

UNIT 3.2 GRADED ASSIGNMENT

Group members

Ifra Saleem (2303.khi.deg.003)

Umaina Siddiqui (2023.KHI.DEG.033)

UNIT 3.2 GRADED ASSIGNMENT

Task:

Implement a single classification model of your choice and try to achieve at least an 80% F1 score on the wine dataset provided by Sklearn.

Solution:

```
[58]: import pandas as pd
df = pd.DataFrame(wine.data, columns = wine.feature_names)
df
```

```
[58]:
```

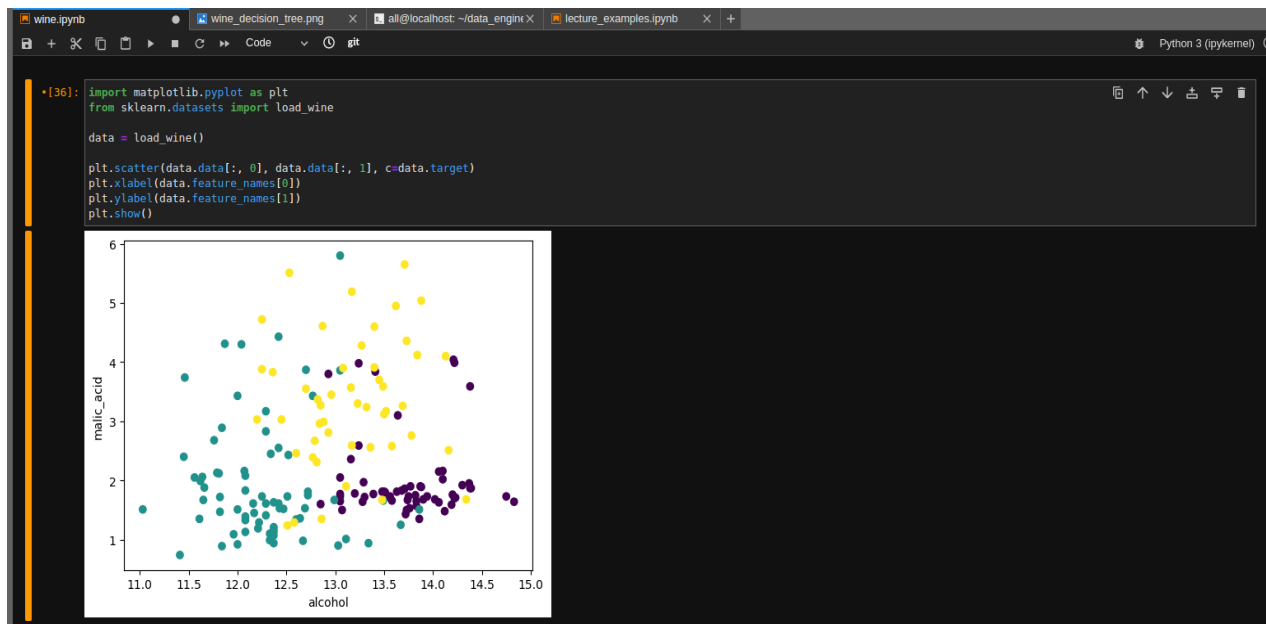
	alcohol	malic_acid	ash	alkalinity_of_ash	magnesium	total_phenols	flavonoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline			
0	14.23	1.71	2.43		15.6	127.0	2.80	3.06		0.28	2.29	5.64	1.04		3.92	1065.0
1	13.20	1.78	2.14		11.2	100.0	2.65	2.76		0.26	1.28	4.38	1.05		3.40	1050.0
2	13.16	2.36	2.67		18.6	101.0	2.80	3.24		0.30	2.81	5.68	1.03		3.17	1185.0
3	14.37	1.95	2.50		16.8	113.0	3.85	3.49		0.24	2.18	7.80	0.86		3.45	1480.0
4	13.24	2.59	2.87		21.0	118.0	2.80	2.69		0.39	1.82	4.32	1.04		2.93	735.0
...
173	13.71	5.65	2.45		20.5	96.0	1.68	0.61		0.52	1.06	7.70	0.64		1.74	740.0
174	13.40	3.91	2.48		23.0	102.0	1.80	0.75		0.43	1.41	7.30	0.70		1.56	750.0
175	13.27	4.28	2.26		20.0	120.0	1.59	0.69		0.43	1.35	10.20	0.59		1.56	835.0
176	13.17	2.59	2.37		20.0	120.0	1.65	0.68		0.53	1.46	9.30	0.60		1.62	840.0
177	14.13	4.10	2.74		24.5	96.0	2.05	0.76		0.56	1.35	9.20	0.61		1.60	560.0

178 rows × 13 columns

Information of wine dataset:

[illegible]

Visualization of the first two features of wine dataset:



By visualizing we can clearly see that the clusters are not well separated so we can apply KNN, decision tree, random forest and logistic regression classification models to this dataset.

k-NN (K nearest Neighbour)

```
[51]: from sklearn.neighbors import KNeighborsClassifier

[52]: wine = datasets.load_wine()
      wine_x = wine.data[:, 0:] #last two columns
      wine_y = wine.target

[53]: x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.2)
      scaler = StandardScaler()
      x_train = scaler.fit_transform(x_train)
      x_test = scaler.transform(x_test)

[54]: model = KNeighborsClassifier()

[55]: model.fit(x_train, y_train)

[55]: KNeighborsClassifier()

[57]: y_pred = model.predict(x_test)
      precision_score(y_test, y_pred, average="micro")

[57]: 0.9722222222222222
```

Random Forest

```
[48]: from sklearn.ensemble import RandomForestClassifier

      wine_x = wine.data
      wine_y = wine.target
      x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.2)
      model = RandomForestClassifier()
      model.fit(x_train, y_train)
      y_pred = model.predict(x_test)
      f1_score(y_test, y_pred, average="micro")

[48]: 0.9722222222222222
```

Decision Tree

```
[60]: from sklearn.tree import DecisionTreeClassifier

[61]: wine = datasets.load_wine()
      wine_x = wine.data[:, 0:]
      wine_y = wine.target

      x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.3)

      model = DecisionTreeClassifier(max_depth=3)
      model.fit(x_train, y_train)

      y_pred = model.predict(x_test)

[62]: f1_score(y_test, y_pred, average="micro")

[62]: 0.8703703703703703
```

```
[65]: export_graphviz(
      model,
      out_file="wine_decision_tree.dot",
      feature_names=wine.feature_names[0:],
      class_names=wine.target_names,
      rounded=True,
      filled=True)

[48]: from sklearn.ensemble import RandomForestClassifier

      wine_x = wine.data
      wine_y = wine.target
      x_train, x_test, y_train, y_test = train_test_split(wine_x, wine_y, test_size=0.2)
      model = RandomForestClassifier()
      model.fit(x_train, y_train)
      y_pred = model.predict(x_test)
      f1_score(y_test, y_pred, average="micro")
```

