مبانی هوش محاسباتی (پروژه دوم)

کیارش آستان‌بوس ۴۰۰۱۲۶۲۵۷۰
محمدرضا نادری ۴۰۰۱۲۶۲۳۸۶

مهر ۱۴۰۳

فهرست مطالب

۱) فاز ۱ **۲**

۱.۱) Raw Data ۲

۲.۱) Sobel ۲

۳.۱) Sobel + hog ۲

۴.۱) filter + hog ۲

۵.۱) نتیجه نهایی ۳

۲) فاز ۲ **۴**

۱.۲) Raw Data ۴

۲.۲) hog + Sobel ۵

۳) فاز ۳ **۶**

۱.۳) Decision Tree ۶

۱.۱.۳) Raw ۶

۲.۱.۳) hog + Sobel ۷

۳.۱.۳) Sobel ۸

۴.۱.۳) Laplacian + hog ۹

۵.۱.۳) نمایش بهترین درخت ۱۰

۶.۱.۳) مقایسه و نتیجه گیری ۱۱

۲.۳) SVM ۱۱

۱.۲.۳) Raw ۱۲

۲.۲.۳) hog + Sobel ۱۳

۳.۳) مقایسه درخت تصمیم و SVM ۱۴

۴) فاز ۴ **۱۶**

۱.۴) DT ۱۶

۲.۴) SVM ۱۷

۵) فاز ۵ **۱۹**

۱) فاز ۱

۱.۱ Raw Data

برای بدست آوردن feature های دیتای خام تصویر را که ابعاد ۲۸*۲۸ دارد را با استفاده از تابع flatten به یک آرایه یک بعد و به اندازه ۷۸۴ تبدیل میکنیم.

۱.۲ Sobel

فیلتر سوبل را با توجه به کرنل زیر با عکس خام که ۲۸*۲۸ است convolve کردیم تا فیلتر سوبل رو عکس اعمال شود. و سپس عکس خروجی یک عکسی ۲۸*۲۸ است که فیلتر سوبل روی آن اعمال شده است و حالا برای بدست آوردن feature ها باید مجدد با استفاده از تابع flatten آن را به یک آرایه یک بعدی به اندازه ۷۸۴ تبدیل میکنیم.

$$G_x = A \cdot \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

$$G_y = A \cdot \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$G = \sqrt{G_x \cdot G_y}$$

۱.۳ Sobel + hog

در این مورد باید باید دو feature را که از سوبل و HOG میگیریم را کنارهم قرار دهیم. برای به دست آوردن HOG از تابع extract_hog_features که در این تابع از تابع آماده hog از کتابخانه skimage است استفاده میکنیم تابع hog به ما دو خروجی features و image_HOG جدید را میدهد که برای ادامه به features نیاز داریم. خروجی که hog میدهد به پارامتر های متفاوتی نیاز دارد: Orientations ، cells per Pixel ، block per cells که به ترتیب بهترین خروجی را برای ما مقدارهای ۹ و (۴ ، ۴) و (۲ ، ۲) داده است و تعداد ۱۲۹۶ فیچر را به ما میدهد که با استفاده از تابع combine_features_batch فیچرهایی که از سوبل به دست آماده و فیچرهای سوبل مرحله قبل را کنار هم قرار میدهیم و یک آرایه به اندازه ۲۰۸۰ را به ما میدهد.

۱.۴ filter + hog

برای بدست آوردن این دو feature مثل قبل تصویر خام را در kernel مورد استفاده برای Laplacian و Sharpening با استفاده از تابع apply_convolution ضرب میکنیم و حاصل را با خروجی feature موجود در مرحله قبل concat میکنیم. حال این خروجی ها آماده برای فاز دو هستند.

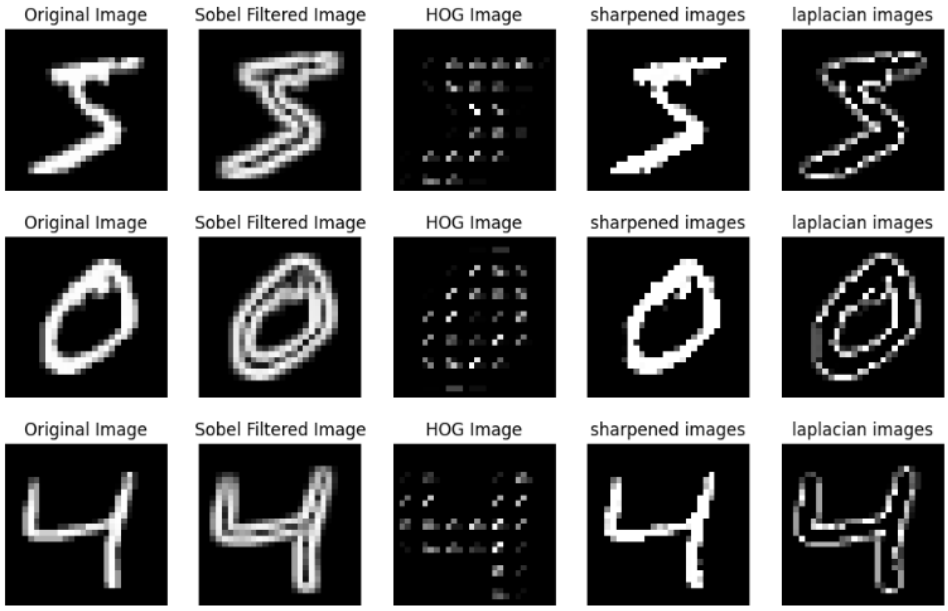
$$S = A \cdot \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$L = A \cdot \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

۵.۱) نتیجه نهایی

در زیر تاثیر هر فیلتر را با توجه به فیلتر های تعریف شده مشخص کردیم

- در sobel و Laplacian لبه های تصویر مشخص میشود
- در sharpening تصویر شارپ تر میشود
- و در hog جهت تغییر را در تصویر مشخص میکند.



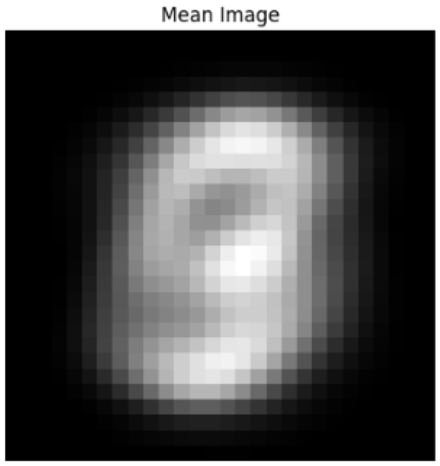
شکل ۱: نتیجه نهایی

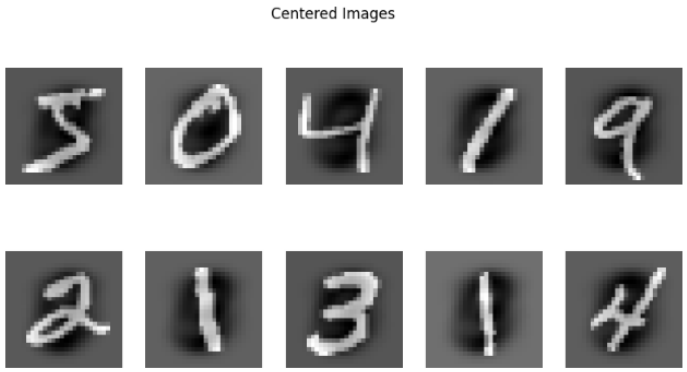
۲) فاز ۲

در این فاز برای عکس های خام و همچنین بردار ویژگی حاصل از hog + sobel به ترتیب میانگین را پیدا میکنیم. سپس centered را محاسبه میکنیم و در نهایت بر روی دیتا ،centered عمل کاهش ابعاد با استفاده از PCA انجام میدهیم.

۱.(۲ Raw Data

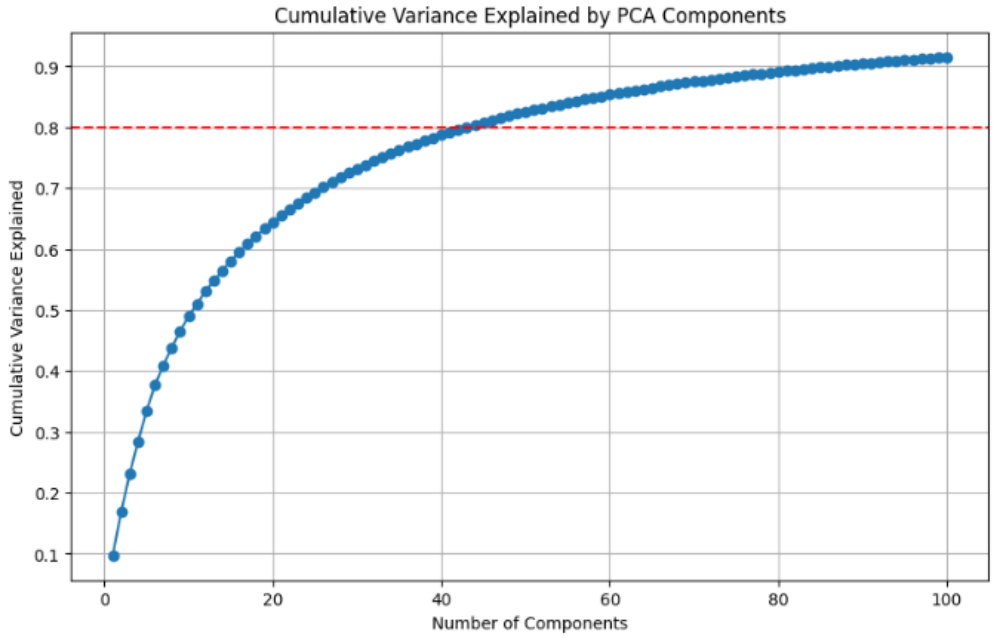
در تصاویر زیر میتوان نتیجه میانگین و همچنین چند نمونه از تصاویر centered را مشاهده کرد.





برای اعمال PCA نیز با استفاده از cumulative explained variance بررسی میکنیم به ازای هر n_component جقدر از واریانس پوشش داده می‌شود. سپس ایn_component که حداقل ۸۰ درصد واریانس را پوشش بدهد را انتخاب میکنیم تا کاهش ابعاد صورت بگیرد. که در اینجا با توجه به نمودار مقدار ۴۳ انتخاب شده است.

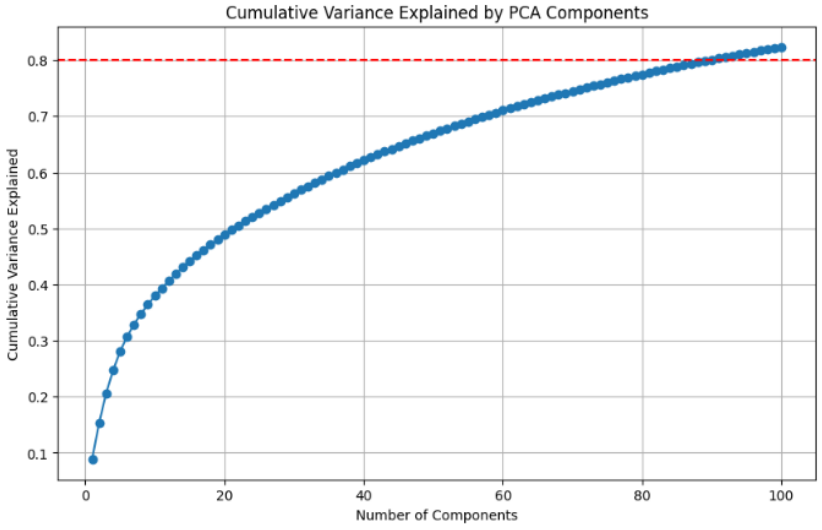
با انتخاب ترشولد برابر ۸۰درصد، هم مقدار خیلی خوبی از واریانس را پوشش میدهیم و همچنین ابعاد دیتا بشدت کاهش می‌یابد. مثلا اگر ترشولد را ۹۰ انتخاب میکردیم باید n_component حدود ۱۰۰ انتخاب میکردیم که دو و نیم برابر مقدار فعلی است.




```
Number of components to explain at least 80% variance: 43
```

hog + Sobel ۲.(۲

در این قسمت هم مثل قسمت قبل عمل میکنیم ولی چون بجای تصویر روی فیچر وکتور کار میکنیم، امکان نمایش centered را نداریم. در این جا توانستیم با ۸۰درصد واریانس، ابعاد را از ۲۰۰۰ به ۹۰ کاهش بدهیم که بسیار مقدار قابل توجهی است.




```
Number of components to explain at least 80% variance: 90
```

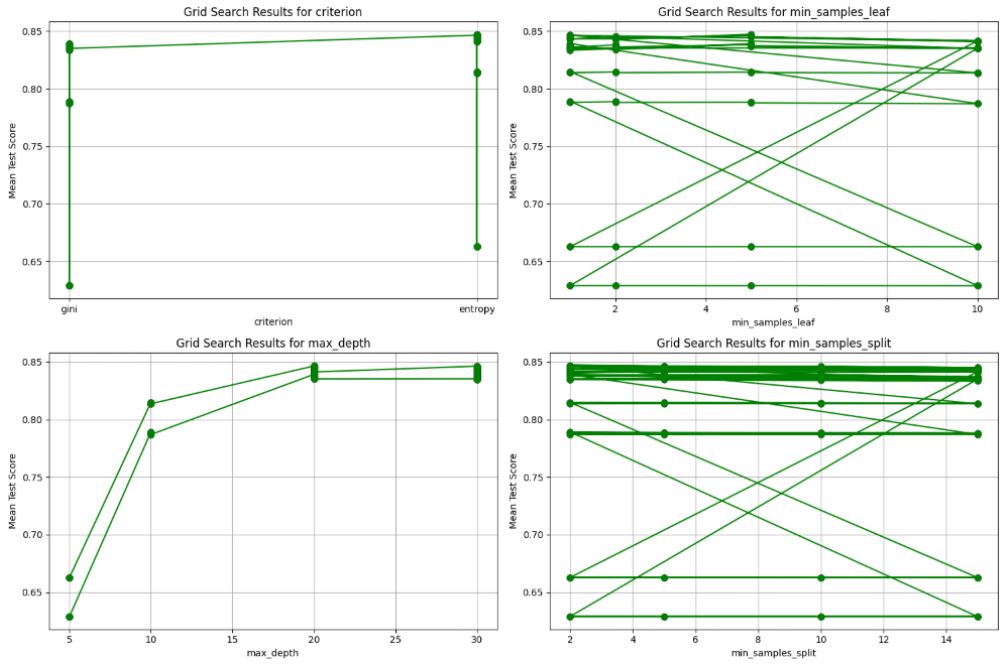
۳) فاز ۳

در این فاز به ساختن مدل های درخت تصمیم و SVM می‌پردازیم که دیتای train آنها با توجه به صورت پروژه، به صورت زیر دسته بنده شده است.

۱.(۳ Decision Tree

در این قسمت، برای هر دیتا گریدسرچ را اجرا میکنیم تا بهترین پارامتر هارا پیدا کنیم و به ازای هر نوع دیتا مقدار max_depth را برای هر گریدسرچ شخصی سازی میکنیم. همچنین برای cross validation ، پارامتر CV را برابر ۵ قرار میدهیم که هنگام گریدسرچ اجرا شود

۱.۱.(۳ Raw




```
param_grid = {
    'criterion': ['gini', 'entropy'],
    'min_samples_leaf': [1, 2, 5, 10],
    'max_depth': [None,5, 10, 20, 30],
    'min_samples_split': [2, 5, 10, 15]
}
```

شکل ۲: پارامتر های گرید سرچ

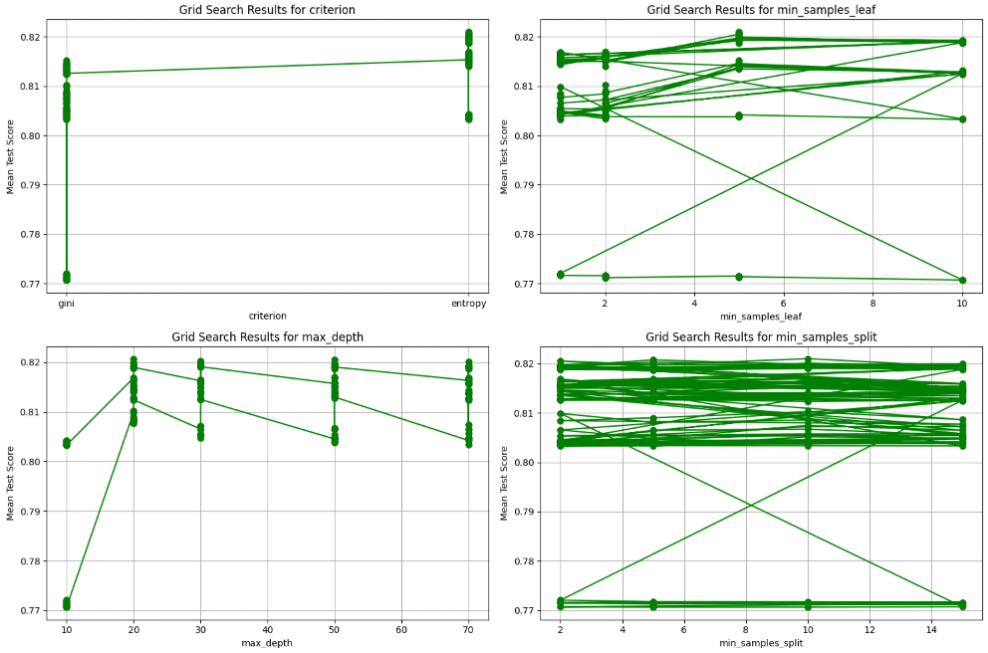
بهترین پارامتر ها به صورت زیر می‌باشند

- **criterion :** entropy
- **min_samples_leaf :** ۵
- **max_depth :** ۲۰
- **min_samples_split :** ۲

دقت مدل: ۸۵ درصد

با توجه به نمودار ها، به صورت کلی آنتروپی دقت بالاتری نسبت به gini داشته است. همچنین با افزایش عمق تا ۲۰، افزایش دقت خوبی داشتیم ولی بعد آن دقت ثابت شد. همچنین میتوان نتیجه گرفت در عمق کمتر از ۲۰ مشکل underfit شدن را داشته ایم.

۲.۱.(۳ hog + Sobel



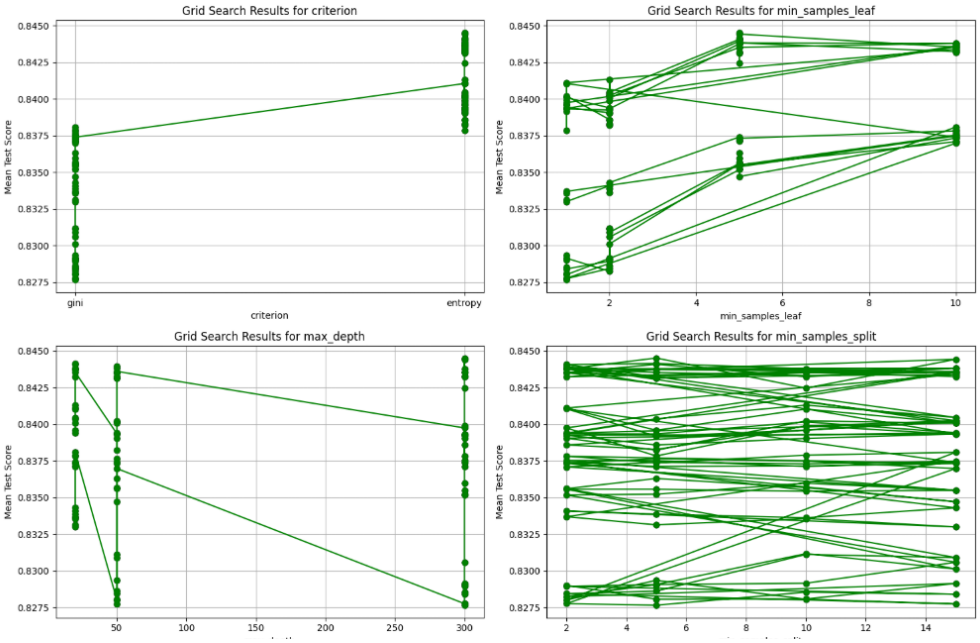

```
param_grid_2 = {
    'criterion': ['gini', 'entropy'],
    'min_samples_leaf': [1, 2, 5, 10],
    'max_depth': [None, 10, 20, 30,50,70],
    'min_samples_split': [2, 5, 10, 15]
}
```

بهترین پارامتر ها به صورت زیر می‌باشند

- **criterion :** entropy
- **min_samples_leaf :** ۵
- **max_depth:** None
- **min_samples_split:** ۱۰

دقت مدل: ۸۲ درصد با توجه به نمودار ها به صورت کلی، دقت در entropy بالاتر بوده. همچنین در minsamples برابر ۵ بیشترین دقت را به ما داده است.

۳.۱.(۳ Sobel




```
param_grid_3 = {
    'criterion': ['gini', 'entropy'],
    'min_samples_leaf': [1, 2, 5, 10],
    'max_depth': [None, 20, 50,300],
    'min_samples_split': [2, 5, 10, 15]
}
```

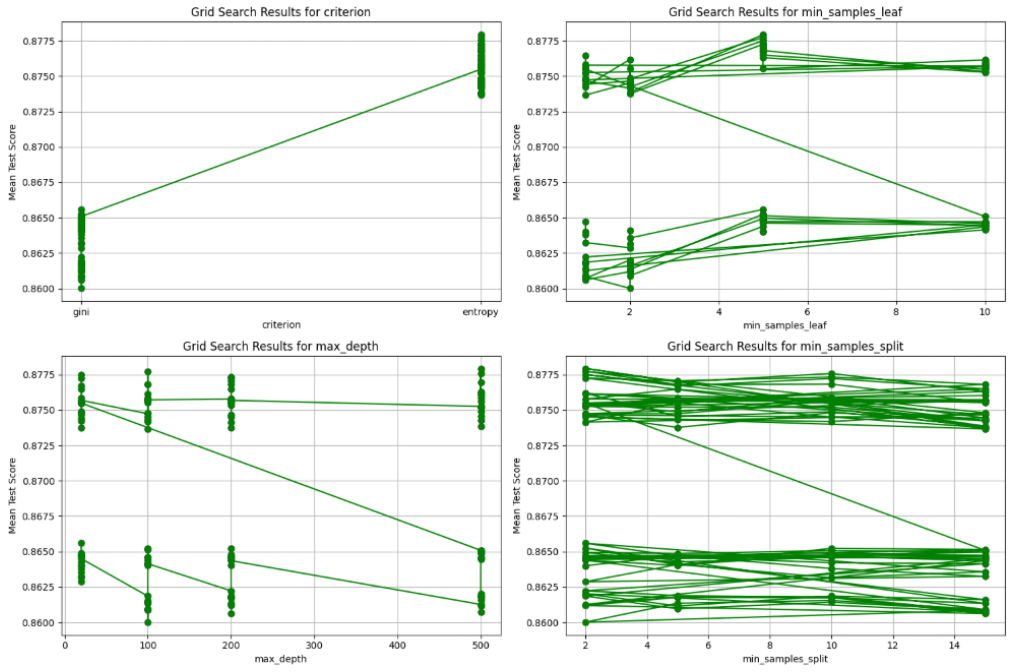
بهترین پارامتر ها به صورت زیر می‌باشند

- **criterion :** entropy
- **min_samples_leaf :** ۵
- **max_depth:** ۳۰۰
- **min_samples_split:** ۵

دقت مدل: ۸۴ درصد

با توجه به نمودار ها، مانند نمونه های قبل، entropy دقت بالاتری داشته است همچنین در minsamples برابر ۵ دقت بالاتری داشتیم.همچنین با افزایش عمق دقت کمی بهتر شده. چون تعداد فیچر های این دیتا بسیار زیاد است (۲۰۰۰ تا) اگر عمق کم باشد باعث underfit شدن میشود

۴.۱.(۳ Laplacian + hog



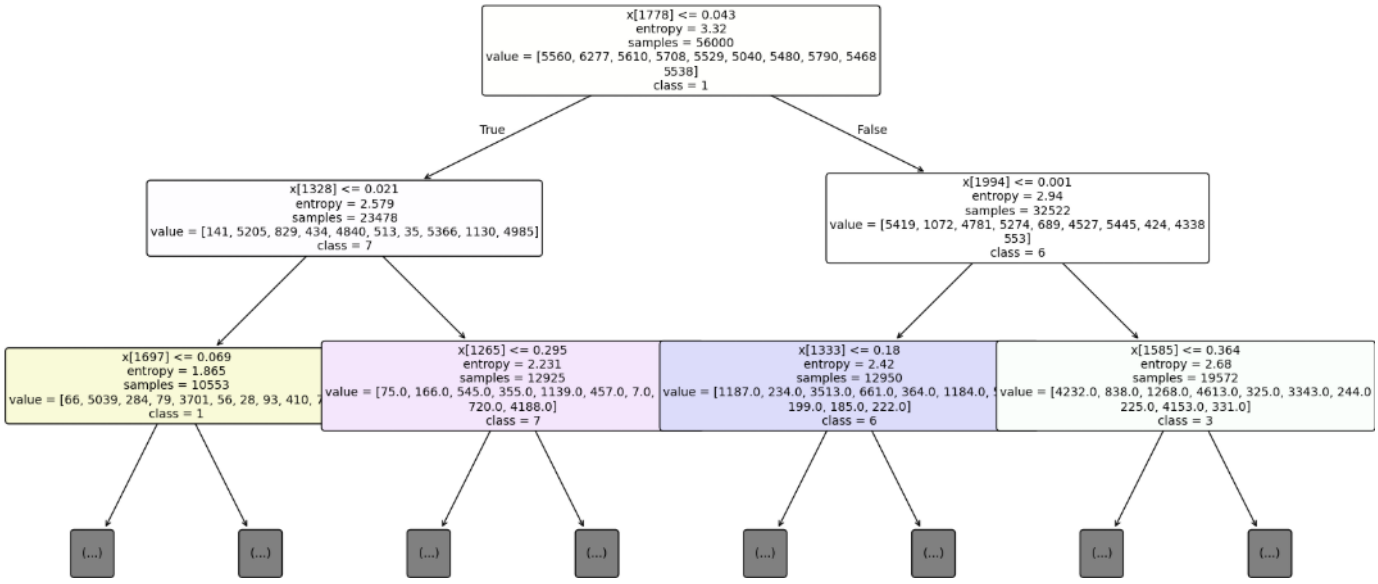

```
param_grid_4 = {
    'criterion': ['gini', 'entropy'],
    'min_samples_leaf': [1, 2, 5, 10],
    'max_depth': [20, 100,200,500],
    'min_samples_split': [2, 5, 10, 15]
}
```

بهترین پارامتر ها به صورت زیر می‌باشند

- **criterion :** entropy
- **min_samples_leaf :** ۵
- **max_depth :** ۵۰۰
- **min_samples_split :** ۲

دقت مدل: ۸۸ درصد با توجه به نمودار ها میتوان مشاهده کرد با افزایش min_samples ، دقت به صورت زیگزاگی بالا پایین میشود. با اینکه در min_samples ۲ دقت بالاتری داریم اما مین سمپلز ۵ انتخاب شده چون ممکن است در حالت های دیگر کراس ولیدیشن، دقت پایینی داشته. همچنین شیب تفاوت دقت در entropy و gini نیز قابل توجه است.

۵.۱.۳) نمایش بهترین درخت



با توجه به درخت، میتوان ۷ تا از فیچر ها که بیشترین تاثیرگذاری را در انتخاب لیبل دارند را مشاهده کرد که با جرکت از سمت ریشه به برگ تاثیر و ارزش آنها کمتر میشود. چون مقدار ویژگی های ما پیوسته هستند، در هر نود شرط تقسیم داده ها مشخص شده است. مثلا در ریشه اگر مقدار فیچر ۱۷۷۸ از ۰۴۳.۰ کمتر مساوی باشد به سمت چپ داده ها تقسیم میشوند و اگر بیشتر باشند، داده ها به سمت راست تقسیم میشوند.

همچنین میتوان مقدار داده هایی که در هر نود تقسیم میشوند را مشاهده کرد. به عنوان مثلا در ریشه که ما ۵۶۰۰۰ داده داریم، ۲۳۴۷۸ داده در سمت چپ قرار میگیرند و ۳۲۵۲۲ داده در سمت راست. یعنی ۳۲۵۲۲ داده مقدار فیچر ۱۷۷۸ آنها از ۰۴۳.۰ بیشتر بوده

همچنین چون ما برای انتخاب فیچر تقسیم کننده در هر نود از entropy استفاده کرده ایم، میتوان آنتروپی هر نود را مشاهده کرد و در نتیجه gain information را نیز محاسبه کرد.

۳).۱.۶ مقایسه و نتیجه گیری

با توجه به نتایج بدست آمده، پایین ترین دقت مربوط به sobel و بالاترین دقت مربوط به laplacian + hog است.
میتوان دلایل این تفاوت درصد هارو به صورت زیر تحلیل کرد.

- **raw :** پیکسل‌ها به صورت مستقیم الگوهای موجود در داده را نمایش می‌دهند. در دیتاست مرتب و ساختارمند مانند MNIST، درخت تصمیم می‌تواند به خوبی این الگوها را یاد بگیرد و دقت نسبتاً بالایی ارائه دهد.
- **sobel :** لتر سوبل لبه‌ها را با تشخیص گرادیان‌ها برجسته می‌کند که برای شناسایی شکل‌ها مفید است. با این حال، برخی جزئیات یا اطلاعات بافت ممکن است از دست برود و به همین دلیل دقت کمی کمتر از داده‌های خام شده است.
- **hog + sobel :**ترکیب ویژگی‌های لبه‌های سوبل و ویژگی‌های HOG پیچیدگی را افزایش می‌دهد. اگرچه HOG ویژگی‌های ساختاری خوبی ارائه می‌دهد، ترکیب آن با سوبل ممکن است اطلاعات تکراری یا متناقض ایجاد کند و توانایی مدل در تعمیم‌دهی را کاهش دهد.
- **laplacian + hog :** HOG ویژگی‌های ساختاری قوی ارائه می‌دهد و فیلتر لاپلاسین تغییرات سریع شدت (گرادیان‌های مرتبه دوم) را شناسایی می‌کند. این دو ویژگی مکمل یکدیگر هستند و ویژگی‌های غنی و متنوعی تولید می‌کنند که باعث بهبود عملکرد مدل و دستیابی به بالاترین دقت در میان تمام ترکیب‌ها می‌شود.

۲).۳ SVM

در این قسمت، برای مقایسه و تاثیر پارامتر های مختلف، ۵ مدل train کردیم. یک مدل با پارامتر های دیفالت و ۴ مدل با پارامتر های gamma و C متفاوت.

Raw ۱.۲.(۳

```
  warnings.warn(
[8 4 8 ... 2 7 1]

accuracy_score:  0.9804285714285714

classification_report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1343
           1       0.99      0.99      0.99      1600
           2       0.97      0.98      0.98      1380
           3       0.98      0.97      0.97      1433
           4       0.98      0.98      0.98      1295
           5       0.98      0.97      0.98      1273
           6       0.99      0.99      0.99      1396
           7       0.98      0.98      0.98      1503
           8       0.98      0.97      0.98      1357
           9       0.97      0.97      0.97      1420

    accuracy                           0.98     14000
   macro avg       0.98      0.98      0.98     14000
weighted avg       0.98      0.98      0.98     14000
```

```
[8 4 8 ... 2 7 1]

accuracy_score:  0.9728571428571429

classification_report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1343
           1       0.98      0.99      0.98      1600
           2       0.97      0.97      0.97      1380
           3       0.96      0.96      0.96      1433
           4       0.97      0.97      0.97      1295
           5       0.97      0.97      0.97      1273
           6       0.98      0.99      0.99      1396
           7       0.97      0.97      0.97      1503
           8       0.97      0.96      0.97      1357
           9       0.96      0.96      0.96      1420

    accuracy                           0.97     14000
   macro avg       0.97      0.97      0.97     14000
weighted avg       0.97      0.97      0.97     14000
```

```
[8 4 8 ... 2 7 1]

accuracy_score:  0.9803571428571428

classification_report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1343
           1       0.99      0.99      0.99      1600
           2       0.97      0.98      0.98      1380
           3       0.98      0.97      0.97      1433
           4       0.98      0.98      0.98      1295
           5       0.98      0.97      0.98      1273
           6       0.99      0.99      0.99      1396
           7       0.98      0.98      0.98      1503
           8       0.98      0.97      0.98      1357
           9       0.97      0.97      0.97      1420

    accuracy                           0.98     14000
   macro avg       0.98      0.98      0.98     14000
weighted avg       0.98      0.98      0.98     14000
```

شکل ۳: مدل با گاما ۰.۰۱ و auto

```
  warnings.warn(
[8 4 8 ... 2 7 1]

accuracy_score:  0.9642142857142857

classification_report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99      1343
           1       0.98      0.99      0.98      1600
           2       0.96      0.96      0.96      1380
           3       0.95      0.95      0.95      1433
           4       0.96      0.97      0.96      1295
           5       0.96      0.96      0.96      1273
           6       0.97      0.98      0.98      1396
           7       0.96      0.96      0.96      1503
           8       0.96      0.95      0.95      1357
           9       0.95      0.94      0.95      1420

    accuracy                           0.96     14000
   macro avg       0.96      0.96      0.96     14000
weighted avg       0.96      0.96      0.96     14000
```

```
  warnings.warn(
[8 4 8 ... 2 7 1]

accuracy_score:  0.9846428571428572

classification_report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1343
           1       0.99      0.99      0.99      1600
           2       0.97      0.99      0.98      1380
           3       0.98      0.98      0.98      1433
           4       0.98      0.99      0.98      1295
           5       0.99      0.98      0.98      1273
           6       0.99      0.99      0.99      1396
           7       0.98      0.99      0.98      1503
           8       0.98      0.98      0.98      1357
           9       0.98      0.97      0.98      1420

    accuracy                           0.98     14000
   macro avg       0.98      0.98      0.98     14000
weighted avg       0.98      0.98      0.98     14000
```

شکل ۴: مدل با C ۰.۱ و ۱۰

hog + Sobel ۲.۲.(۳

```
[8 4 8 ... 2 7 1]

accuracy_score:  0.9675714285714285

classification_report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      1343
           1       0.99      0.99      0.99      1600
           2       0.96      0.97      0.97      1380
           3       0.95      0.95      0.95      1433
           4       0.97      0.96      0.96      1295
           5       0.97      0.95      0.96      1273
           6       0.98      0.99      0.98      1396
           7       0.98      0.97      0.97      1503
           8       0.95      0.96      0.95      1357
           9       0.94      0.96      0.95      1420

    accuracy                           0.97     14000
   macro avg       0.97      0.97      0.97     14000
weighted avg       0.97      0.97      0.97     14000
```

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

accuracy_score:  0.11428571428571428

classification_report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1343
           1       0.11      1.00      0.21      1600
           2       0.00      0.00      0.00      1380
           3       0.00      0.00      0.00      1433
           4       0.00      0.00      0.00      1295
           5       0.00      0.00      0.00      1273
           6       0.00      0.00      0.00      1396
           7       0.00      0.00      0.00      1503
           8       0.00      0.00      0.00      1357
           9       0.00      0.00      0.00      1420

    accuracy                           0.11     14000
   macro avg       0.01      0.10      0.02     14000
weighted avg       0.01      0.11      0.02     14000
```

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

accuracy_score:  0.11428571428571428

classification_report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      1343
           1       0.11      1.00      0.21      1600
           2       0.00      0.00      0.00      1380
           3       0.00      0.00      0.00      1433
           4       0.00      0.00      0.00      1295
           5       0.00      0.00      0.00      1273
           6       0.00      0.00      0.00      1396
           7       0.00      0.00      0.00      1503
           8       0.00      0.00      0.00      1357
           9       0.00      0.00      0.00      1420

    accuracy                           0.11     14000
   macro avg       0.01      0.10      0.02     14000
weighted avg       0.01      0.11      0.02     14000
```

شکل ۵: مدل با گاما ۰۱. و auto

```
[8 4 8 ... 2 7 1]

accuracy_score:  0.9423571428571429

classification_report:
              precision    recall  f1-score   support

           0       0.96      0.97      0.97      1343
           1       0.97      0.98      0.98      1600
           2       0.94      0.96      0.95      1380
           3       0.91      0.91      0.91      1433
           4       0.93      0.94      0.93      1295
           5       0.95      0.90      0.93      1273
           6       0.97      0.97      0.97      1396
           7       0.96      0.93      0.95      1503
           8       0.92      0.92      0.92      1357
           9       0.90      0.94      0.92      1420

    accuracy                           0.94     14000
   macro avg       0.94      0.94      0.94     14000
weighted avg       0.94      0.94      0.94     14000
```

```
[8 4 8 ... 2 7 1]

accuracy_score:  0.9743571428571428

classification_report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      1343
           1       0.99      0.99      0.99      1600
           2       0.97      0.98      0.97      1380
           3       0.96      0.96      0.96      1433
           4       0.97      0.98      0.98      1295
           5       0.98      0.96      0.97      1273
           6       0.99      0.98      0.99      1396
           7       0.98      0.98      0.98      1503
           8       0.96      0.96      0.96      1357
           9       0.96      0.97      0.96      1420

    accuracy                           0.97     14000
   macro avg       0.97      0.97      0.97     14000
weighted avg       0.97      0.97      0.97     14000
```

شکل ۶: مدل با C ۰.۱ و ۱۰

میتوان مشاهده کرد که دقت مدل در گاما های مختلف بسیار پایین است.پارامتر gamma در SVM تعیین می‌کند که تاثیر هر نمونه آموزشی تا چه فاصله‌ای گسترش پیدا کند:.

در این دو مورد میتوان احتمال داد که چون گاما بسیار کوچک است، مدل فقط مناطق بسیار گسترده‌ای را در نظر بگیرد و مرز تصمیم ساده‌ای ایجاد کند که نتواند پیچیدگی داده را به خوبی مدل کند. یعنی مدل ممکن است همه داده‌ها را به کلاس غالب (مثل کلاس ۱) اختصاص دهد، و مدل احتمال زیاد underfit شده است.

همچنین هم در این قسمت هم در قسمت دیتا ،raw میتوان دید که دقت مدل با C برابر ۱۰ بیشتر از ۰.۱ است که میتوان به صورت زیر تحلیلش کرد.

- **C=۰.۰۱:** مقدار پایین C تاکید بیشتری بر بیشینه کردن حاشیه بین کلاس‌ها دارد، حتی اگر برخی نمونه‌های آموزشی به درستی طبقه‌بندی نشوند. این منجر به یک حاشیه بزرگ‌تر می‌شود که ممکن است برای داده‌های نویزی یا دارای هم‌پوشانی بالا مفید باشد. اما برای ،MNIST که داده‌ها به خوبی در فضای ویژگی‌ها جدا شده‌اند، ممکن است مدل دچار underfit شود و دقت کاهش یابد.
- **C=۱۰ :** مقدار بالای C تاکید بیشتری بر کاهش خطاهای طبقه‌بندی دارد، به این معنی که مدل تلاش بیشتری می‌کند تا تمام نمونه‌های آموزشی را به درستی طبقه‌بندی کند. این باعث ایجاد حاشیه کوچک‌تر می‌شود که به داده‌ها نزدیک‌تر است. برای دیتاست‌هایی مانند ،MNIST که داده‌ها تمیز و ساختارمند هستند، این تطابق دقیق‌تر معمولاً منجر به دقت بهتر در داده‌های آموزشی و تست می‌شود.

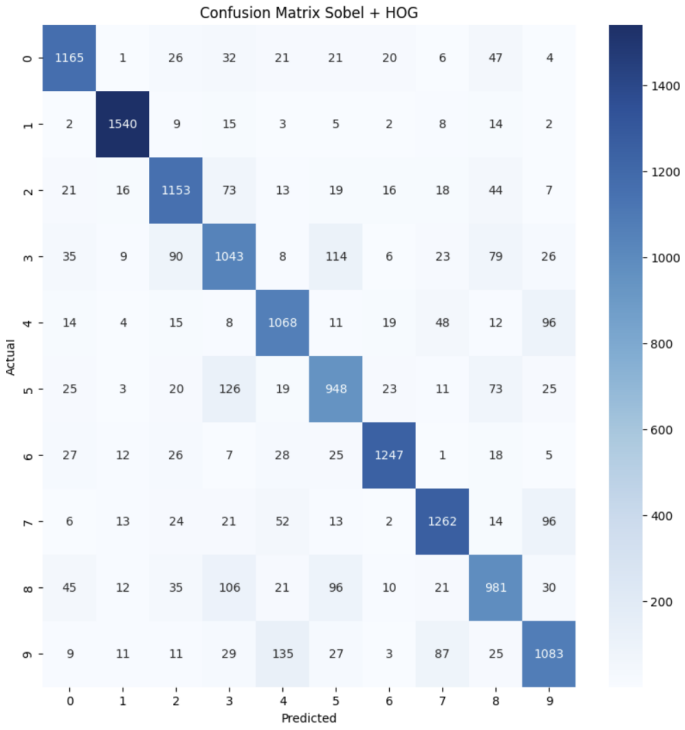
۳).۳ مقایسه درخت تصمیم و SVM

با توجه به نتایج بدست آمده، به وضوح میتوان اختلاف قابل توجه این دو مدل را مشاهده کرد. از دلایل دقت بالاتر SVM میتوان به موارد زیر اشاره کرد.

- **مدیریت داده‌های با ابعاد بالا:** SVM برای داده‌های با ابعاد بالا (مانند ویژگی‌های HOG و سوبل) بسیار موثر است، زیرا هدف آن بیشینه کردن حاشیه بین کلاس‌ها است. در مقابل، درخت تصمیم ممکن است با پیچیدگی این ویژگی‌ها دچار overfit شود و به خوبی تعمیم ندهد.
- **مرزهای تصمیم‌گیری:** SVM از hyperplane برای جداسازی کلاس‌ها استفاده می‌کند که می‌توانند روابط پیچیده بین داده‌ها را بهتر مدل کنند. اما درخت تصمیم فقط با تقسیم‌بندی‌های axis-aligned داده‌ها را جدا می‌کند که انعطاف‌پذیری کمتری دارد.

۴) فاز ۴

DT ۱.(۴




```
predict array:  [8 4 5 ... 2 7 1]

accuracy_score:  0.8207142857142857

classification_report:
              precision    recall  f1-score   support

           0       0.86      0.87      0.87      1343
           1       0.95      0.96      0.96      1600
           2       0.82      0.84      0.83      1380
           3       0.71      0.73      0.72      1433
           4       0.78      0.82      0.80      1295
           5       0.74      0.74      0.74      1273
           6       0.93      0.89      0.91      1396
           7       0.85      0.84      0.84      1503
           8       0.75      0.72      0.74      1357
           9       0.79      0.76      0.78      1420

    accuracy                           0.82     14000
   macro avg       0.82      0.82      0.82     14000
weighted avg       0.82      0.82      0.82     14000
```

: : Report Classification

- **Precision :**درصد نمونه‌های واقعی یک کلاس که به درستی شناسایی شده‌اند.
- **Recall :**درصد نمونه‌های واقعی یک کلاس که به درستی شناسایی شده‌اند.
- **F1-Score :**میانگین هارمونیک بین دقت و بازیابی.

کلاس ۱ : با دقت ۹۵.۰ و F1-Score ۹۶.۰، بهترین عملکرد را دارد

کلاس‌های ۳، ۴ و ۹: : عملکرد ضعیف‌تری نسبت به سایر کلاس‌ها دارند (دقت و F1-Score در حدود ۷۱.۰ تا ۷۹.۰).

کلاس ۶ : :عملکرد نسبتاً خوبی دارد با دقت ۹۳.۰ و F1-Score ۹۱.۰.

کلاس ۰ : دقت ۸۶.۰، بازیابی ۸۷.۰، و F1-Score ۸۷.۰. عملکرد مدل در این کلاس خوب است.

با توجه به مفهموم confusion matrix میتوانیم نتیجه گیری زیر را داشته باشیم.

کلاس‌های با دقت پایین :

- **کلاس ۸ :** تعداد پیش‌بینی‌های درست برای این کلاس ۹۸۲ است، اما خطاهایی نسبتاً زیادی دارد.

۹۲ نمونه به کلاس ۵ پیش‌بینی شده‌اند

- **کلاس ۹:** دقت برای این کلاس نسبتاً خوب است (۱۰۸۸ پیش‌بینی درست)، اما نمونه‌های زیادی (۱۳۳) به‌اشتباه به کلاس ۵ اختصاص داده شده‌اند.

کلاس‌های با دقت بالا:

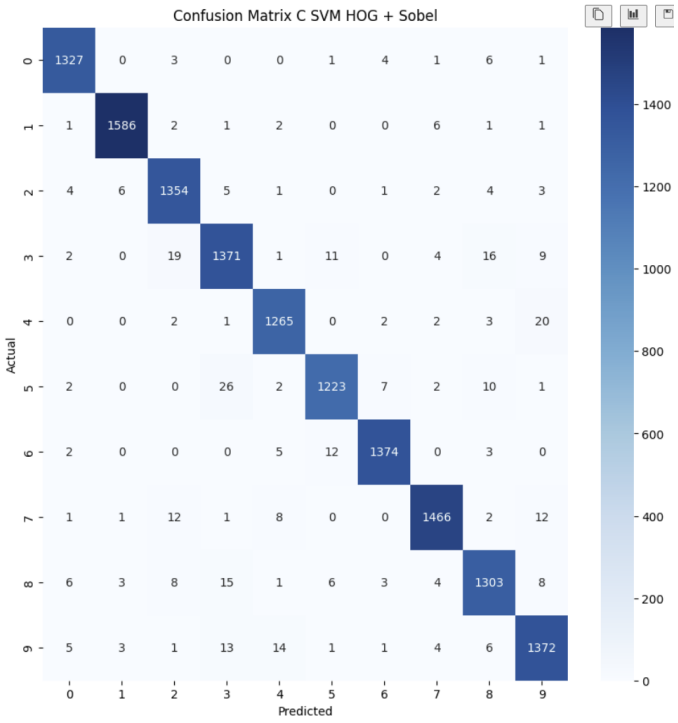
- **کلاس ۱ :** مدل ۱۵۴۰ نمونه را به‌درستی پیش‌بینی کرده و تعداد خطاهای آن بسیار کم است. این نشان‌دهنده دقت بالای مدل در شناسایی این کلاس است.

تحلیل کلی :

مدل عملکرد خوبی در شناسایی اکثر کلاس‌ها داشته است.ما برخی از کلاس ها (مانند ۳، ۸ و ۹) با سایر کلاس ها اشتباه گرفته شده اند که ممکن است به دلایل زیر باشد:

- شباهت زیاد ویژگی‌های این کلاس‌ها.
- عدم تعادل در تعداد نمونه‌ها بین کلاس‌ها.
- کمبود داده‌های آموزشی کافی برای این کلاس‌ها.

SVM ۲.(۴




```
[8 4 8 ... 2 7 1]

accuracy_score:  0.9743571428571428

classification_report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.99      1343
           1       0.99      0.99      0.99      1600
           2       0.97      0.98      0.97      1380
           3       0.96      0.96      0.96      1433
           4       0.97      0.98      0.98      1295
           5       0.98      0.96      0.97      1273
           6       0.99      0.98      0.99      1396
           7       0.98      0.98      0.98      1503
           8       0.96      0.96      0.96      1357
           9       0.96      0.97      0.96      1420

    accuracy                           0.97     14000
   macro avg       0.97      0.97      0.97     14000
weighted avg       0.97      0.97      0.97     14000
```

Report Classification : :

هر کلاس (از ۰ تا ۹) عملکرد بسیار خوبی داشته و مقادیر دقت، بازیابی و F۱-Score بالای ۹۶٪ هستند.

تحلیل ماتریس گمراهی :

- **عملکرد قوی مدل:** مدل بیشتر نمونه‌ها را به درستی پیش‌بینی کرده است (اعداد در خانه‌های قطری بزرگترند).
- **خطاهای پراکنده:** اشتباهات بین ارقام نزدیک‌تر مانند ۸ و ۹ یا ۴ و ۷ بیشتر است.
- عدد ۱ با دقت بسیار بالا پیش‌بینی شده است (۱۵۸۶ بار درست).
- ارقام ۳ و ۸ نسبت به بقیه کمی خطای بیشتری داشتند.
- این ماتریس نشان می‌دهد که مدل عملکرد خوبی دارد، اما هنوز خطاهای کمی در طبقه‌بندی اعداد وجود دارد که ممکن است ناشی از شباهت ظاهری ارقام یا محدودیت ویژگی‌ها باشد.

۵) فاز ۵

	max_depth	min_samples_split	min_samples_leaf	train_accuracy	test_accuracy
48	nan	2	1	1.000000	0.821643
32	15.000000	2	1	0.987071	0.824143
52	nan	5	1	0.983518	0.821643
36	15.000000	5	1	0.974161	0.824786
60	nan	10	1	0.961911	0.819929
44	15.000000	10	1	0.955429	0.821857
63	nan	10	5	0.940661	0.823286
59	nan	20	5	0.917357	0.824714
61	nan	10	10	0.904804	0.828643
57	nan	20	10	0.904804	0.828643
41	15.000000	20	10	0.903143	0.828286
33	15.000000	2	10	0.903143	0.828286
45	15.000000	10	10	0.903143	0.828286
37	15.000000	5	10	0.903143	0.828286
62	nan	10	20	0.869661	0.822929
58	nan	20	20	0.869661	0.822929
54	nan	5	20	0.869661	0.822929
50	nan	2	20	0.869661	0.822929
42	15.000000	20	20	0.869429	0.822929
46	15.000000	10	20	0.869429	0.822929

برای اینکه یک درخت را overfit کنیم، چندین راه حل داریم. افزایش عمق، کاهش min_samples برای برگ و تقسیم. چون بهترین مدل گرید سرچ عمق برابر None دارد، بیشتر باید با استفاده از پارامتر های حداقل نمونه درخت را overfit کنیم.

طبق نتیجه بدست آمده از پارامتر های مختلف، ، با استفاده از کمترین حداقل نمونه برای تقسیم و برگ، و همچنین گذاشتن عمق برابر None درخت به طور کامل بر روی دیتا ،overfit train شده است و دقت ۱۰۰ درصد داریم و احتلاف ۱۸ درصدی بین دیتا تمرین و تست وجود دارد.

با محدود کردن هایپرپارامتر ها، به عنوان مثال مشخص کردن حداکثر عمق و افزایش حداقل نمونه ها، میتوان در سطر

های زرد شده مشاهده کرد که اختلاف درصد بین داده های تست و تمرین به آرامی کاهش می‌یابد تا زمانی که به سطر سبز که بهترین پارامتر ها است میرسیم. البته با بیشتر محدود کردن پارامتر ها میتوانیم به اختلاف درصد ۴ درصدی بین پارامتر های تست و تمرین برسیم.