cross-validation

December 20, 2023

0.1 This is a project of finding the best model for our data using cross validation

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df = pd.read_excel("Pumpkin.xlsx")
```

[3]:
```
df
```

[3]:

	Area	Perimeter	Major_Axis_Length	Minor_Axis_Length	Convex_Area	\
0	56276	888.242	326.1485	220.2388	56831	
1	76631	1068.146	417.1932	234.2289	77280	
2	71623	1082.987	435.8328	211.0457	72663	
3	66458	992.051	381.5638	222.5322	67118	
4	66107	998.146	383.8883	220.4545	67117	
…	…	…	…	…	…	
2495	79637	1224.710	533.1513	190.4367	80381	
2496	69647	1084.318	462.9416	191.8210	70216	
2497	87994	1210.314	507.2200	222.1872	88702	
2498	80011	1182.947	501.9065	204.7531	80902	
2499	84934	1159.933	462.8951	234.5597	85781	

	Equiv_Diameter	Eccentricity	Solidity	Extent	Roundness	\
0	267.6805	0.7376	0.9902	0.7453	0.8963	
1	312.3614	0.8275	0.9916	0.7151	0.8440	
2	301.9822	0.8749	0.9857	0.7400	0.7674	
3	290.8899	0.8123	0.9902	0.7396	0.8486	
4	290.1207	0.8187	0.9850	0.6752	0.8338	
…	…	…	…	…	…	
2495	318.4289	0.9340	0.9907	0.4888	0.6672	
2496	297.7874	0.9101	0.9919	0.6002	0.7444	
2497	334.7199	0.8990	0.9920	0.7643	0.7549	
2498	319.1758	0.9130	0.9890	0.7374	0.7185	
2499	328.8485	0.8621	0.9901	0.7360	0.7933	

	Aspect_Ration	Compactness	Class
0	1.4809	0.8207	Çerçevelik
1	1.7811	0.7487	Çerçevelik

```
2            2.0651       0.6929     Çerçevelik
3            1.7146       0.7624     Çerçevelik
4            1.7413       0.7557     Çerçevelik
…             …           …           …
2495         2.7996       0.5973  Ürgüp Sivrisi
2496         2.4134       0.6433  Ürgüp Sivrisi
2497         2.2828       0.6599  Ürgüp Sivrisi
2498         2.4513       0.6359  Ürgüp Sivrisi
2499         1.9735       0.7104  Ürgüp Sivrisi

[2500 rows x 13 columns]
```

[4]:
```
from sklearn.preprocessing import LabelEncoder
```

[5]:
```
le = LabelEncoder()
```

[6]:
```
df.Class = le.fit_transform(df.Class)
```

[7]:
```
inputs = df.drop(["Class"],axis=1)
```

[8]:
```
targets = df["Class"]
```

[9]:
```
from sklearn.model_selection import StratifiedKFold
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn import tree
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
```

[10]:
```
tree_classifier = DecisionTreeClassifier()
```

[58]:
```
logi=␣
 ↪cross_val_score(LogisticRegression(solver='liblinear',multi_class='ovr'),inputs,targets,cv=
```

[68]:
```
RFC=cross_val_score(RandomForestClassifier(n_estimators=40),inputs,targets,cv=6)
```

[93]:
```
Svm= cross_val_score(SVC(kernel="linear"),inputs,targets,cv=3)
```

[94]:
```
DT = cross_val_score(tree_classifier,inputs,targets,cv=3)
```

[95]:
```
logiavg=np.mean(logi)
RFCavg=np.mean(RFC)
Svmavg=np.mean(Svm)
Dtree =np.mean(DT)
```

[96]:
```
logiavg
```

[96]: 0.8763937229568087

[97]:
```
RFCavg
```

[97]: 0.883986080366476

[98]:
```
Svmavg
```

[98]: 0.8723926021247829

[99]:
```
Dtree
```

[99]: 0.8323972274761223

0.1.1 Here we compared 4 models such as Logistic regression, Random forest classification, Support vector classification and Decision tree classification using cross validation method. From all the 4, random forest model have high score which is 0.883986080366476. So using random forest model we can predict more accurately than other 3 models for this data set

[ ]:
```

```