In [2]:
```
import pandas as pd
df=pd.read_csv("D:\mokshith\data_banknote_authentication (1).csv")
df
```

Out[2]:

	variance	skewness	curtosis	entropy	class
0	3.62160	8.66610	-2.8073	-0.44699	0
1	4.54590	8.16740	-2.4586	-1.46210	0
2	3.86600	-2.63830	1.9242	0.10645	0
3	3.45660	9.52280	-4.0112	-3.59440	0
4	0.32924	-4.45520	4.5718	-0.98880	0
...	...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949	1
1368	-1.38870	-4.87730	6.4774	0.34179	1
1369	-3.75030	-13.45860	17.5932	-2.77710	1
1370	-3.56370	-8.38270	12.3930	-1.28230	1
1371	-2.54190	-0.65804	2.6842	1.19520	1

1372 rows × 5 columns

In [3]:
```
x=df.drop('class',axis=1)
x
```

Out[3]:

	variance	skewness	curtosis	entropy
0	3.62160	8.66610	-2.8073	-0.44699
1	4.54590	8.16740	-2.4586	-1.46210
2	3.86600	-2.63830	1.9242	0.10645
3	3.45660	9.52280	-4.0112	-3.59440
4	0.32924	-4.45520	4.5718	-0.98880
...	...	...	...	...
1367	0.40614	1.34920	-1.4501	-0.55949
1368	-1.38870	-4.87730	6.4774	0.34179
1369	-3.75030	-13.45860	17.5932	-2.77710
1370	-3.56370	-8.38270	12.3930	-1.28230
1371	-2.54190	-0.65804	2.6842	1.19520

1372 rows × 4 columns

In [4]:
```
y=df['class']
y
```

Out[4]:
```
0       0
1       0
2       0
3       0
4       0
       ..
1367    1
1368    1
1369    1
1370    1
1371    1
Name: class, Length: 1372, dtype: int64
```

In [5]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain, ytest=train_test_split(x,y,test_size =0.20,random_state=0)
print("xtrain shape : ", xtrain.shape)
print("xtest shape  : ", xtest.shape)
print("ytrain shape : ", ytrain.shape)
print("ytest shape  : ", ytest.shape)
```

```
xtrain shape :  (1097, 4)
xtest shape  :  (275, 4)
ytrain shape :  (1097,)
ytest shape  :  (275,)
```

In [6]:
```
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(xtrain,ytrain)
```

Out[6]:
```
LogisticRegression
LogisticRegression()
```

In [7]:
```
test_prediction=model.predict(xtest)
test_prediction
```

Out[7]:
```
array([1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1], dtype=int64)
```

In [8]:
```
train_Acuracy=model.score(xtrain,ytrain)
print('Train_Accuracy:',train_Acuracy)
test_Acuracy=model.score(xtest,ytest)
print('Test_Accuracy:',test_Acuracy)
```

```
Train_Accuracy: 0.9899726526891522
Test_Accuracy: 0.9927272727272727
```

In [24]:
```
from sklearn.metrics import classification_report
print(classification_report(ytest,test_prediction))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       157
           1       0.98      1.00      0.99       118

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275
```

In [9]:
```
from sklearn.ensemble import RandomForestClassifier
regressor=RandomForestClassifier()
regressor.fit(xtrain,ytrain)
```

Out[9]:
```
▾ RandomForestClassifier

RandomForestClassifier()
```

In [20]:
```
test_prediction=model.predict(xtest)
test_prediction
```

Out[20]:
```
array([1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1], dtype=int64)
```

In [21]:
```
train_Acuracy=model.score(xtrain,ytrain)
print('Train_Accuracy:',train_Acuracy)
test_Acuracy=model.score(xtest,ytest)
print('Test_Accuracy:',test_Acuracy)
```

```
Train_Accuracy: 0.9899726526891522
Test_Accuracy: 0.9927272727272727
```

In [23]:
```
from sklearn.metrics import classification_report
print(classification_report(ytest,test_prediction))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       157
           1       0.98      1.00      0.99       118

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275
```

In [12]:
```
from sklearn.metrics import accuracy_score
from sklearn.neural_network import MLPClassifier
```

In [13]:
```
ann_model = MLPClassifier()
ann_model.fit(xtrain, ytrain)
```

Out[13]:
```
MLPClassifier

MLPClassifier()
```

In [14]:
```
ann_predictions = ann_model.predict(xtest)
ann_predictions
```

Out[14]:
```
array([1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0,
       0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
       1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,
       1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1,
       0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0,
       1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1], dtype=int64)
```

In [15]:
```
ann_accuracy = accuracy_score(ytest, ann_predictions)
print("ANN Accuracy:", ann_accuracy)
```

```
ANN Accuracy: 1.0
```

In [16]:
```
train_Acuracy=ann_model.score(xtrain,ytrain)
print('Train_Accuracy:',train_Acuracy)
test_Acuracy=ann_model.score(xtest,ytest)
print('Test_Accuracy:',test_Acuracy)
```

```
Train_Accuracy: 1.0
Test_Accuracy: 1.0
```

In [17]:
```
from sklearn.metrics import classification_report
print(classification_report(ytest,ann_predictions))
```

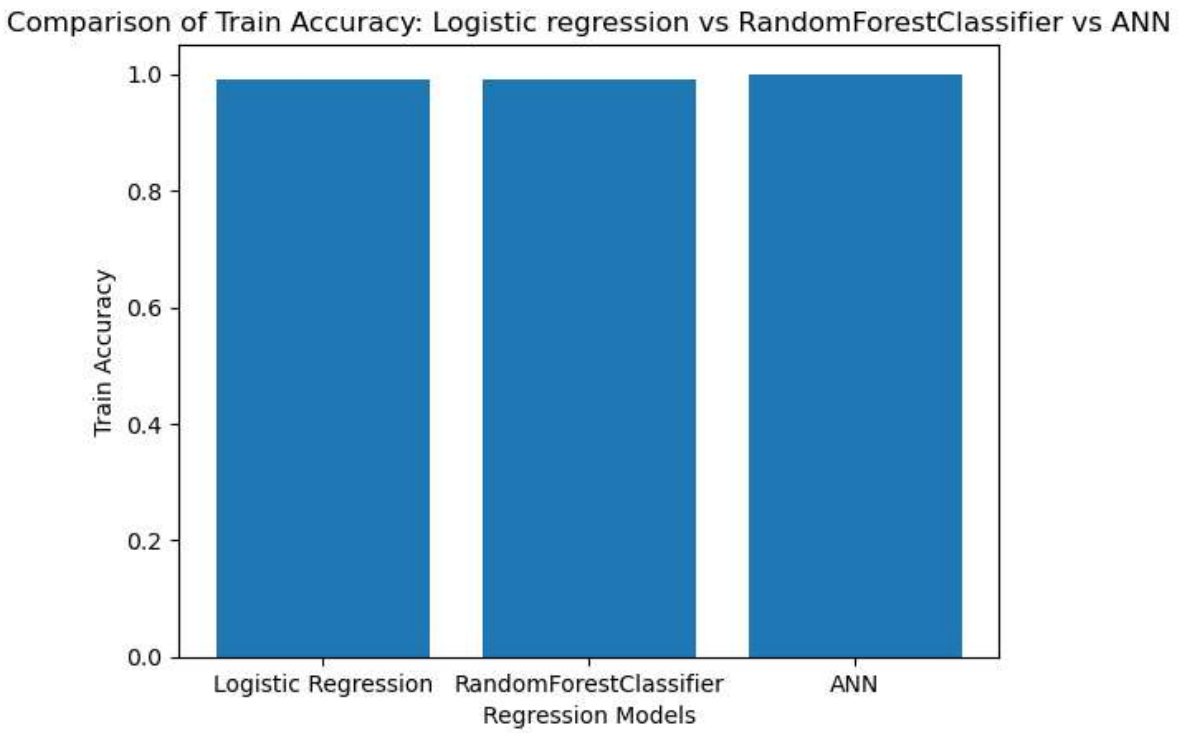
```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       157
           1       1.00      1.00      1.00       118

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275
```

In [18]:
```
import matplotlib.pyplot as plt
x=0.9899726526891522
y= 0.9899726526891522
z=1.0
accuracy_scores = [x,y,z]
model_names = ['Logistic Regression', 'RandomForestClassifier','ANN']
plt.bar(model_names, accuracy_scores)
plt.xlabel('Regression Models')
plt.ylabel('Train Accuracy')
plt.title('Comparison of Train Accuracy: Logistic regression vs RandomForestCla

plt.show()
```



In [ ]: