every predicted result.
- Step 4 − At last, select the most voted prediction result as the final prediction result.

The following diagram will illustrate its working –

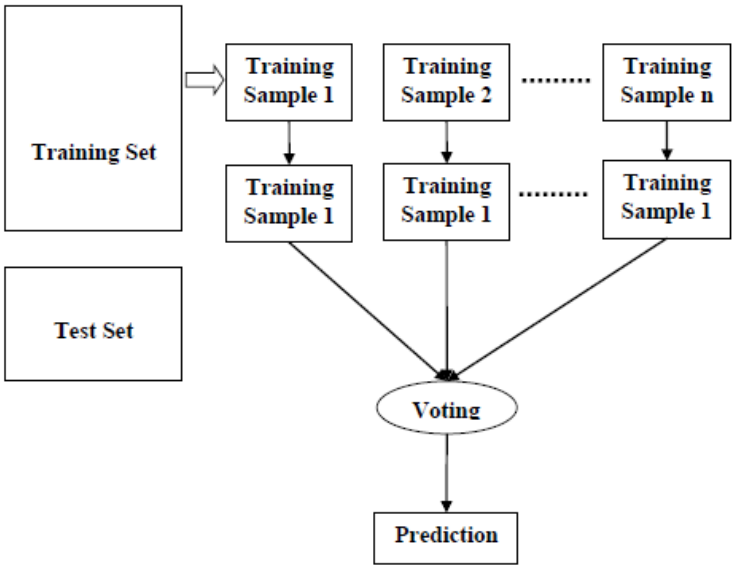


Fig. Random Forest

Classification Model Results:

The Results from running the 3 classification models are as follows:

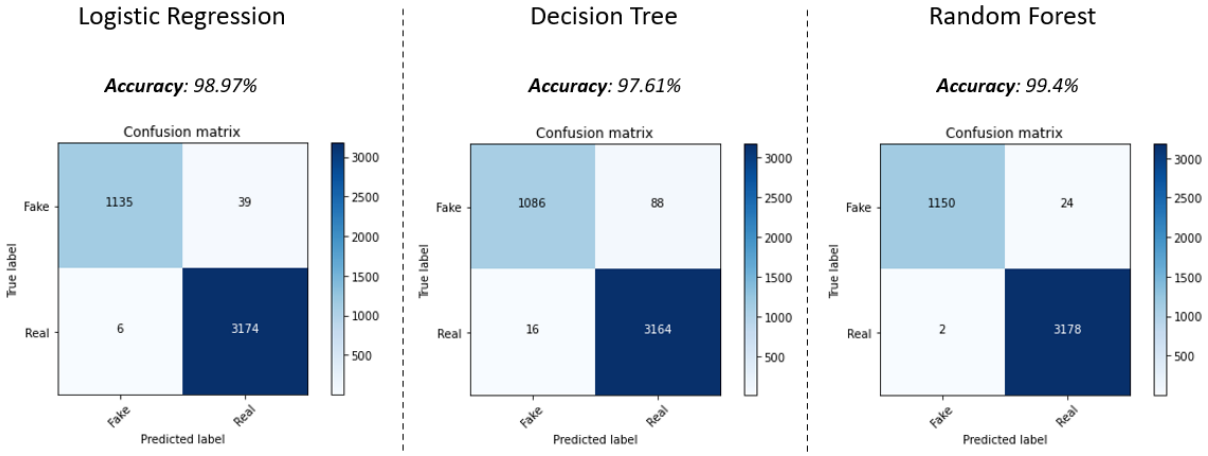


Fig. Classification Results

As we can observe, we get the highest accuracy from the Random Forest Algorithm. What is extremely striking is the fact that it has extremely few misclassified real tweets (i.e., 2) and subsequently also comparatively few misclassified fake tweets, i.e., only 24 compared to 39 and 88 in LR and DT respectively.

We believe Random forest is better than the other two because:

- It emphasizes feature selection — weighs certain features as more important than others.
- It does not assume that the model has a linear relationship — like regression models do.
- It utilizes ensemble learning. If we were to use just 1 decision tree, we wouldn't be using ensemble learning. A random forest takes random samples, forms many decision trees, and then averages out the leaf nodes to get a clearer model.

Analysis of Results

- **What part of the methodology worked?**
    - We were able to develop a highly accurate classification model that helps identify fake tweets with over ~97% accuracy across all classification algorithms
- **Why did the methodology work?**
    - The methodology worked due to the work we put into extracting and preprocessing data, understanding the features, and appropriate implementing the classification models
- **How to improve?**
    - While the current model only leverages the text's features (bag of words, tfidf, tokens etc.) within its pipeline, we can incorporate more features of a tweet such as the person tweeting, replies, comments, likes etc. to make it more extensive and efficient
- **How to utilize your results? What business insights can be derived from your analysis?**
    - The results help us realize that tweet content of a text itself is pretty revealing about the fakeness/realness of the tweet, however, in case the tweets' content gets more similar to real tweets, we might need to incorporate more features to account for a tweet in its entirety

Conclusion & Future Work

While we were able to develop a very accurate classification model, just basing it on the text's features might not be sufficient for the future if fake tweet's text starts to resemble that of a real tweet. In that scenario, we might need to incorporate more user features such as tweet poster, number of followers, likes, comments, shares, date of tweet, date of user joining twitter etc. Once all these features can be retrieved easily, and subsequently packaged into a single predicition model, we anticipate being able to improve the model to account for a lot more features beyond just the text.

.

References

Akpan, N. (2020). The very real consequences of fake news stories and why. *PBS NewsHour*.

Hershy, A. (n.d.). *Towards Data Science*. Retrieved from Is Random Forest better than Logistic Regression? (a comparison): https://towardsdatascience.com/is-random-forest-better-than-logistic-regression-a-comparison-7a0f068963e4

Jackie Ayoub, X. J. (2021). Combat COVID-19 infodemic using explainable natural language processing models. *www.ncbi.nlm.nih.gov*.

Limeng Cui, D. L. (n.d.). *CoAID: COVID-19 Healthcare Misinformation Dataset*. Retrieved from https://arxiv.org/abs/2006.00885

Liz Hamel, L. L. (2021). KFF COVID-19 Vaccine Monitor: Media and Misinformation. *kff.org*.

Mohammed N. Alenezi, Z. M. (2021). Machine Learning in Detecting COVID-19 Misinformation. *www.mdpi.com*.

Parth Patwa, S. S. (2021). Fighting an Infodemic: COVID-19 Fake News Dataset. *https://arxiv.org/pdf/2011.03327.pdf*.

Paul, K. (2018). False news stories are 70% more likely to be retweeted on Twitter than true ones. *marketwatch.com*.