ml5

November 3, 2024

Ensemble Learning: Implement Random Forest Classifier model to predict the safety of the car.

[36]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import␣
 ↪accuracy_score,confusion_matrix,classification_report
```

[37]:
```
df=pd.read_csv(r"C:\Users\dell\Desktop\DMV and ML\ML Datasets\car_evaluation.
 ↪csv")
```

[38]:
```
df.head()
```

[38]:
```
  Buying price Maintance cost No of doors No of persons lug_boot safety  \
0        vhigh          vhigh           2             2    small    low
1        vhigh          vhigh           2             2    small    med
2        vhigh          vhigh           2             2    small   high
3        vhigh          vhigh           2             2      med    low
4        vhigh          vhigh           2             2      med    med

  Decision
0    unacc
1    unacc
2    unacc
3    unacc
4    unacc
```

[39]:
```
# columns = ["buying", "maint", "doors", "persons", "lug_boot", "safety",␣
 ↪"class"]
# df.columns = columns
```

[40]:
```
df.head()
```

[40]:
```
  Buying price Maintance cost No of doors No of persons lug_boot safety  \
0        vhigh          vhigh           2             2    small    low
1        vhigh          vhigh           2             2    small    med
2        vhigh          vhigh           2             2    small   high
```

```
3         vhigh          vhigh           2             2      med    low
4         vhigh          vhigh           2             2      med    med

  Decision
0    unacc
1    unacc
2    unacc
3    unacc
4    unacc
```

[41]:
```
df.dtypes
```

[41]:
```
Buying price      object
Maintance cost    object
No of doors       object
No of persons     object
lug_boot          object
safety            object
Decision          object
dtype: object
```

[42]:
```
# df = pd.get_dummies(df,columns=columns[:-1],drop_first=True)
# df
```

[43]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
for col in df.columns[:-1]:
    df[col] = le.fit_transform(df[col])
```

[44]:
```
df
```

[44]:
```
      Buying price  Maintance cost  No of doors  No of persons  lug_boot  \
0                3               3            0              0         2
1                3               3            0              0         2
2                3               3            0              0         2
3                3               3            0              0         1
4                3               3            0              0         1
…              …             …          …            …        …
1723             1               1            3              2         1
1724             1               1            3              2         1
1725             1               1            3              2         0
1726             1               1            3              2         0
1727             1               1            3              2         0

      safety Decision
0          1    unacc
1          2    unacc
```

```
2          0    unacc
3          1    unacc
4          2    unacc
…       …      …
1723       2     good
1724       0    vgood
1725       1    unacc
1726       2     good
1727       0    vgood

[1728 rows x 7 columns]
```

[45]:
```
x=df.drop("Decision",axis=1)
y=df["Decision"]
```

[46]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪2,random_state=42)
x_train.shape,x_test.shape
```

[46]:
```
((1382, 6), (346, 6))
```

[47]:
```
model=RandomForestClassifier(random_state=42)
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
```

[48]:
```
matrix=confusion_matrix(y_test,y_pred)
print(matrix)
```

```
[[ 75   6   2   0]
 [  0  11   0   0]
 [  0   0 235   0]
 [  1   0   0  16]]
```

[51]:
```
acc_score=accuracy_score(y_test,y_pred)*100
print(f"Accuracy: {accuracy_score(y_test,y_pred)*100}%")
print(acc_score)
```

```
Accuracy: 97.39884393063583%
97.39884393063583
```

[50]:
```
report=classification_report(y_test,y_pred)
print(report)
```

```
              precision    recall  f1-score   support

         acc       0.99      0.90      0.94        83
        good       0.65      1.00      0.79        11
       unacc       0.99      1.00      1.00       235
       vgood       1.00      0.94      0.97        17

    accuracy                           0.97       346
   macro avg       0.91      0.96      0.92       346
weighted avg       0.98      0.97      0.98       346
```

Random Forest is an ensemble learning method used for classification and regression tasks. It builds a collection of decision trees (each trained on a random subset of the data and a random subset of features). Each tree provides a prediction, and the Random Forest combines these predictions by averaging them in regression or taking the majority vote in classification. This approach improves accuracy, reduces overfitting, and enhances the model's ability to generalize well to unseen data.