In [25]:
```
import pandas as pd
```

In [26]:
```
data=pd.read_csv('C:/Users/GPT BANTWAL/Documents/Breast_Cancer_data.csv')
data
```

Out[26]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothne
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	
...	...	...	...	...	...	...	
564	926424	M	21.56	22.39	142.00	1479.0	
565	926682	M	20.13	28.25	131.20	1261.0	
566	926954	M	16.60	28.08	108.30	858.1	
567	927241	M	20.60	29.33	140.10	1265.0	
568	92751	B	7.76	24.54	47.92	181.0	

569 rows × 32 columns

In [27]:
```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
data['diagnosis']=le.fit_transform(data['diagnosis'])
data['diagnosis'].unique()
```

Out[27]:
```
array([1, 0])
```

In [31]:
```
x=data.drop(['diagnosis'],axis=1)
x
```

Out[31]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	842302	17.99	10.38	122.80	1001.0	0.11840
1	842517	20.57	17.77	132.90	1326.0	0.08474
2	84300903	19.69	21.25	130.00	1203.0	0.10960
3	84348301	11.42	20.38	77.58	386.1	0.14250
4	84358402	20.29	14.34	135.10	1297.0	0.10030
...	...	...	...	...	...	...
564	926424	21.56	22.39	142.00	1479.0	0.11100
565	926682	20.13	28.25	131.20	1261.0	0.09780
566	926954	16.60	28.08	108.30	858.1	0.08455
567	927241	20.60	29.33	140.10	1265.0	0.11780
568	92751	7.76	24.54	47.92	181.0	0.05263

569 rows × 31 columns

In [12]:
```
y=data.diagnosis
y
```

Out[12]:
```
0      1
1      1
2      1
3      1
4      1
      ..
564    1
565    1
566    1
567    1
568    0
Name: diagnosis, Length: 569, dtype: int32
```

In [13]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.2,random_state=10
```

In [14]:
```
print("xtrain:",xtrain.shape)
print("xtest:",xtest.shape)
print("ytrain:",ytrain.shape)
print("ytest:",ytest.shape)
```

```
xtrain: (455, 31)
xtest: (114, 31)
ytrain: (455,)
ytest: (114,)
```

In [15]:
```
from sklearn.svm import SVC
model=SVC(kernel='rbf',random_state=1,C=1,gamma='auto')
model.fit(xtrain,ytrain)
```

Out[15]:
```
SVC
SVC(C=1, gamma='auto', random_state=1)
```

In [16]:
```
train_predictions=model.predict(xtrain)
test_predictions=model.predict(xtest)
```

In [17]:
```
train_accuracy=model.score(xtrain,ytrain)
print("Accuracy of the model on train data=",train_accuracy)
test_accuracy=model.score(xtest,ytest)
print("Accuracy of the model on train data=",test_accuracy)
```

```
Accuracy of the model on train data= 1.0
Accuracy of the model on train data= 0.5701754385964912
```

In [18]:
```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(xtrain,ytrain)
```

Out[18]:
```
LogisticRegression
LogisticRegression()
```

In [19]:
```
train_predictions=model.predict(xtrain)
test_predictions=model.predict(xtest)
```

In [20]:
```
print("Accuracy score of the model on training data:",model.score(xtrain,ytr
print("Accuracy score of the model on test data:",model.score(xtest,ytest))
```

```
Accuracy score of the model on training data: 0.6417582417582418
Accuracy score of the model on test data: 0.5701754385964912
```

In [21]:
```
from sklearn.neural_network import MLPClassifier
nn_model= MLPClassifier(hidden_layer_sizes=(50))
nn_model.fit(xtrain,ytrain)
predict_digit=nn_model.predict(xtrain)
print(xtrain.shape)
```

```
(455, 31)
```

In [22]:
```
from sklearn.metrics import classification_report
print(classification_report(ytrain,predict_digit))
```

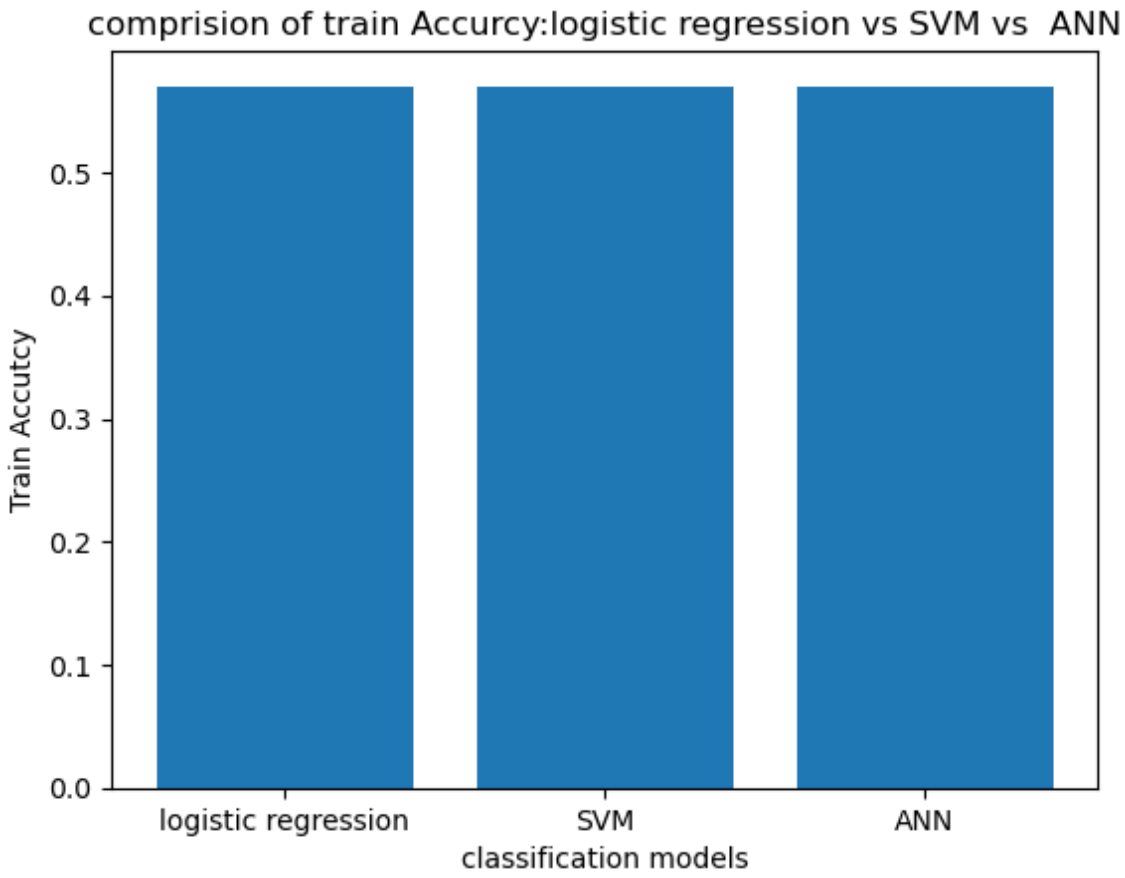
```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       292
           1       0.36      1.00      0.53       163

    accuracy                           0.36       455
   macro avg       0.18      0.50      0.26       455
weighted avg       0.13      0.36      0.19       455


C:\Users\GPT BANTWAL\anaconda3\Lib\site-packages\sklearn\metrics\_classifi
cation.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defi
ned and being set to 0.0 in labels with no predicted samples. Use `zero_di
vision` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\GPT BANTWAL\anaconda3\Lib\site-packages\sklearn\metrics\_classifi
cation.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defi
ned and being set to 0.0 in labels with no predicted samples. Use `zero_di
vision` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\GPT BANTWAL\anaconda3\Lib\site-packages\sklearn\metrics\_classifi
cation.py:1469: UndefinedMetricWarning: Precision and F-score are ill-defi
ned and being set to 0.0 in labels with no predicted samples. Use `zero_di
vision` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [23]:
```
print("Accuracy score of the model on training data:",model.score(xtrain,ytr
print("Accuracy score of the model on test data:",model.score(xtest,ytest))
```

```
Accuracy score of the model on training data: 0.6417582417582418
Accuracy score of the model on test data: 0.5701754385964912
```

In [34]:
```
import matplotlib.pyplot as plt
x=0.5701754385964912
y=0.5701754385964912
z= 0.5701754385964912
accuracy_scores=[x,y,z]
model_names =['logistic regression','SVM','ANN']
plt.bar(model_names,accuracy_scores)
plt.xlabel('classification models')
plt.ylabel('Train Accutcy')
plt.title('comprision of train Accurcy:logistic regression vs SVM vs  ANN')
```

Out[34]:
```
Text(0.5, 1.0, 'comprision of train Accurcy:logistic regression vs SVM vs
ANN')
```



In [ ]: