

```
In [ ]: from sklearn.ensemble import RandomForestClassifier  
import pandas as pd  
import matplotlib.pyplot as plt
```

```
In [ ]: X = pd.read_table('D:\\PythonS\\DataSets\\FSU Applied Machine Learning DS\\MADELON\\madelon_train.txt')  
Y = pd.read_table('D:\\PythonS\\DataSets\\FSU Applied Machine Learning DS\\MADELON\\madelon_target.txt')  
Xtest = pd.read_table('D:\\PythonS\\DataSets\\FSU Applied Machine Learning DS\\MADELON\\madelon_test.txt')  
Ytest = pd.read_table('D:\\PythonS\\DataSets\\FSU Applied Machine Learning DS\\MADELON\\madelon_labels.txt')
```

```
In [ ]: X.isnull().value_counts()
```

```
Out[ ]:  
0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  
2   13  14  15  16  17  18  19  20  21  22  23  24  
25  26  27  28  29  30  31  32  33  34  35  36  37  
7   38  39  40  41  42  43  44  45  46  47  48  49  
50  51  52  53  54  55  56  57  58  59  60  61  62  
2   63  64  65  66  67  68  69  70  71  72  73  74  
75  76  77  78  79  80  81  82  83  84  85  86  87  
7   88  89  90  91  92  93  94  95  96  97  98  99  
100 101 102 103 104 105 106 107 108 109 110 111 112  
12  113 114 115 116 117 118 119 120 121 122 123 124  
4   125 126 127 128 129 130 131 132 133 134 135 136  
137 138 139 140 141 142 143 144 145 146 147 148 149  
49  150 151 152 153 154 155 156 157 158 159 160 161  
1   162 163 164 165 166 167 168 169 170 171 172 173  
174 175 176 177 178 179 180 181 182 183 184 185 186  
86  187 188 189 190 191 192 193 194 195 196 197 198  
8   199 200 201 202 203 204 205 206 207 208 209 210  
211 212 213 214 215 216 217 218 219 220 221 222 223  
23  224 225 226 227 228 229 230 231 232 233 234 235  
5   236 237 238 239 240 241 242 243 244 245 246 247  
248 249 250 251 252 253 254 255 256 257 258 259 260  
60  261 262 263 264 265 266 267 268 269 270 271 272  
2   273 274 275 276 277 278 279 280 281 282 283 284  
285 286 287 288 289 290 291 292 293 294 295 296 297  
97  298 299 300 301 302 303 304 305 306 307 308 309  
9   310 311 312 313 314 315 316 317 318 319 320 321  
322 323 324 325 326 327 328 329 330 331 332 333 334  
34  335 336 337 338 339 340 341 342 343 344 345 346  
6   347 348 349 350 351 352 353 354 355 356 357 358  
359 360 361 362 363 364 365 366 367 368 369 370 371  
71  372 373 374 375 376 377 378 379 380 381 382 383  
3   384 385 386 387 388 389 390 391 392 393 394 395  
396 397 398 399 400 401 402 403 404 405 406 407 408  
08  409 410 411 412 413 414 415 416 417 418 419 420  
0   421 422 423 424 425 426 427 428 429 430 431 432  
433 434 435 436 437 438 439 440 441 442 443 444 445  
45  446 447 448 449 450 451 452 453 454 455 456 457  
7   458 459 460 461 462 463 464 465 466 467 468 469  
470 471 472 473 474 475 476 477 478 479 480 481 482  
82  483 484 485 486 487 488 489 490 491 492 493 494  
4   495 496 497 498 499 500  
False F  
alse False Fa  
lse False Fal  
se False Fal  
se False Fals  
e False  
False False False False False False False False False False False F  
alse False Fa  
lse False Fal  
se False Fals  
e False  
False False False False False False False False False False False F  
alse False Fa  
lse False Fal  
se False Fals  
e False  
False False False False False False False False False False False F  
alse False Fa  
lse False Fal  
se False Fals
```

```
In [ ]: X = X.drop(500, axis = 1)
```

```
In [ ]: rfc = RandomForestClassifier(random_state=0, max_features= 22 )
rfc = rfc.fit(X,Y)
```

```
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning:  
A column-vector y was passed when a 1d array was expected. Please change the shape  
of y to (n_samples,), for example using ravel().  
    return fit method(estimator, *args, **kwargs)
```

```
In [ ]: rfc.score(X,Y)
```

Out[1]: 1.0

```
In [ ]: Xtest.isnull().value_counts()
```

```
Out[ ]:  
0   1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  
2   13  14  15  16  17  18  19  20  21  22  23  24  
25  26  27  28  29  30  31  32  33  34  35  36  37  
7   38  39  40  41  42  43  44  45  46  47  48  49  
50  51  52  53  54  55  56  57  58  59  60  61  62  
2   63  64  65  66  67  68  69  70  71  72  73  74  
75  76  77  78  79  80  81  82  83  84  85  86  87  
7   88  89  90  91  92  93  94  95  96  97  98  99  
100 101 102 103 104 105 106 107 108 109 110 111 112  
12  113 114 115 116 117 118 119 120 121 122 123 124  
4   125 126 127 128 129 130 131 132 133 134 135 136  
137 138 139 140 141 142 143 144 145 146 147 148 149  
49  150 151 152 153 154 155 156 157 158 159 160 161  
1   162 163 164 165 166 167 168 169 170 171 172 173  
174 175 176 177 178 179 180 181 182 183 184 185 186  
86  187 188 189 190 191 192 193 194 195 196 197 198  
8   199 200 201 202 203 204 205 206 207 208 209 210  
211 212 213 214 215 216 217 218 219 220 221 222 223  
23  224 225 226 227 228 229 230 231 232 233 234 235  
5   236 237 238 239 240 241 242 243 244 245 246 247  
248 249 250 251 252 253 254 255 256 257 258 259 260  
60  261 262 263 264 265 266 267 268 269 270 271 272  
2   273 274 275 276 277 278 279 280 281 282 283 284  
285 286 287 288 289 290 291 292 293 294 295 296 297  
97  298 299 300 301 302 303 304 305 306 307 308 309  
9   310 311 312 313 314 315 316 317 318 319 320 321  
322 323 324 325 326 327 328 329 330 331 332 333 334  
34  335 336 337 338 339 340 341 342 343 344 345 346  
6   347 348 349 350 351 352 353 354 355 356 357 358  
359 360 361 362 363 364 365 366 367 368 369 370 371  
71  372 373 374 375 376 377 378 379 380 381 382 383  
3   384 385 386 387 388 389 390 391 392 393 394 395  
396 397 398 399 400 401 402 403 404 405 406 407 408  
08  409 410 411 412 413 414 415 416 417 418 419 420  
0   421 422 423 424 425 426 427 428 429 430 431 432  
433 434 435 436 437 438 439 440 441 442 443 444 445  
45  446 447 448 449 450 451 452 453 454 455 456 457  
7   458 459 460 461 462 463 464 465 466 467 468 469  
470 471 472 473 474 475 476 477 478 479 480 481 482  
82  483 484 485 486 487 488 489 490 491 492 493 494  
4   495 496 497 498 499 500  
False F  
alse False Fa  
lse False Fal  
se False Fal  
se False Fals  
e False  
False False False False False False False False False False False F  
alse False Fa  
lse False Fal  
se False Fals  
e False  
False False False False False False False False False False False F  
alse False Fa  
lse False Fal  
se False Fals  
e False  
False False False False False False False False False False False F  
alse False Fa  
lse False Fal  
se False Fals
```

```
In [ ]: Xtest = Xtest.drop(500, axis = 1)
```

```
In [ ]: rfc.score(Xtest, Ytest)
```

Out[]: 0.73

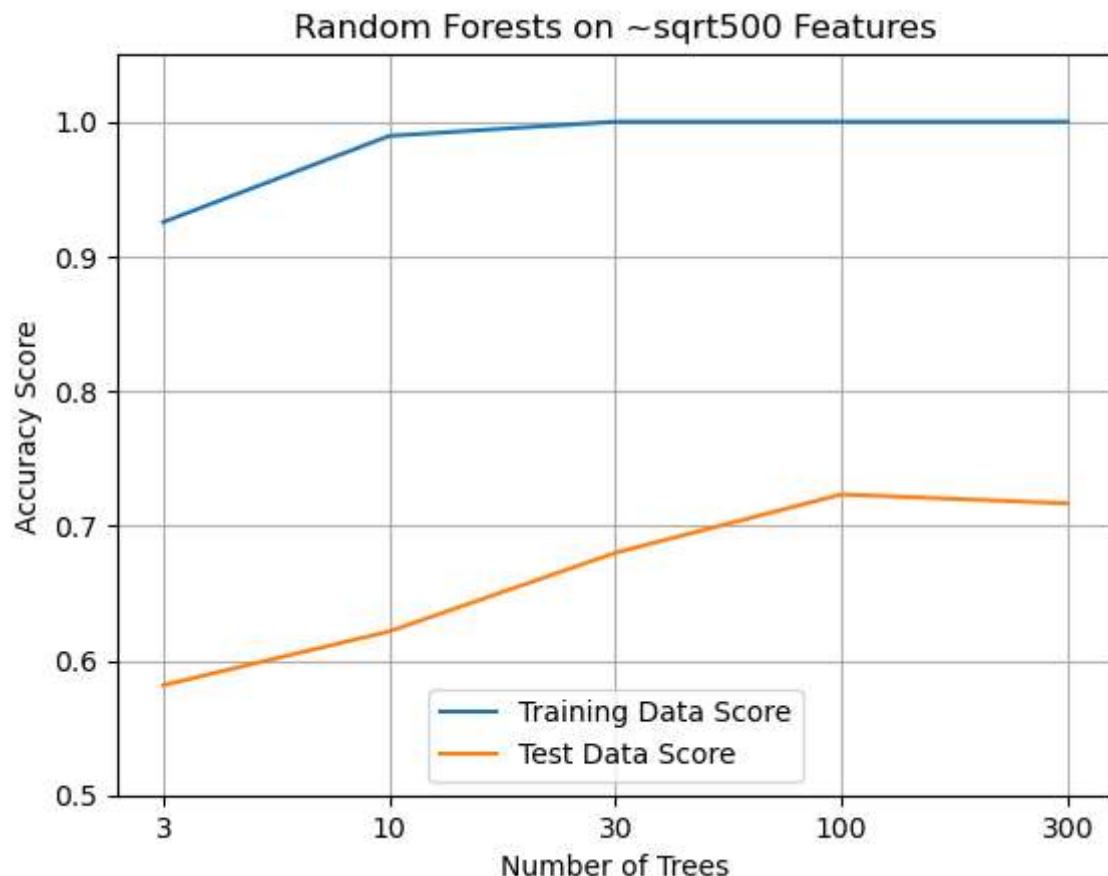
```
In [ ]: ntrees = [3,10,30,100,300]
        forests_sqrt = []
        for n in ntrees:
            forests_sqrt.append(Ran
```

```
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning
g: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning
g: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning
g: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning
g: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning
g: A column-vector y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
```

```
In [ ]: train_scores = []
        test_scores = []
```

```
for f in forests_sqrt:  
    train_scores.append(f.score(X,Y))  
    test_scores.append(f.score(Xtest,Ytest))
```

```
In [ ]: fig, ax = plt.subplots()  
plt.plot([1,2,3,4,5], train_scores, label = 'Training Data Score')  
plt.plot([1,2,3,4,5], test_scores, label = 'Test Data Score')  
plt.xlabel('Number of Trees')  
plt.ylabel('Accuracy Score')  
plt.legend(loc = 'lower center')  
plt.ylim(0.5,1.05)  
plt.yticks()  
ax.set_xticks([1,2,3,4,5])  
ax.set_xticklabels(['3', '10', '30', '100', '300' ])  
plt.grid()  
plt.title('Random Forests on ~sqrt500 Features')  
plt.show()
```



```
In [ ]: for t in train_scores:  
    print(1-t)
```

```
0.07450000000000001  
0.010499999999999954  
0.0  
0.0  
0.0
```

```
In [ ]: for t in test_scores:  
    print(1-t)
```

```
0.4183333333333333  
0.3783333333333333  
0.31999999999999995  
0.27666666666666666  
0.2833333333333333
```

```
In [ ]: forests_ln = []  
for n in ntrees:  
    forests_ln.append(RandomForestClassifier(n_estimators=n, random_state=n, max_featu  
  
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning:  
g: A column-vector y was passed when a 1d array was expected. Please change the shape  
of y to (n_samples,), for example using ravel().  
    return fit_method(estimator, *args, **kwargs)  
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning:  
g: A column-vector y was passed when a 1d array was expected. Please change the shape  
of y to (n_samples,), for example using ravel().  
    return fit_method(estimator, *args, **kwargs)  
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning:  
g: A column-vector y was passed when a 1d array was expected. Please change the shape  
of y to (n_samples,), for example using ravel().  
    return fit_method(estimator, *args, **kwargs)  
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning:  
g: A column-vector y was passed when a 1d array was expected. Please change the shape  
of y to (n_samples,), for example using ravel().  
    return fit_method(estimator, *args, **kwargs)  
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning:  
g: A column-vector y was passed when a 1d array was expected. Please change the shape  
of y to (n_samples,), for example using ravel().  
    return fit_method(estimator, *args, **kwargs)
```

```
In [ ]: train_scores_ln = []  
test_scores_ln = []  
for f in forests_ln:  
    train_scores_ln.append(f.score(X,Y))  
    test_scores_ln.append(f.score(Xtest,Ytest))
```

```
In [ ]: fig, ax = plt.subplots()  
plt.plot([1,2,3,4,5], train_scores_ln, label = 'Training Data Score')  
plt.plot([1,2,3,4,5], test_scores_ln, label = 'Test Data Score')  
plt.xlabel('Number of Trees')  
plt.ylabel('Accuracy Score')  
plt.legend(loc = 'center')  
plt.ylim(0.5,1.05)  
plt.yticks()  
ax.set_xticks([1,2,3,4,5])  
ax.set_xticklabels(['3', '10', '30', '100', '300' ])  
plt.grid()  
plt.title('Random Forests on ~ln500 Features')  
plt.show()
```

Random Forests on ~ln500 Features



```
In [ ]: for t in train_scores_ln:  
    print(1-t)
```

```
0.08150000000000002  
0.00749999999999951  
0.0  
0.0  
0.0
```

```
In [ ]: for t in test_scores_ln:  
    print(1-t)
```

```
0.4649999999999997  
0.4599999999999996  
0.4283333333333334  
0.3866666666666667  
0.3566666666666667
```

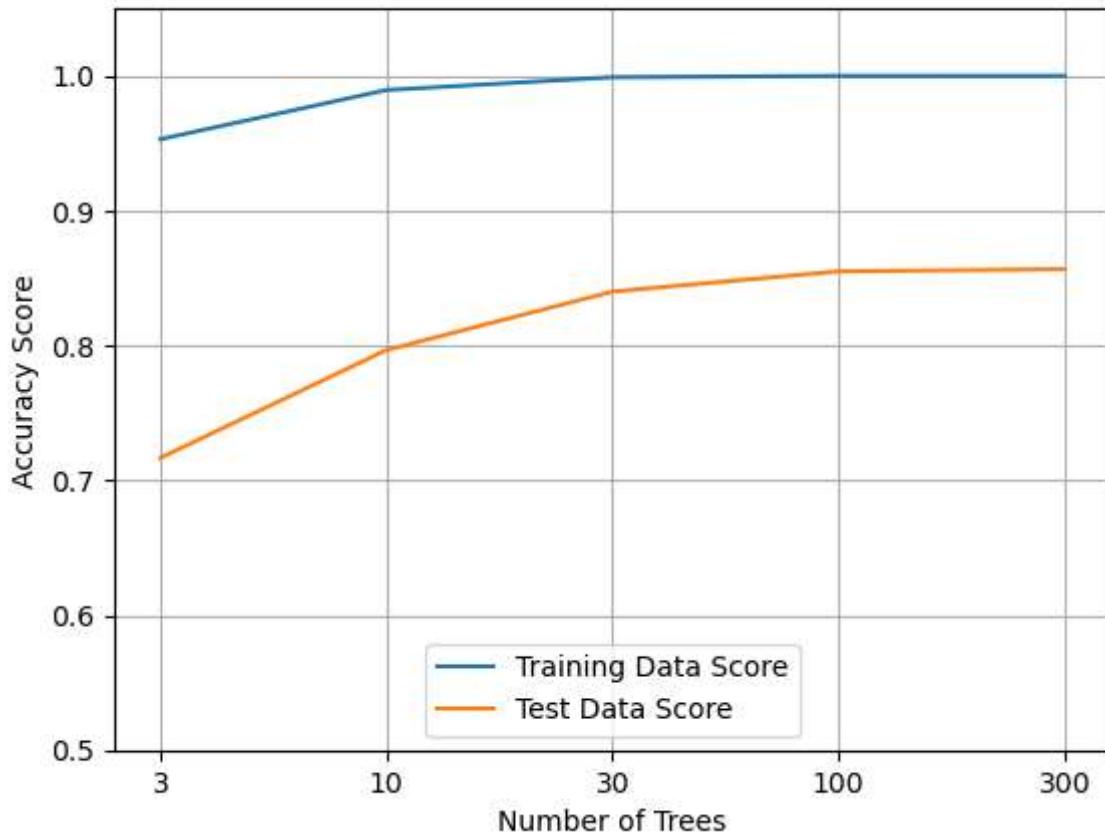
```
In [ ]: #takes about 4m 20s to run  
forests_500 = []  
for n in ntrees:  
    forests_500.append(RandomForestClassifier(n_estimators=n, random_state=n, max_feat
```

```
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
c:\Users\evans\anaconda3\Lib\site-packages\sklearn\base.py:1151: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
    return fit_method(estimator, *args, **kwargs)
```

```
In [ ]: train_scores_500 = []
test_scores_500 = []
for f in forests_500:
    train_scores_500.append(f.score(X,Y))
    test_scores_500.append(f.score(Xtest,Ytest))
```

```
In [ ]: fig, ax = plt.subplots()
plt.plot([1,2,3,4,5], train_scores_500, label = 'Training Data Score')
plt.plot([1,2,3,4,5], test_scores_500, label = 'Test Data Score')
plt.xlabel('Number of Trees')
plt.ylabel('Accuracy Score')
plt.legend(loc = 'lower center')
plt.ylim(0.5,1.05)
plt.yticks()
ax.set_xticks([1,2,3,4,5])
ax.set_xticklabels(['3', '10', '30', '100', '300' ])
plt.grid()
plt.title('Random Forests on 500 Features')
plt.show()
```

Random Forests on 500 Features



```
In [ ]: for t in train_scores_500:  
    print(1-t)
```

```
0.04700000000000004  
0.010499999999999954  
0.001000000000000009  
0.0  
0.0
```

```
In [ ]: for t in test_scores_500:  
    print(1-t)
```

```
0.2833333333333333  
0.2033333333333337  
0.1600000000000003  
0.1450000000000002  
0.1433333333333333
```