Misinformation presents a huge challenge in online society. As a result, there have been many attempts to identify and classify misinformation. Specifically, in social networking sites, blogs, as well as online newspapers.

Fake news refers to information content that is false, misleading or whose source cannot be verified. This content may be generated to intentionally damage reputations, deceive, or to gain attention.

The model to be build will help to predict the fake news using NLP techniques and machine learning algorithms such as Naive Bayes, Random Forest and logistic regression.

The dataset to be used consists of the following features:

1. id: the index of the row
2. title: title of the news
3. author: the author of the article
4. text: The article written
5. label: Whether it is fake or not (0 for not fake, 1 for fake)

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split


import warnings
warnings.filterwarnings("ignore")

from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler
from imblearn.under_sampling import TomekLinks
from collections import Counter
from imblearn.over_sampling import SMOTE

from sklearn.model_selection import GridSearchCV, RandomizedSearchCV

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.metrics import accuracy_score,recall_score,precision_score,f1_score,confusio
```

Reading Data

In [2]:

```
df = pd.read_csv("/Users/HP/Desktop/train.csv")
df
```

Out[2]:

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1

	id	title	author	text	label
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...	...	...
20795	20795	Rapper T.I.: Trump a ‘Poster Child For White S...	Jerome Hudson	Rapper T. I. unloaded on black celebrities who...	0
20796	20796	N.F.L. Playoffs: Schedule, Matchups and Odds -...	Benjamin Hoffman	When the Green Bay Packers lost to the Washing...	0
20797	20797	Macy’s Is Said to Receive Takeover Approach by...	Michael J. de la Merced and Rachel Abrams	The Macy’s of today grew from the union of sev...	0
20798	20798	NATO, Russia To Hold Parallel Exercises In Bal...	Alex Ansary	NATO, Russia To Hold Parallel Exercises In Bal...	1
20799	20799	What Keeps the F-35 Alive	David Swanson	David Swanson is an author, activist, journa...	1

20800 rows × 5 columns

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20800 entries, 0 to 20799
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   id      20800 non-null  int64 
 1   title   20242 non-null  object
 2   author  18843 non-null  object
 3   text    20761 non-null  object
 4   label   20800 non-null  int64 
dtypes: int64(2), object(3)
memory usage: 812.6+ KB
```

Checking for nulls and duplicates

In [4]:

```
df.isnull().sum()
```

Out[4]:

```
id           0
title      558
author    1957
text        39
label        0
dtype: int64
```

- All of the featues that have missing values are categorical features in which we can't replace it with the mean or median
- Our datset is too huge we have around 20800 rows , we can use the dropping of these nulls values to reduce it a bit to give us advantage of reducing the time of trainng the model

In [5]:

```
df = df.dropna(axis = 0)
df
```

Out[5]:

	id	title	author	text	label

	id	title	author	text	label
0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1
4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...	...	...
20795	20795	Rapper T.I.: Trump a ’Poster Child For White S...	Jerome Hudson	Rapper T. I. unloaded on black celebrities who...	0
20796	20796	N.F.L. Playoffs: Schedule, Matchups and Odds -...	Benjamin Hoffman	When the Green Bay Packers lost to the Washing...	0
20797	20797	Macy’s Is Said to Receive Takeover Approach by...	Michael J. de la Merced and Rachel Abrams	The Macy’s of today grew from the union of sev...	0
20798	20798	NATO, Russia To Hold Parallel Exercises In Bal...	Alex Ansary	NATO, Russia To Hold Parallel Exercises In Bal...	1
20799	20799	What Keeps the F-35 Alive	David Swanson	David Swanson is an author, activist, journa...	1

18285 rows × 5 columns

- Decided to drop all the nulls values as we have a large dataset

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
0
```

Data Cleaning

In [8]:

```
df.reset_index(inplace = True)
df
```

Out[8]:

	index	id	title	author	text	label
0	0	0	House Dem Aide: We Didn’t Even See Comey’s Let...	Darrell Lucus	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	1	1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Daniel J. Flynn	Ever get the feeling your life circles the rou...	0
2	2	2	Why the Truth Might Get You Fired	Consortiumnews.com	Why the Truth Might Get You Fired October 29, ...	1
3	3	3	15 Civilians Killed In Single US Airstrike Hav...	Jessica Purkiss	Videos 15 Civilians Killed In Single US Airstr...	1

	index	id	title	author	text	label
4	4	4	Iranian woman jailed for fictional unpublished...	Howard Portnoy	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...	...	...	...
18280	20795	20795	Rapper T.I.: Trump a 'Poster Child For White S...	Jerome Hudson	Rapper T. I. unloaded on black celebrities who...	0
18281	20796	20796	N.F.L. Playoffs: Schedule, Matchups and Odds -...	Benjamin Hoffman	When the Green Bay Packers lost to the Washing...	0
18282	20797	20797	Macy's Is Said to Receive Takeover Approach by...	Michael J. de la Merced and Rachel Abrams	The Macy's of today grew from the union of sev...	0
18283	20798	20798	NATO, Russia To Hold Parallel Exercises In Bal...	Alex Ansary	NATO, Russia To Hold Parallel Exercises In Bal...	1
18284	20799	20799	What Keeps the F-35 Alive	David Swanson	David Swanson is an author, activist, journa...	1

18285 rows × 6 columns

- We just resetted the index to make the dataset look better but it doesn't affect the performance of the model at all

In [9]:

```
df.drop(['index', 'id','author'], axis = 1, inplace = True)
df
```

Out[9]:

	title	text	label
0	House Dem Aide: We Didn't Even See Comey's Let...	House Dem Aide: We Didn't Even See Comey's Let...	1
1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Ever get the feeling your life circles the rou...	0
2	Why the Truth Might Get You Fired	Why the Truth Might Get You Fired October 29, ...	1
3	15 Civilians Killed In Single US Airstrike Hav...	Videos 15 Civilians Killed In Single US Airstr...	1
4	Iranian woman jailed for fictional unpublished...	Print \nAn Iranian woman has been sentenced to...	1
...	...	...	...
18280	Rapper T.I.: Trump a 'Poster Child For White S...	Rapper T. I. unloaded on black celebrities who...	0
18281	N.F.L. Playoffs: Schedule, Matchups and Odds -...	When the Green Bay Packers lost to the Washing...	0
18282	Macy's Is Said to Receive Takeover Approach by...	The Macy's of today grew from the union of sev...	0
18283	NATO, Russia To Hold Parallel Exercises In Bal...	NATO, Russia To Hold Parallel Exercises In Bal...	1
18284	What Keeps the F-35 Alive	David Swanson is an author, activist, journa...	1

18285 rows × 3 columns

- dropped out the columns of index, id and author as there are no use of them
- no effect of them on our model

In [10]:

```
df['title_text'] = df['title'] + ' ' + df['text']
df
```

Out[10]:

	title	text	label	title_text
0	House Dem Aide: We Didn’t Even See Comey’s Let...	House Dem Aide: We Didn’t Even See Comey’s Let...	1	House Dem Aide: We Didn’t Even See Comey’s Let...
1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	Ever get the feeling your life circles the rou...	0	FLYNN: Hillary Clinton, Big Woman on Campus - ...
2	Why the Truth Might Get You Fired	Why the Truth Might Get You Fired October 29, ...	1	Why the Truth Might Get You Fired Why the Trut...
3	15 Civilians Killed In Single US Airstrike Hav...	Videos 15 Civilians Killed In Single US Airstr...	1	15 Civilians Killed In Single US Airstrike Hav...
4	Iranian woman jailed for fictional unpublished...	Print \nAn Iranian woman has been sentenced to...	1	Iranian woman jailed for fictional unpublished...
...	...	...	...	...
18280	Rapper T.I.: Trump a ’Poster Child For White S...	Rapper T. I. unloaded on black celebrities who...	0	Rapper T.I.: Trump a ’Poster Child For White S...
18281	N.F.L. Playoffs: Schedule, Matchups and Odds -...	When the Green Bay Packers lost to the Washing...	0	N.F.L. Playoffs: Schedule, Matchups and Odds -...
18282	Macy’s Is Said to Receive Takeover Approach by...	The Macy’s of today grew from the union of sev...	0	Macy’s Is Said to Receive Takeover Approach by...
18283	NATO, Russia To Hold Parallel Exercises In Bal...	NATO, Russia To Hold Parallel Exercises In Bal...	1	NATO, Russia To Hold Parallel Exercises In Bal...
18284	What Keeps the F-35 Alive	David Swanson is an author, activist, journa...	1	What Keeps the F-35 Alive David Swanson is a...

18285 rows × 4 columns

- Combined both the title and the text values in a single column so that we can train them easily and the datset look much better

In [11]:

```
title_text = df['title_text']

df.drop(['title','text','title_text'],axis = 1, inplace =True)

df.insert(0 , 'title_text', title_text)

df
```

Out[11]:

	title_text	label
0	House Dem Aide: We Didn’t Even See Comey’s Let...	1
1	FLYNN: Hillary Clinton, Big Woman on Campus - ...	0
2	Why the Truth Might Get You Fired Why the Trut...	1
3	15 Civilians Killed In Single US Airstrike Hav...	1
4	Iranian woman jailed for fictional unpublished...	1
...	...	...
18280	Rapper T.I.: Trump a ’Poster Child For White S...	0
18281	N.F.L. Playoffs: Schedule, Matchups and Odds -...	0
18282	Macy’s Is Said to Receive Takeover Approach by...	0

	title_text	label
18283	NATO, Russia To Hold Parallel Exercises In Bal...	1
18284	What Keeps the F-35 Alive David Swanson is a...	1

18285 rows × 2 columns

- Just adjusted the position of the column of (title_text)
- dropped out the columns of title and text as now there is no use of them

Data preprocessing

In [12]:

```
import re
import nltk
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [13]:

```
print(stopwords.words('english'))
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've",
"you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himsel
f', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'the
m', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "tha
t'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have',
'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'i
f', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'ag
ainst', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to',
'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'the
n', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each',
'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same',
'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "s
hould've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn',
"couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'hav
en', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn',
"needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't",
'won', "won't", 'wouldn', "wouldn't"]
```

- these are the english stopwords that needs to be removed from the text as there is no use of them

In [14]:

```
ps = PorterStemmer()

def stemming(text):
    stemmed_text = re.sub('[^a-zA-Z]',' ',text)
    stemmed_text = stemmed_text.lower()
    stemmed_text = stemmed_text.split()
    stemmed_text = [ps.stem(word) for word in stemmed_text if not word in stopwords.words
    stemmed_text = ' '.join(stemmed_text)
    return stemmed_text
```

- Created this function to first, remove any text that isn't of alphabets A to Z
- Secondly, lower all the words left
- Thirdly, splitting of the words
- Applying of the Stemming which is changing the word to it's root word

- finally, joining of all the words to return the text back

In [15]:

```
df['title_text'] = df['title_text'].apply(stemming)
```

In [16]:

```
df['title_text']
```

Out[16]:

```
0        hous dem aid even see comey letter jason chaff...
1        flynn hillari clinton big woman campu breitbar...
2        truth might get fire truth might get fire octo...
3        civilian kill singl us airstrik identifi video...
4        iranian woman jail fiction unpublish stori wom...
                               ...
18280    rapper trump poster child white supremaci rapp...
18281    n f l playoff schedul matchup odd new york tim...
18282    maci said receiv takeov approach hudson bay ne...
18283    nato russia hold parallel exercis balkan nato ...
18284    keep f aliv david swanson author activist jour...
Name: title_text, Length: 18285, dtype: object
```

Splitting of data

In [27]:

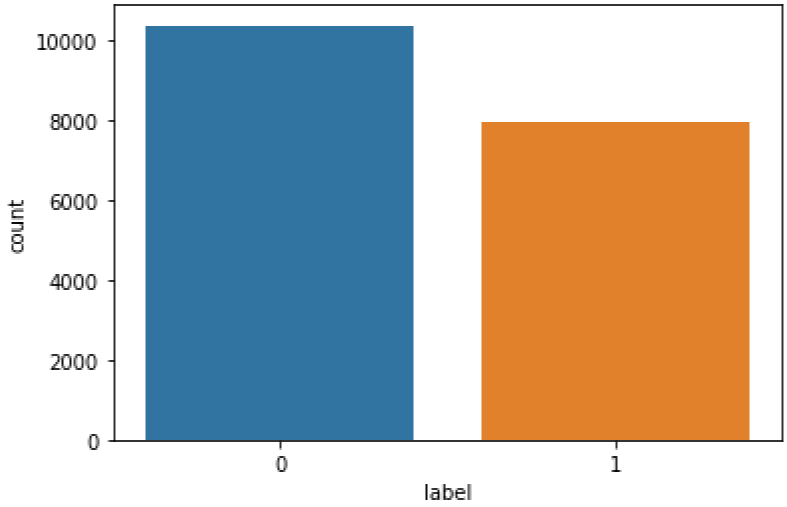
```
x = df['title_text']
y = df['label']
```

Checking of data imbalancement

In [28]:

```
sns.countplot(x = df['label'])
print(df['label'].value_counts())
```

```
0    10361
1     7924
Name: label, dtype: int64
```



- As it is represented, the number of labels that is 1 is approximately 80% of the label 0
- this 20 % imbalancement of data can affect our training of the model negatively
- OverSampling or UnderSampling is required
- Decided to choose UnderSampling as the dataset is huge so we can use the advantage of reducing the training time.

Handling imbalanced data

In [29]:

```
Counter(y)
```

Out[29]:

```
Counter({1: 7924, 0: 10361})
```

In [30]:

```
ruc = RandomUnderSampler(random_state = 42)
x,y = ruc.fit_resample(x.values.reshape(-1,1),y)
```

In [31]:

```
Counter(y)
```

Out[31]:

```
Counter({0: 7924, 1: 7924})
```

train test split

In [32]:

```
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2, random_state =42

x_train,x_val,y_train,y_val = train_test_split(x_train,y_train,test_size = 0.2, random_sta
```

TfidfVectorizer

In [33]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorization = TfidfVectorizer()

x_train = vectorization.fit_transform(x_train.ravel())
x_val = vectorization.transform(x_val.ravel())
x_test = vectorization.transform(x_test.ravel())
```

- Here we converted the textual data to numerical data in which the model can understand
- We just applied a fit_transform function for the x_train
- Applied a transform function only for the (x_val,x_test)

Naive Bayes

In [35]:

```
from sklearn.naive_bayes import MultinomialNB

## check overfitting

nb = MultinomialNB()
nb.fit(x_train,y_train)

x_train_pred = nb.predict(x_train)
train_score = accuracy_score(y_train,x_train_pred)
print(f'the train score is ={train_score}')

x_val_pred = nb.predict(x_val)
val_score = accuracy_score(y_val,x_val_pred)
print(f'the valid score is ={val_score}')
```

```
# scores are approximately near so no overfitting
```

```
the train score is =0.931966081640702
the valid score is =0.9002365930599369
```

In [37]:

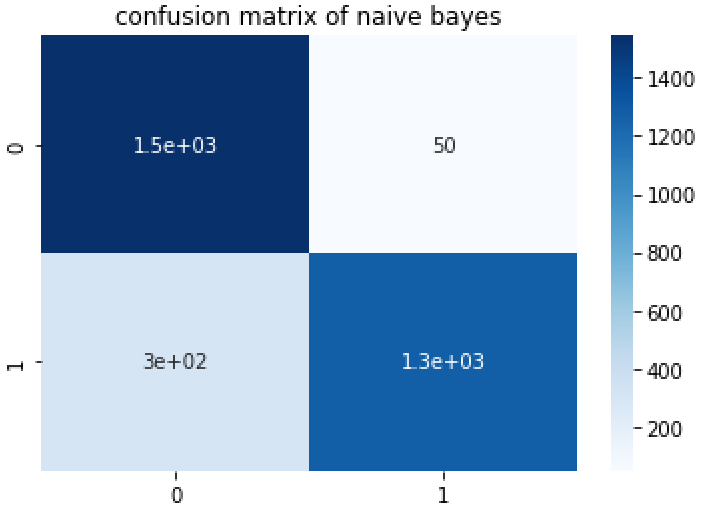
```
nb = MultinomialNB()

model_naive = nb.fit( x_train, y_train)
y_pred = nb.predict(x_test)
```

In [38]:

```
cnf_mat = confusion_matrix(y_test,y_pred)
sns.heatmap( cnf_mat , annot = True , cmap = 'Blues')
plt.title('confusion matrix of naive bayes')
```

Out[38]:

```
Text(0.5, 1.0, 'confusion matrix of naive bayes')
```



In [39]:

```
accuracy_naive = accuracy_score(y_test, y_pred)
print(f'the accuracy of the naive bayes model is = {accuracy_naive * 100} %')
recall = recall_score(y_test, y_pred)
print(f'the recall of the naive bayes model is = {recall * 100} %')
precision = precision_score(y_test, y_pred)
print(f'the precision of the naive bayes model is = {precision * 100} %')
f1 = f1_score(y_test, y_pred)
print(f'the f1_score of the naive bayes model is = {f1 * 100} %')
```

```
the accuracy of the naive bayes model is = 88.83280757097792 %
the recall of the naive bayes model is = 80.71065989847716 %
the precision of the naive bayes model is = 96.2178517397882 %
the f1_score of the naive bayes model is = 87.78467908902692 %
```

In [40]:

```
report = classification_report(y_test, y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.84      0.97      0.90      1594
           1       0.96      0.81      0.88      1576

    accuracy                           0.89      3170
   macro avg       0.90      0.89      0.89      3170
weighted avg       0.90      0.89      0.89      3170
```

Random Forest

In [49]:

```
# Use grid search to find best value
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators = 500)

params_grid = {
    'max_depth':[3,4,5,6,7,8,9,10],
    'criterion': ['gini', 'entropy', 'log_loss']
}

grid = GridSearchCV(
    rf,
    params_grid,
    cv = 5
)
grid.fit(x_train,y_train)

print(f'the best value is = {grid.best_params_}')
```

```
the best value is = {'criterion': 'log_loss', 'max_depth': 10}
```

In [51]:

```
## check overfitting

rf = RandomForestClassifier(n_estimators = 500, bootstrap = True,n_jobs = -1, max_depth =
rf.fit(x_train,y_train)

x_train_pred = rf.predict(x_train)
train_score = accuracy_score(y_train,x_train_pred)
print(f'the train score is ={train_score}')

x_val_pred = rf.predict(x_val)
val_score = accuracy_score(y_val,x_val_pred)
print(f'the valid score is ={val_score}')

# scores are approximately near so no overfitting
```

```
the train score is =0.9245710905146913
the valid score is =0.876577287066246
```

In [52]:

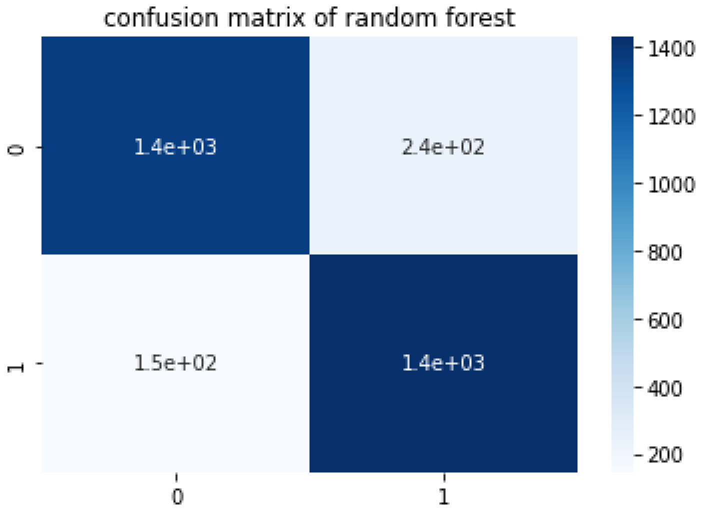
```
rf = RandomForestClassifier(n_estimators = 500, bootstrap = True,n_jobs = -1,max_depth = 1

rf.fit(x_train,y_train)
y_pred = rf.predict(x_test)
```

In [53]:

```
cnf_mat = confusion_matrix(y_test,y_pred)
sns.heatmap( cnf_mat , annot = True , cmap = 'Blues')
plt.title('confusion matrix of random forest')
```

Out[53]:

```
Text(0.5, 1.0, 'confusion matrix of random forest')
```



In [54]:

```
accuracy_random = accuracy_score(y_test, y_pred)
print(f'the accuracy of the random forest model is = {accuracy_random * 100} %')
recall = recall_score(y_test, y_pred)
print(f'the recall of the random forest model is = {recall * 100} %')
precision = precision_score(y_test, y_pred)
print(f'the precision of the random forest model is = {precision * 100} %')
f1 = f1_score(y_test, y_pred)
print(f'the f1_score of the random forest model is = {f1 * 100} %')
```

```
the accuracy of the random forest model is = 87.98107255520505 %
the recall of the random forest model is = 90.73604060913706 %
the precision of the random forest model is = 85.88588588588588 %
the f1_score of the random forest model is = 88.24436902190683 %
```

In [55]:

```
report = classification_report(y_test, y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.90      0.85      0.88      1594
           1       0.86      0.91      0.88      1576

    accuracy                           0.88      3170
   macro avg       0.88      0.88      0.88      3170
weighted avg       0.88      0.88      0.88      3170
```

Logistic Regression

In [56]:

```
from sklearn.linear_model import LogisticRegression

## check overfitting

lr = LogisticRegression(random_state = 42)
lr.fit(x_train,y_train)

x_train_pred = lr.predict(x_train)
train_score = accuracy_score(y_train,x_train_pred)
print(f'the train score is ={train_score}')

x_val_pred = lr.predict(x_val)
val_score = accuracy_score(y_val,x_val_pred)
print(f'the valid score is ={val_score}')
```

```
# scores are equal so no overfitting
```

```
the train score is =0.9740682311181227
the valid score is =0.9522870662460567
```

In [57]:

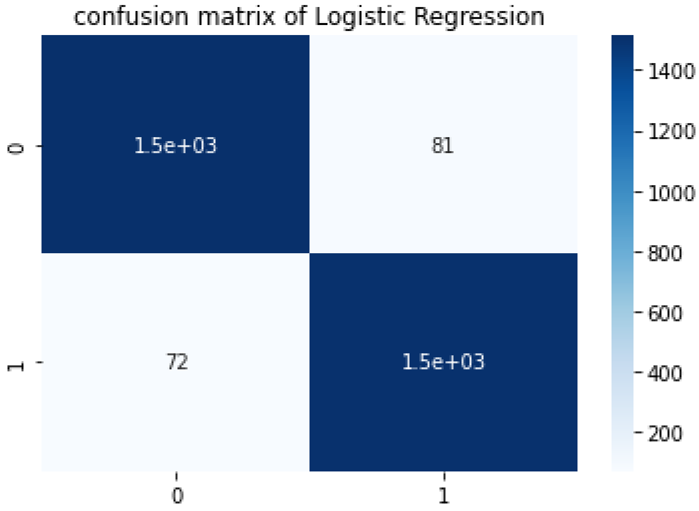
```
lr = LogisticRegression(random_state = 42)

log = lr.fit(x_train,y_train)
y_pred = log.predict(x_test)
```

In [58]:

```
cnf_mat = confusion_matrix(y_test,y_pred)
sns.heatmap( cnf_mat , annot = True , cmap = 'Blues')
plt.title('confusion matrix of Logistic Regression')
```

Out[58]:

```
Text(0.5, 1.0, 'confusion matrix of Logistic Regression')
```



In [59]:

```
accuracy_log = accuracy_score(y_test, y_pred)
print(f'the accuracy of the logistic regression model is = {accuracy_log * 100} %')
recall = recall_score(y_test, y_pred)
print(f'the recall of the logistic regression model is = {recall * 100} %')
precision = precision_score(y_test, y_pred)
print(f'the precision of the logistic regression model is = {precision * 100} %')
f1 = f1_score(y_test, y_pred)
print(f'the f1_score of the logistic regression model is = {f1 * 100} %')
```

```
the accuracy of the logistic regression model is = 95.17350157728707 %
the recall of the logistic regression model is = 95.43147208121827 %
the precision of the logistic regression model is = 94.88958990536278 %
the f1_score of the logistic regression model is = 95.1597595697564 %
```

In [60]:

```
report = classification_report(y_test, y_pred)
print(report)
```

```
              precision    recall  f1-score   support

           0       0.95      0.95      0.95      1594
           1       0.95      0.95      0.95      1576

    accuracy                           0.95      3170
   macro avg       0.95      0.95      0.95      3170
weighted avg       0.95      0.95      0.95      3170
```

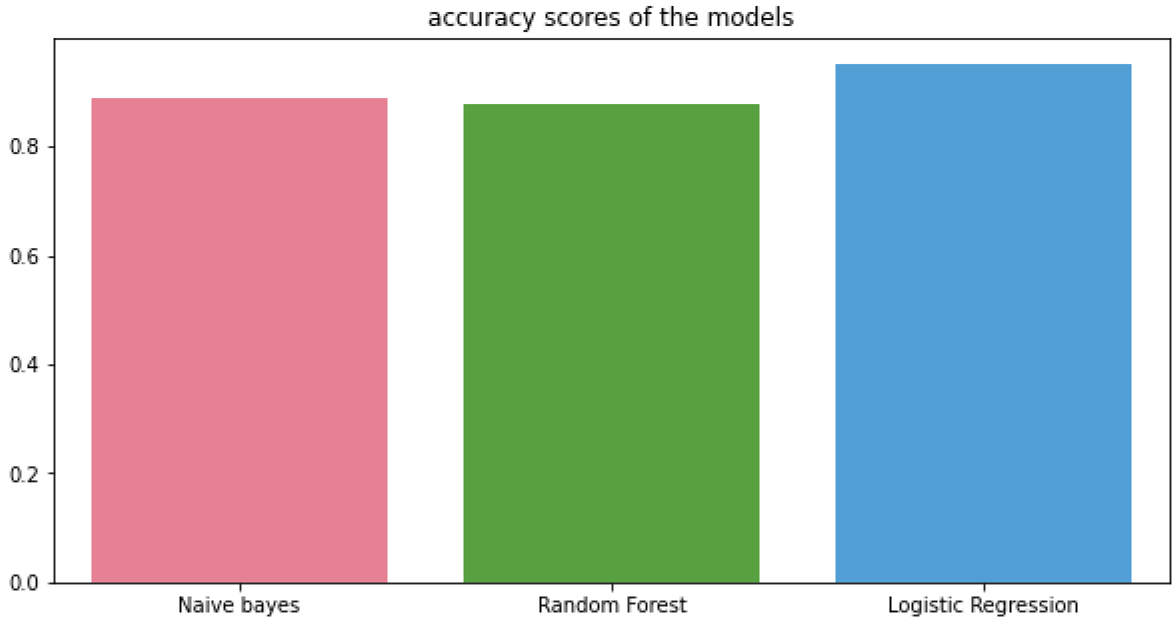
Comparison of the evaluation of the models (Accuracy)

In [61]:

```
models = ['Naive bayes', 'Random Forest', 'Logistic Regression']
scores = [accuracy_naive, accuracy_random, accuracy_log]

plt.figure(figsize = (10,5))
sns.barplot(x = models, y = scores, data =df , palette = 'husl')
plt.title('accuracy scores of the models')
```

Out[61]:

```
Text(0.5, 1.0, 'accuracy scores of the models')
```



- We can conclude from the plot above that Logistic Regression classifier scored the highest accuracy
- Naive bayes classifier was much not much near to the accuracy scored by the Logistic Regression classifier than the accuracy socred by random forest classifier
- Surprisingly, Random forest Classifier had the lowest score among all of the classifiers

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: