CS156 (Introduction to AI), Fall 2022

Homework 5 submission

Roster Name: Preet LNU

Student ID: 014755741

Email address: preet.lnu@sjsu.edu

References and sources

https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

DecisionTrees.Breast.ipynb

Solution

Load libraries and set random number generator seed

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import plot_confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn import tree
```

```
np.random.seed(42)
```

Code the solution

```
airline_file = pd.read_csv(r'/content/homework5_input_data.csv')
```

1. Load the dataset.

```
df = pd.DataFrame(airline_file, columns=airline_file.columns)
df.head()
columns = df.columns[:-1]
X = df[columns]
Y = df['satisfaction']
df['satisfaction'] = Y

class_names = ['neutral or dissatisfied', 'satisfied']

print(X.shape, Y.shape)
```

```
(103594, 22) (103594,)
```

```
df.describe()
```

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking
count	103594.000000	103594.000000	103594.000000	103594.000000	103594.000000
mean	39.380466	1189.325202	2.729753	3.060081	2.756984
std	15.113125	997.297235	1.327866	1.525233	1.398934
min	7.000000	31.000000	0.000000	0.000000	0.000000
25%	27.000000	414.000000	2.000000	2.000000	2.000000
50%	40.000000	842.000000	3.000000	3.000000	3.000000
75%	51.000000	1743.000000	4.000000	4.000000	4.000000
max	85.000000	4983.000000	5.000000	5.000000	5.000000

2. Convert categorical variables to numeric format

```
unconverted = ['Gender', 'Customer Type', 'Type of Travel', 'Class']

int_df = df.select_dtypes(include=['int64', 'float64']).copy()

df_numeric = pd.get_dummies(df, columns=unconverted, prefix=unconverted)
df_numeric
```

	Age	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	On boar
0	13	460	3	4	3	1	5	
1	25	235	3	2	3	3	1	
2	26	1142	2	2	2	2	5	
3	25	562	2	5	5	5	2	
4	61	214	3	3	3	3	4	
...	...	...	...	...	...	...	...	
103589	23	192	2	1	2	3	2	
103590	49	2347	4	4	4	4	2	
103591	30	1995	1	1	1	3	4	
103592	22	1000	1	1	1	5	1	
103593	27	1723	1	3	3	3	1	

103594 rows × 28 columns

```
satisfaction_col = df_numeric['satisfaction']
df_numeric = df_numeric.drop(columns=['satisfaction'])
df_numeric.insert(loc=27, column='satisfaction', value=satisfaction_col)

print(df_numeric)

new_columns = df_numeric.columns[:-1]
X_new = df_numeric[new_columns]
Y_new = df_numeric['satisfaction']

print(Y_new)
df_numeric['satisfaction'] = Y_new
```

```
        Age  Flight Distance  Inflight wifi service  \
0        13              460                      3
1        25              235                      3
2        26             1142                      2
```

```
3        25              562                      2
4        61              214                      3
...     ...              ...                    ...
103589   23              192                      2
103590   49             2347                      4
103591   30             1995                      1
103592   22             1000                      1
103593   27             1723                      1

        Departure/Arrival time convenient  Ease of Online booking  \
0                                       4                       3
1                                       2                       3
2                                       2                       2
3                                       5                       5
4                                       3                       3
...                                   ...                     ...
103589                                  1                       2
103590                                  4                       4
103591                                  1                       1
103592                                  1                       1
103593                                  3                       3

        Gate location  Food and drink  Online boarding  Seat comfort  \
0                   1               5                3             5
1                   3               1                3             1
2                   2               5                5             5
3                   5               2                2             2
4                   3               4                5             5
...               ...             ...              ...           ...
103589              3               2                2             2
103590              4               2                4             5
103591              3               4                1             5
103592              5               1                1             1
103593              3               1                1             1

        Inflight entertainment  ...  Gender_Female  Gender_Male  \
0                            5  ...              0            1
1                            1  ...              0            1
2                            5  ...              1            0
3                            2  ...              1            0
4                            3  ...              0            1
...                        ...  ...            ...          ...
103589                       2  ...              1            0
103590                       5  ...              0            1
103591                       4  ...              0            1
103592                       1  ...              1            0
103593                       1  ...              0            1

        Customer Type_Loyal Customer  Customer Type_disloyal Customer   \
0                                  1                                0
1                                  0                                1
2                                  1                                0
3                                  1                                0
4                                  1                                0
```

3. Break the data into the training and test datasets.

```
X_train, X_test, Y_train, Y_test = train_test_split(X_new, Y_new, test_size=0.2, rando
X_train.shape, Y_train.shape, X_test.shape, Y_test.shape
```

```
    ((82875, 27), (82875,), (20719, 27), (20719,))
```

4. Train a decision tree model to predict the class variable. Report 5-fold cross-validation accuracies.

```
model = DecisionTreeClassifier(random_state=0)

cross_vals = cross_val_score(model, X_train, Y_train, cv=5)
print('Individual cross-validation accuracies: ' + str(cross_vals))
print('Mean cross validation accuracy: ' + str(cross_vals.mean()))
```

```
    Individual cross-validation accuracies: [0.94365008 0.94129713 0.94449472 0.9453
    Mean cross validation accuracy: 0.9435414781297133
```

5. Train a decision tree model on all the training data and report prediction accuracy on the test data.

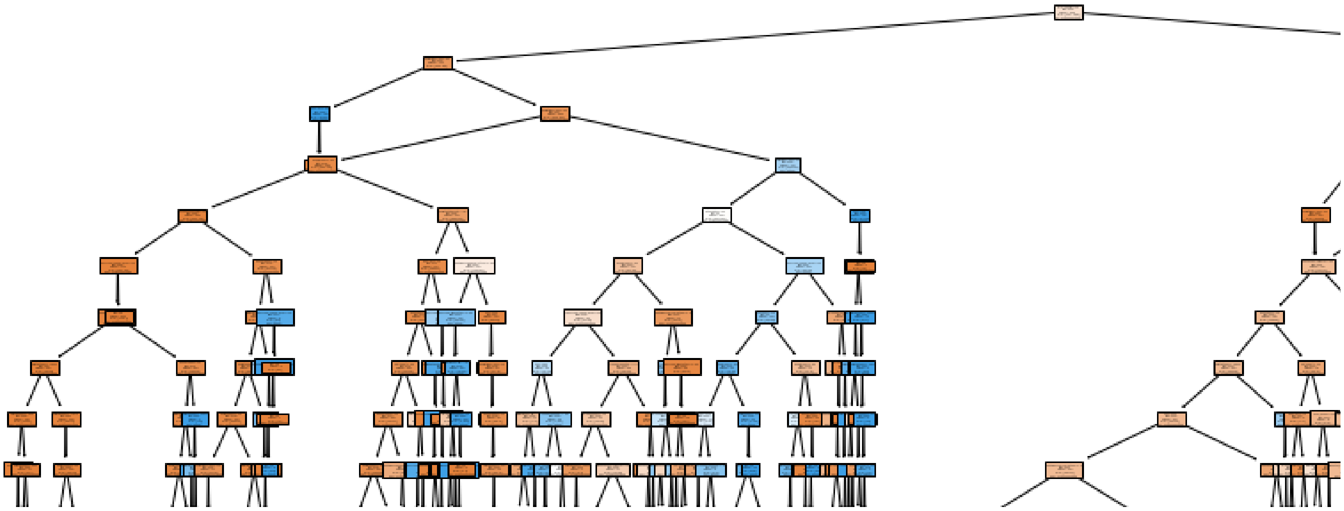
```
model.fit(X_train, Y_train)

print('Accuracy of decision tree model on training set: {:.2f}'.format(model.score(X_t

print('Accuracy of decision tree model on test set: {:.2f}'.format(model.score(X_test,
```

```
    Accuracy of decision tree model on training set: 1.00
    Accuracy of decision tree model on test set: 0.95
```

```
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(model, feature_names=new_columns, class_names=class_names, filled=T
```



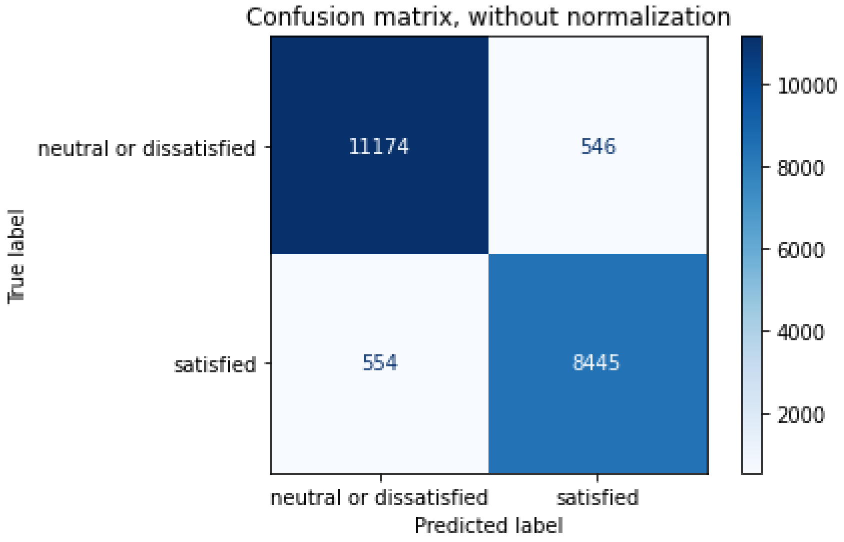
6. Plot two confusion matrices for test set predictions

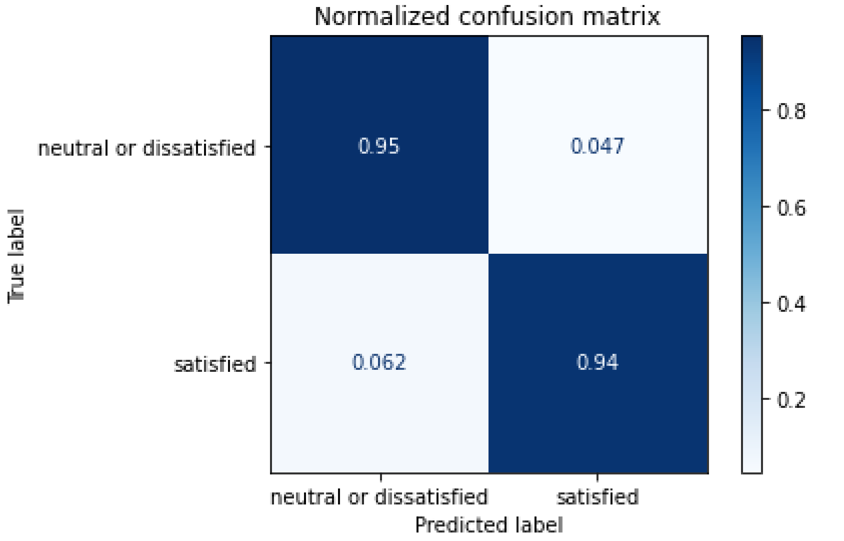
```
np.set_printoptions(precision=2)
titles_options = [("Confusion matrix, without normalization", None),
                  ("Normalized confusion matrix", 'true')]
for title, normalize in titles_options:
    disp = plot_confusion_matrix(model, X_test, Y_test,
                                 display_labels=class_names,
                                 cmap=plt.cm.Blues,
                                 normalize=normalize)
    disp.ax_.set_title(title)

    print(title)
    print(disp.confusion_matrix)

plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWa
  warnings.warn(msg, category=FutureWarning)
Confusion matrix, without normalization
[[11174   546]
 [  554  8445]]
Normalized confusion matrix
[[0.95 0.05]
 [0.06 0.94]]
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWa
  warnings.warn(msg, category=FutureWarning)
```


Confusion matrix, without normalization
True label
neutral or dissatisfied
satisfied
neutral or dissatisfied
satisfied
Predicted label


Normalized confusion matrix
True label
neutral or dissatisfied
satisfied
neutral or dissatisfied
satisfied
Predicted label