«به نام خدا»

تکلیف چهارم – سوال اول – مرضیه علیدادی – 9631983

(کد های مربوط، در دو فرمت py. و ipynb. ضمیمه شده اند. – فایل dot_file.dot نیز ضمیمه شده است.)

.1

b) در این دیتاســـت missing value ای وجود ندارد. مقادیر Null ای وجود ندارد. مقدار 0 برای attribute های این دیتاســـت valid است و نمی توان آن را به عنوان missing value در نظر گرفت. همه ی attribute های input از نوع integer هستند و نمی توانند دارای مقادیری از نوع string , ... (مثلا 'missed') به مفهوم missing value باشند.

داده های عددی را به این صورت normalize کردم:

	COMPACTNESS	CIRCULARITY	DISTANCE_CIRCULARITY	RADIUS_RATIO	PR.AXIS_ASPECT_RATIO	MAX.LENGTH_ASPECT_RATIO	SCATTER_RATIO	ELON
0	0.150904	0.076246	0.131843	0.282747	0.114370	0.015885	0.257332	
1	0.158697	0.071501	0.146489	0.245893	0.099403	0.015695	0.259844	
2	0.122330	0.058812	0.124682	0.245836	0.077632	0.011762	0.243483	
3	0.166015	0.073190	0.146379	0.283833	0.112462	0.016066	0.257056	
4	0.135075	0.069921	0.111238	0.325770	0.163679	0.082634	0.236779	
...	...	...	...	...	...	...	...	
841	0.142922	0.059935	0.133701	0.281233	0.098355	0.012294	0.259718	
842	0.145091	0.074991	0.136940	0.265729	0.107596	0.017933	0.259208	
843	0.114897	0.058532	0.109477	0.240633	0.072624	0.013007	0.240633	
844	0.164335	0.068791	0.149048	0.278987	0.110831	0.013376	0.257968	
845	0.179869	0.076180	0.139663	0.260280	0.116386	0.010581	0.253932	

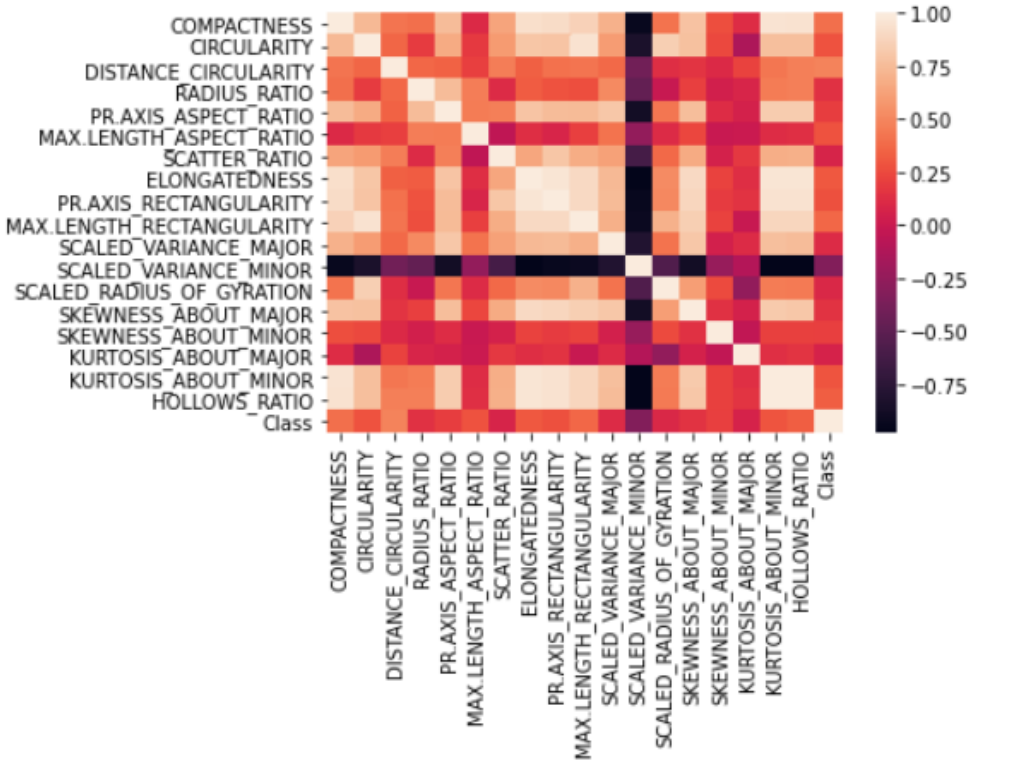
846 rows × 19 columns

برای encode کردن متغیر Class، با توجه به اینکه جز input ها نیست و با توجه به اینکه در تحلیل ها اثر ندارد، حساسیتی روی مقادیری که می گیرد، نداریم. پس از Ordinal encoding ساده استفاده کردم. و به این صورت شد:

F_GYRATION	SKEWNESS_ABOUT_MAJOR	SKEWNESS_ABOUT_MINOR	KURTOSIS_ABOUT_MAJOR	KURTOSIS_ABOUT_MINOR	HOLLOWS_RATIO	Class
0.292278	0.111193	0.009531	0.025415	0.297043	0.312928	3.0
0.275539	0.125562	0.015695	0.024415	0.329601	0.347040	3.0
0.258774	0.085866	0.016467	0.010586	0.221134	0.230544	2.0
0.226709	0.112462	0.010711	0.017851	0.355237	0.369518	3.0
0.298755	0.201818	0.014302	0.017480	0.286042	0.290809	0.0
...	...	...	...	...	...	...
0.228982	0.110649	0.010758	0.038420	0.288917	0.299675	2.0
0.286922	0.117377	0.001630	0.032605	0.303224	0.321157	3.0
0.216787	0.075875	0.003252	0.004336	0.202696	0.217871	2.0
0.282809	0.126118	0.000000	0.047772	0.363066	0.372620	2.0
0.277209	0.154475	0.002116	0.038090	0.393595	0.402059	3.0

c) در این نمودار، نزدیک ترین رنگ ها به سـفید و مشـکی، نشـان دهنده ی بیشـترین correlation هسـتند. طبق نمودار heatmap، 6 تا attribute هایی که بیشرین correlation را با Class دارند، به ترتیب از بیشترین به کمترین، عبارتند از:
(ولی رابطه ی قابل توجه و نزدیک به 1 ای ندارند.)

DISTANCE_CIRCULARITY
COMPACTNESS
MAX.LENGTH_RECTANGULARITY
HOLLOWS_RATIO
SCALED_VARIANCE_MINOR
ELONGATEDNESS



e) توزیع دسته های ستون Class، به ترتیب در دو مجموعه ی آموزش و تست:

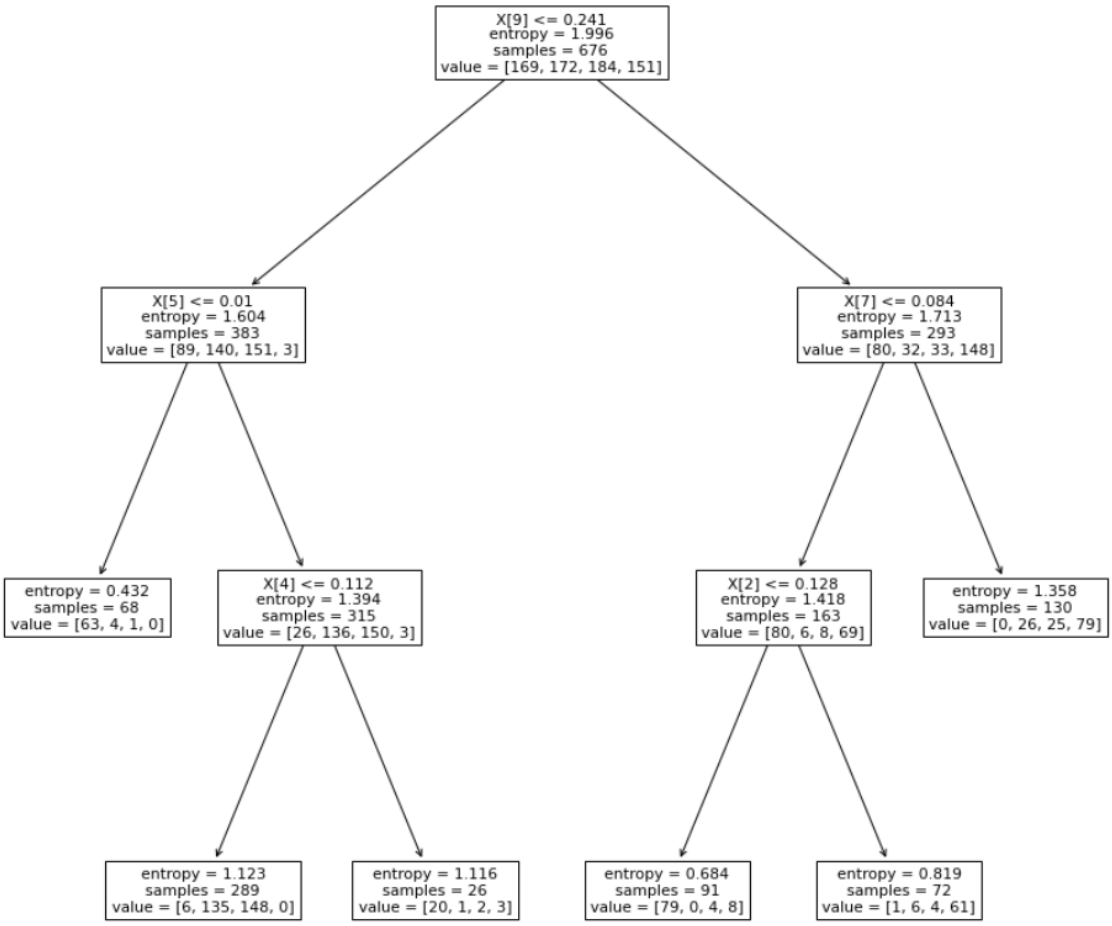
```
y_train.value_counts()
```

```
2.0    184
1.0    172
0.0    169
3.0    151
Name: Class, dtype: int64
```

```
y_test.value_counts()
```

```
0.0    49
3.0    48
1.0    40
2.0    33
Name: Class, dtype: int64
```

(f



g) با اســتفاده از قابلیت های sklearn با اســتفاده از درخت تصــمیمی که در بخش قبل تولید کرده بودم، متغیر هدف را برای داده های تست پیشبینی کردم. نتیجه را با مقادیر واقعی متغیر هدف مقایسه کردم. بدین صورت شد:

```
print(confusion_matrix(y_test, y_pred))
```

```
[[44  0  4  1]
 [ 1  0 31  8]
 [ 0  0 29  4]
 [ 5  0  0 43]]
```

```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

         0.0       0.88      0.90      0.89        49
         1.0       0.00      0.00      0.00        40
         2.0       0.45      0.88      0.60        33
         3.0       0.77      0.90      0.83        48

    accuracy                           0.68       170
   macro avg       0.53      0.67      0.58       170
weighted avg       0.56      0.68      0.61       170
```

- ماتریس تهیه شده بدین صورت تفسیر می شود:

هر سطر نشان دهنده ی یکی از دسته های واقعی (actual) است. و هر ستون نشاندهنده ی یکی از دسته های پیشبینی شده.

داده هایی که در دسته ی 0 قرار می گیرند، 49 تا هستند. که 44 تا از آن ها درست پیشبینی شده اند.

داده هایی که در دسته ی 1 قرار می گیرند، 40 تا هستند. که 0 تا از آن ها درست پیشبینی شده اند.

داده هایی که در دسته ی 2 قرار می گیرند، 33 تا هستند. که 29 تا از آن ها درست پیشبینی شده اند.

داده هایی که در دسته ی 3 قرار می گیرند، 48 تا هستند. که 43 تا از آن ها درست پیشبینی شده اند.

- گزارشی که در ادامه آمده، همین مطالب را به صورت درصدی بیان می کند:

دقت پیشبینی داده های دسته های 0 و 1 و 2 و 3، به ترتیب برابر 90% و 0% و 88% و 90% است.

و به طور کلی، دقت مدل برابر 68% است.

h)

- max_features نشـــاندهنده ی حداکثر تعداد feature هایی اســـت، که در یک درخت واحد می توانند دخیل باشـــند.

آپشن های متفاوتی برای اسفاده از این پارامتر داریم:

Auto/None: همه ی feature های مورد نیاز و موثر را در درخت تصـــمیم می آورد به تشـــخیص خودش. اگر کلا از این پارامتر استفاده نکنیم، به صورت پیش فرض این آپشن اجرا می شود.

sqrt: ریشـــه دوم تعداد feature ها را حســـاب می کند. و از آن به عنوان تعداد feature های دخیل در درخت تصمیم استفاده می کند. یک آپشن دیگر به همین صورت با $\log_2$ هم موجود است.

0.x: تعداد x% از تعداد کل feature ها را حســـاب می کند. و به همان تعداد از feature ها را در درخت دخیل می‌کند.

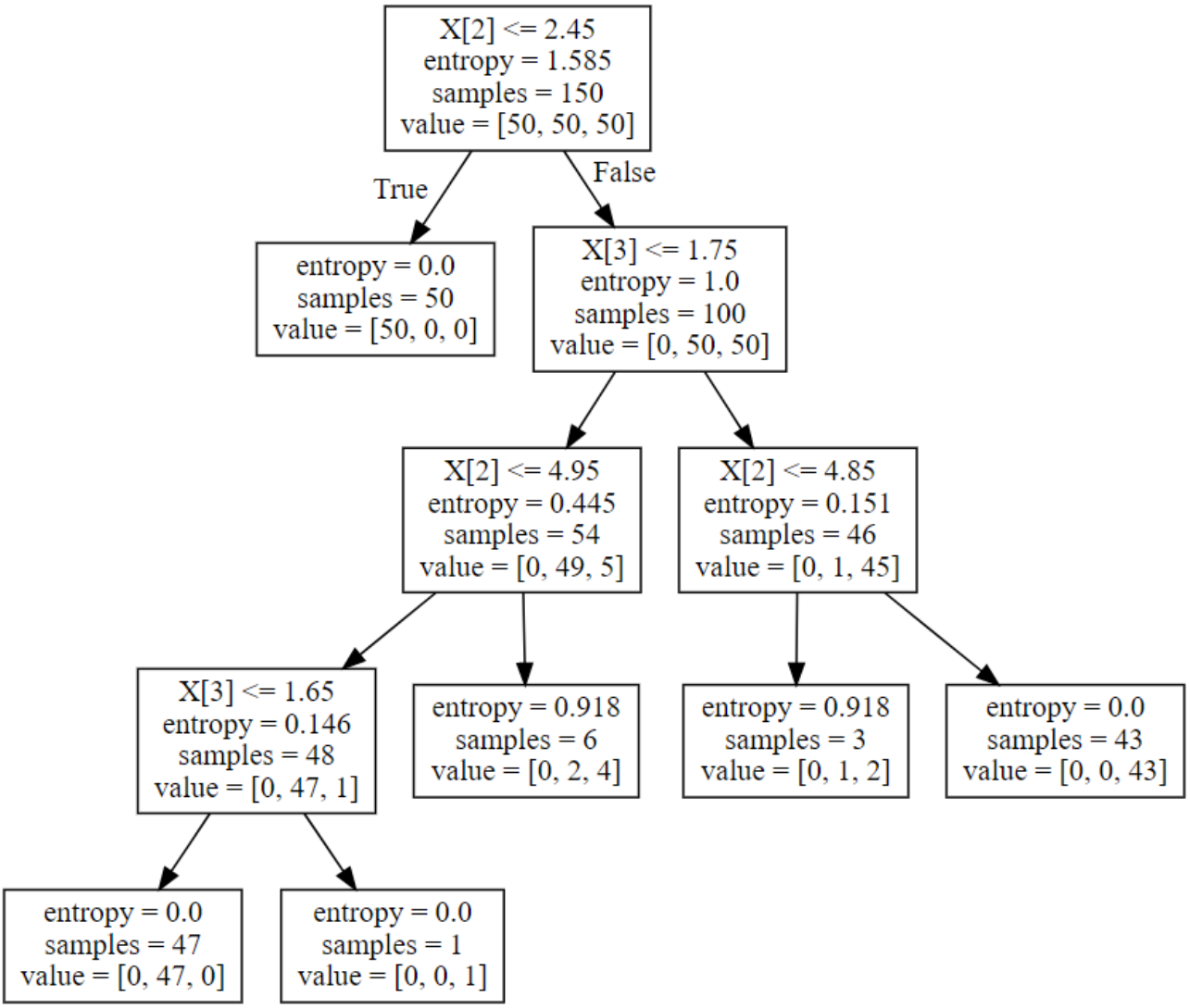
- max_leaf_nodes: تعداد حداکثر node های انتهایی (برگ های) درخت را مشخص می کند.

i) یک فایل با فرمت dot. ساختم و نتیجه ی این متد را در آن ذخیره کردم:

```
<_io.TextIOWrapper name='Desktop/dot_file.dot' mode='w' encoding='cp1256'>
```

فایل ضمیمه شده است.

گراف حاصل از فایل dot_file.dot:


X[2] <= 2.45
entropy = 1.585
samples = 150
value = [50, 50, 50]
True
False
entropy = 0.0
samples = 50
value = [50, 0, 0]
X[3] <= 1.75
entropy = 1.0
samples = 100
value = [0, 50, 50]
X[2] <= 4.95
entropy = 0.445
samples = 54
value = [0, 49, 5]
X[2] <= 4.85
entropy = 0.151
samples = 46
value = [0, 1, 45]
X[3] <= 1.65
entropy = 0.146
samples = 48
value = [0, 47, 1]
entropy = 0.918
samples = 6
value = [0, 2, 4]
entropy = 0.918
samples = 3
value = [0, 1, 2]
entropy = 0.0
samples = 43
value = [0, 0, 43]
entropy = 0.0
samples = 47
value = [0, 47, 0]
entropy = 0.0
samples = 1
value = [0, 0, 1]