

CS156 (Introduction to AI), Spring 2021

Homework 5 submission

Roster Name: Neeval Kumar

Preferred Name (if different): Chosen Name

Student ID: 011877086

Email address:kumar.neeval@gmail.com

Any special notes or anything you would like to communicate to me about this homework submission goes in here.

References and sources

List all your references and sources here. This includes all sites/discussion boards/blogs/posts/etc. where you grabbed some code examples.

Solution

Load libraries and set random number generator seed

```
In [45]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
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
from string import ascii_lowercase
```

```
In [46]: np.random.seed(42)
```

Read csv, create ascii dictionary

```
In [47]: mushroom_df = pd.read_csv('homework5_input_data.csv')
letters = {}

for i in range(len(ascii_lowercase)):
    letters[ascii_lowercase[i]] = i

letters['?'] = 26
mushroom_df = mushroom_df.replace(letters)
mushroom_df.astype(np.float64)
Y = mushroom_df['class']
X = mushroom_df.drop(columns = ['class'])
```

Split, cross validation score, and accuracy

```
In [48]: X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)
model = DecisionTreeClassifier(random_state=0)
scores = cross_val_score(model, X_train, Y_train, cv=5)
print("Mean accuracy of all 5 folds: ", np.mean(scores))
```

Mean accuracy of all 5 folds: 1.0

```
In [49]: model.fit(X_train, Y_train)

print('Accuracy of Decision tree model on training set: {:.2f}'.format(model.score(X_train, Y_train)))

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

Accuracy of Decision tree model on training set: 1.00
Accuracy of Decision tree model on test set: 1.00

Confusion Matrix

```
In [50]: 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=['p', 'e'],
                                cmap=plt.cm.Blues,
                                normalize=normalize)

    disp.ax_.set_title(title)

    print(title)
    print(disp.confusion_matrix)

plt.show()
```

Confusion matrix, without normalization

```
[[852  0]
 [ 0 773]]
```

Normalized confusion matrix

```
[[1. 0.]
 [0. 1.]]
```

