Devanshu Surana
PC-23, 1032210755
Panel C, Batch C1

ML Lab Assignment 5

FAQ's

1) What is an Ensemble model?

→ It is a model (machine learning) technique that combines multiple individual models to improve overall performance and predictive accuracy.

2) What are bagging, boosting and stacking?

→ • Bagging: Constructs multiple models in parallel using random subsets of the training data and averages their predictions to reduce variance.

• Boosting: Builds models sequentially, with each subsequent model focusing more on instances misclassified by previous models thereby reducing bias.

• Stacking: Combines predictions from multiple model using a meta learner which learns how to best weigh or combine these predictions to make the final prediction.

3) Can we ensemble multiple models of same ML algorithm

→ Yes, you can ensemble multiple models of same ML algorithm, often by training them on different subse-ts of data or with different hyperparameters, and then combining their predictions to improve overall performance.

4) How can we identify the weights of different models?

→ You can identify the weights of different models in an ensemble by either assigning equal weighs or using techniques like cross-validation, grid search, to determine optimal weights based on performance metrics such as accuracy or loss.

5) What are the benefit of ensemble model?

→ Ensemble models offer improved predictive accuracy robustness to overfitting, and the ability to capture complex patterns by combining multiple indivi-dual models.

ml-lab-5

April 17, 2024

[13]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import cross_val_predict, train_test_split
from sklearn.metrics import confusion_matrix, classification_report,
  accuracy_score
from sklearn.ensemble import RandomForestClassifier
```

[14]:
```
breast_cancer = load_breast_cancer()
X = breast_cancer.data
y = breast_cancer.target
```

[15]:
```
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size = 0.2,
  random_state=42)
```

[16]:
```
rf = RandomForestClassifier(n_estimators=100)
```

[17]:
```
rf.fit(X_train, y_train)
```

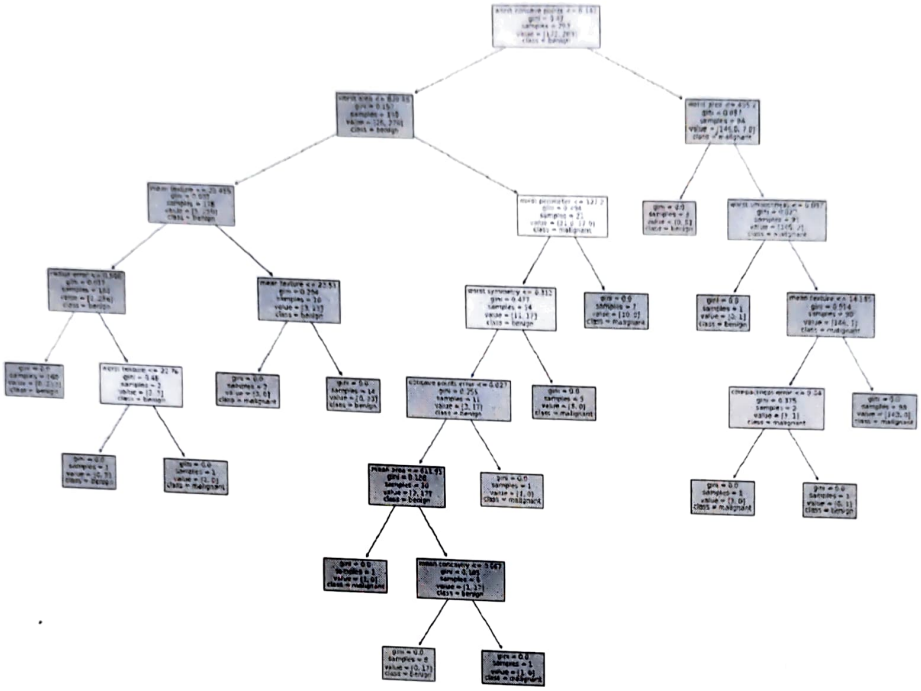
[17]
```
RandomForestClassifier()
```

[18]:
```
y_pred = rf.predict(X_test)
```

[19]:
```
accuracy = accuracy_score(y_test,y_pred)
print("Accuracy:",accuracy)
```

```
Accuracy: 0.9649122807017544
```

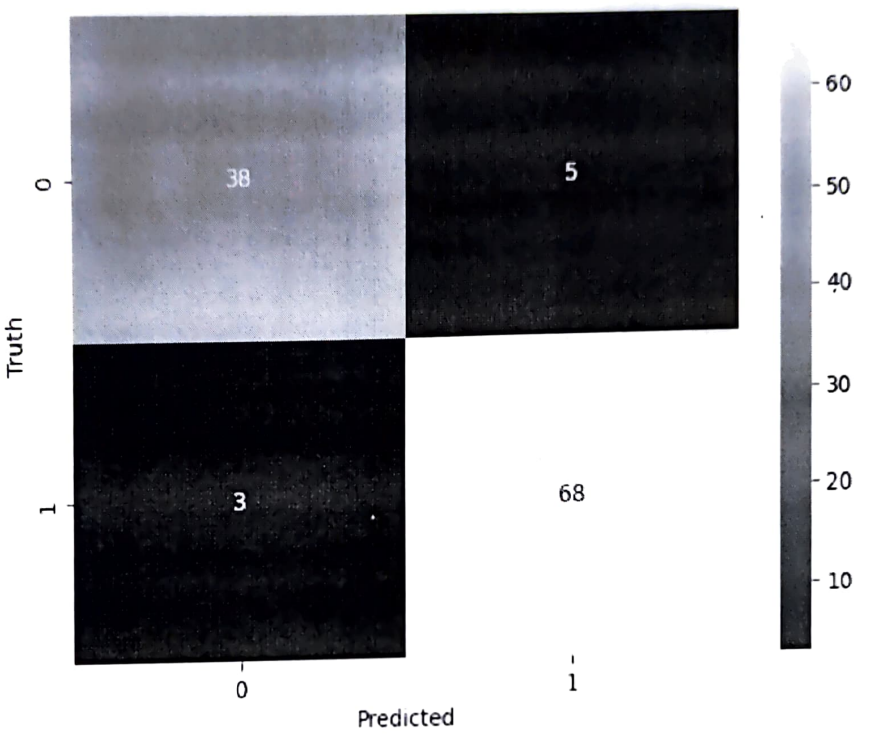
[20]:
```
from sklearn import tree
plt.figure(figsize=(20, 15))
#for i in range(5):
    #plt.subplot(5,5, i +1)
tree.plot_tree(rf.estimators_[75], filled=True, feature_names=breast_cancer.
  feature_names, class_names=breast_cancer.target_names)
plt.show()
```



[21]:
```
y_pred_test = cross_val_predict(rf, X_test, y_test, cv=5)
conf_mat = confusion_matrix(y_test, y_pred_test)
class_report = classification_report(y_test, y_pred_test)
```

[22]:
```
import seaborn as sns
plt.figure(figsize=(7,5))
sns.heatmap(conf_mat, annot=True)
plt.xlabel('Predicted')
plt.ylabel('Truth')
```

[22]:
```
Text(58.222222222222214, 0.5, 'Truth')
```



[23]:
```
print("Confusion Matrix:")
print(conf_mat)
print("\nClassification Report:")
print(class_report)
```

```
Confusion Matrix:
[[38  5]
 [ 3 68]]

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.88      0.90        43
           1       0.93      0.96      0.94        71

    accuracy                           0.93       114
   macro avg       0.93      0.92      0.92       114
weighted avg       0.93      0.93      0.93       114
```

[24]:
```
print("cross_val_predict ")
print(cross_val_predict)
```

```
cross_val_predict
<function cross_val_predict at 0x7f1f33c34ee0>
```

[ ]: