6rckikose

January 30, 2025

```
[10]: import numpy as np
      import pandas as pd
      data=pd.read_csv(r'/content/hypothyroid.csv')
      data.head()
```

```
[10]:   age sex on thyroxine query on thyroxine on antithyroid medication sick  \
      0  41   F            f                  f                         f    f
      1  23   F            f                  f                         f    f
      2  46   M            f                  f                         f    f
      3  70   F            t                  f                         f    f
      4  70   F            f                  f                         f    f

        pregnant thyroid surgery I131 treatment query hypothyroid  … TT4 measured  \
      0        f               f              f                 f  …            t
      1        f               f              f                 f  …            t
      2        f               f              f                 f  …            t
      3        f               f              f                 f  …            t
      4        f               f              f                 f  …            t

         TT4 T4U measured   T4U FTI measured  FTI TBG measured TBG referral source  \
      0  125            t  1.14            t  109            f   ?            SVHC
      1  102            f     ?            f    ?            f   ?           other
      2  109            t  0.91            t  120            f   ?           other
      3  175            f     ?            f    ?            f   ?           other
      4   61            t  0.87            t   70            f   ?             SVI

        binaryClass
      0           P
      1           P
      2           P
      3           P
      4           P

      [5 rows x 30 columns]
```

```
[ ]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 3772 entries, 0 to 3771
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        3772 non-null   object
 1   sex                        3772 non-null   object
 2   on thyroxine               3772 non-null   object
 3   query on thyroxine         3772 non-null   object
 4   on antithyroid medication  3772 non-null   object
 5   sick                       3772 non-null   object
 6   pregnant                   3772 non-null   object
 7   thyroid surgery            3772 non-null   object
 8   I131 treatment             3772 non-null   object
 9   query hypothyroid          3772 non-null   object
 10  query hyperthyroid         3772 non-null   object
 11  lithium                    3772 non-null   object
 12  goitre                     3772 non-null   object
 13  tumor                      3772 non-null   object
 14  hypopituitary              3772 non-null   object
 15  psych                      3772 non-null   object
 16  TSH measured               3772 non-null   object
 17  TSH                        3772 non-null   object
 18  T3 measured                3772 non-null   object
 19  T3                         3772 non-null   object
 20  TT4 measured               3772 non-null   object
 21  TT4                        3772 non-null   object
 22  T4U measured               3772 non-null   object
 23  T4U                        3772 non-null   object
 24  FTI measured               3772 non-null   object
 25  FTI                        3772 non-null   object
 26  TBG measured               3772 non-null   object
 27  TBG                        3772 non-null   object
 28  referral source            3772 non-null   object
 29  binaryClass                3772 non-null   object
dtypes: object(30)
memory usage: 884.2+ KB
```

[ ]:

```
print(data.isnull().sum())
```

```
age                          0
sex                          0
on thyroxine                 0
query on thyroxine           0
on antithyroid medication    0
sick                         0
pregnant                     0
thyroid surgery              0
I131 treatment               0
```

```
query hypothyroid            0
query hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
hypopituitary                0
psych                        0
TSH measured                 0
TSH                          0
T3 measured                  0
T3                           0
TT4 measured                 0
TT4                          0
T4U measured                 0
T4U                          0
FTI measured                 0
FTI                          0
TBG measured                 0
TBG                          0
referral source              0
binaryClass                  0
dtype: int64
```

[6]:

```python
from sklearn.preprocessing import LabelEncoder

# Create a LabelEncoder instance
enc = LabelEncoder()

# Encode only categorical columns
categorical_cols = data.select_dtypes(include=['object']).columns
for col in categorical_cols:
    data[col] = enc.fit_transform(data[col])

data.info()  # Check the data types again
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3772 entries, 0 to 3771
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        3772 non-null   int64
 1   sex                        3772 non-null   int64
 2   on thyroxine               3772 non-null   int64
 3   query on thyroxine         3772 non-null   int64
 4   on antithyroid medication  3772 non-null   int64
 5   sick                       3772 non-null   int64
 6   pregnant                   3772 non-null   int64
 7   thyroid surgery            3772 non-null   int64
 8   I131 treatment             3772 non-null   int64
 9   query hypothyroid          3772 non-null   int64
 10  query hyperthyroid         3772 non-null   int64
 11  lithium                    3772 non-null   int64
 12  goitre                     3772 non-null   int64
 13  tumor                      3772 non-null   int64
 14  hypopituitary              3772 non-null   int64
 15  psych                      3772 non-null   int64
 16  TSH measured               3772 non-null   int64
 17  TSH                        3772 non-null   int64
 18  T3 measured                3772 non-null   int64
 19  T3                         3772 non-null   int64
 20  TT4 measured               3772 non-null   int64
 21  TT4                        3772 non-null   int64
 22  T4U measured               3772 non-null   int64
 23  T4U                        3772 non-null   int64
 24  FTI measured               3772 non-null   int64
 25  FTI                        3772 non-null   int64
 26  TBG measured               3772 non-null   int64
 27  TBG                        3772 non-null   int64
 28  referral source            3772 non-null   int64
 29  binaryClass                3772 non-null   int64
dtypes: int64(30)
memory usage: 884.2 KB
```

```
[ ]: data.head()
```

```
[ ]:    age  sex  on thyroxine  query on thyroxine  on antithyroid medication  \
     0   34    1             0                   0                          0
     1   15    1             0                   0                          0
     2   40    2             0                   0                          0
     3   67    1             1                   0                          0
     4   67    1             0                   0                          0

        sick  pregnant  thyroid surgery  I131 treatment  query hypothyroid  …  \
     0     0         0                0               0                  0  …
     1     0         0                0               0                  0  …
     2     0         0                0               0                  0  …
     3     0         0                0               0                  0  …
     4     0         0                0               0                  0  …

        TT4 measured  TT4  T4U measured  T4U  FTI measured  FTI  TBG measured  TBG  \
     0             1   28             1   72             1   10             0    0
     1             1    3             0  146             0  234             0    0
     2             1   10             1   48             1   22             0    0
     3             1   83             0  146             0  234             0    0
     4             1  201             1   44             1  199             0    0
```

```
   referral source  binaryClass
0                1            1
1                4            1
2                4            1
3                4            1
4                3            1

[5 rows x 30 columns]
```

[ ]:

```
data=data.drop_duplicates()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 3711 entries, 0 to 3771
Data columns (total 30 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   age                        3711 non-null   int64
 1   sex                        3711 non-null   int64
 2   on thyroxine               3711 non-null   int64
 3   query on thyroxine         3711 non-null   int64
 4   on antithyroid medication  3711 non-null   int64
 5   sick                       3711 non-null   int64
 6   pregnant                   3711 non-null   int64
 7   thyroid surgery            3711 non-null   int64
 8   I131 treatment             3711 non-null   int64
 9   query hypothyroid          3711 non-null   int64
 10  query hyperthyroid         3711 non-null   int64
 11  lithium                    3711 non-null   int64
 12  goitre                     3711 non-null   int64
 13  tumor                      3711 non-null   int64
 14  hypopituitary              3711 non-null   int64
 15  psych                      3711 non-null   int64
 16  TSH measured               3711 non-null   int64
 17  TSH                        3711 non-null   int64
 18  T3 measured                3711 non-null   int64
 19  T3                         3711 non-null   int64
 20  TT4 measured               3711 non-null   int64
 21  TT4                        3711 non-null   int64
 22  T4U measured               3711 non-null   int64
 23  T4U                        3711 non-null   int64
 24  FTI measured               3711 non-null   int64
 25  FTI                        3711 non-null   int64
 26  TBG measured               3711 non-null   int64
 27  TBG                        3711 non-null   int64
 28  referral source            3711 non-null   int64
 29  binaryClass                3711 non-null   int64
dtypes: int64(30)
memory usage: 898.8 KB
```

[ ]:
```
data.describe()
```

[ ]:
```
               age          sex  on thyroxine  query on thyroxine  \
count  3711.000000  3711.000000   3711.000000         3711.000000
mean     46.493937     1.266505      0.125034            0.013473
std      20.863642     0.525220      0.330802            0.115306
min       0.000000     0.000000      0.000000            0.000000
25%      28.000000     1.000000      0.000000            0.000000
50%      50.000000     1.000000      0.000000            0.000000
75%      63.000000     2.000000      0.000000            0.000000
max      93.000000     2.000000      1.000000            1.000000

       on antithyroid medication         sick     pregnant  thyroid surgery  \
count                3711.000000  3711.000000  3711.000000      3711.000000
mean                    0.011318     0.039612     0.014282         0.014282
std                     0.105795     0.195072     0.118666         0.118666
min                     0.000000     0.000000     0.000000         0.000000
25%                     0.000000     0.000000     0.000000         0.000000
50%                     0.000000     0.000000     0.000000         0.000000
75%                     0.000000     0.000000     0.000000         0.000000
max                     1.000000     1.000000     1.000000         1.000000

       I131 treatment  query hypothyroid  …  TT4 measured          TT4  \
count     3711.000000        3711.000000  …   3711.000000  3711.000000
mean         0.015899           0.063056  …      0.953921   119.133118
std          0.125100           0.243096  …      0.209685    98.238113
min          0.000000           0.000000  …      0.000000     0.000000
25%          0.000000           0.000000  …      1.000000    21.000000
50%          0.000000           0.000000  …      1.000000    79.000000
75%          0.000000           0.000000  …      1.000000   226.000000
max          1.000000           1.000000  …      1.000000   241.000000

       T4U measured          T4U  FTI measured          FTI  TBG measured  \
count   3711.000000  3711.000000   3711.000000  3711.000000        3711.0
mean       0.911884    64.854756      0.912423   108.715980           0.0
std        0.283502    31.330172      0.282718    97.032357           0.0
min        0.000000     0.000000      0.000000     0.000000           0.0
25%        1.000000    46.000000      1.000000    17.000000           0.0
50%        1.000000    57.000000      1.000000    56.000000           0.0
75%        1.000000    71.000000      1.000000   221.000000           0.0
max        1.000000   146.000000      1.000000   234.000000           0.0

          TBG  referral source  binaryClass
```

```
count  3711.0      3711.000000  3711.000000
mean      0.0         3.267583     0.921584
std       0.0         1.097079     0.268861
min       0.0         0.000000     0.000000
25%       0.0         3.000000     1.000000
50%       0.0         4.000000     1.000000
75%       0.0         4.000000     1.000000
max       0.0         4.000000     1.000000

[8 rows x 30 columns]
```

[ ]:

```python
# Convert relevant columns to numeric, forcing errors to NaN
data['age'] = pd.to_numeric(data['age'], errors='coerce')
data['TT4'] = pd.to_numeric(data['TT4'], errors='coerce')
data['T4U'] = pd.to_numeric(data['T4U'], errors='coerce')
data['FTI'] = pd.to_numeric(data['FTI'], errors='coerce')

# Check for missing values after conversion
print(data.isnull().sum())

# Handle missing values if necessary (e.g., drop or fill)
data = data.dropna()  # Example: drop rows with missing values

# Normalize the columns
data['age'] = (data['age'] - data['age'].min()) / (data['age'].max() - data['age'].min())
data['TT4'] = (data['TT4'] - data['TT4'].min()) / (data['TT4'].max() - data['TT4'].min())
data['T4U'] = (data['T4U'] - data['T4U'].min()) / (data['T4U'].max() - data['T4U'].min())
data['FTI'] = (data['FTI'] - data['FTI'].min()) / (data['FTI'].max() - data['FTI'].min())
```

```
age                          0
sex                          0
on thyroxine                 0
query on thyroxine           0
on antithyroid medication    0
sick                         0
pregnant                     0
thyroid surgery              0
I131 treatment               0
query hypothyroid            0
query hyperthyroid           0
lithium                      0
goitre                       0
tumor                        0
```

```
hypopituitary                0
psych                        0
TSH measured                 0
TSH                          0
T3 measured                  0
T3                           0
TT4 measured                 0
TT4                          0
T4U measured                 0
T4U                          0
FTI measured                 0
FTI                          0
TBG measured                 0
TBG                          0
referral source              0
binaryClass                  0
dtype: int64
```

[ ]:
```
data.head()
```

[ ]:
```
        age  sex  on thyroxine  query on thyroxine  on antithyroid medication  \
0  0.365591    1             0                   0                          0
1  0.161290    1             0                   0                          0
2  0.430108    2             0                   0                          0
3  0.720430    1             1                   0                          0
4  0.720430    1             0                   0                          0

   sick  pregnant  thyroid surgery  I131 treatment  query hypothyroid  …  \
0     0         0                0               0                  0  …
1     0         0                0               0                  0  …
2     0         0                0               0                  0  …
3     0         0                0               0                  0  …
4     0         0                0               0                  0  …

   TT4 measured       TT4  T4U measured       T4U  FTI measured       FTI  \
0             1  0.116183             1  0.493151             1  0.042735
1             1  0.012448             0  1.000000             0  1.000000
2             1  0.041494             1  0.328767             1  0.094017
3             1  0.344398             0  1.000000             0  1.000000
4             1  0.834025             1  0.301370             1  0.850427

   TBG measured  TBG  referral source  binaryClass
0             0    0                1            1
1             0    0                4            1
2             0    0                4            1
3             0    0                4            1
4             0    0                3            1
```

```
[5 rows x 30 columns]
```

[ ]:

```
y=data['binaryClass']
x=data.drop(['binaryClass'],axis=1)
```

[ ]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest= train_test_split(x,y,test_size=0.1,stratify=y)
print(xtrain.shape)
print(xtest.shape)
print(ytrain.shape)
print(ytest.shape)
```

```
(3339, 29)
(372, 29)
(3339,)
(372,)
```

[ ]:

```
# Ensure the cell that loads the training data is executed
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.1, stratify=y)

# Check for errors in data loading
if 'xtrain' in globals():
    # Fit the XGBClassifier model
    from xgboost import XGBClassifier
    svm_model = XGBClassifier()
    svm_model.fit(xtrain, ytrain)
else:
    print("Model not found in the notebook. Please ensure a model is trained to␣
 ↪view feature importance.")
```

[ ]:

```
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report

# Make predictions on the test set
predictions = svm_model.predict(xtest)

# Calculate accuracy
percentage = svm_model.score(xtest, ytest)

# Generate confusion matrix
res = confusion_matrix(ytest, predictions)

# Print validation confusion matrix
print("Validation Confusion Matrix:")
print(res)
```

```python
# Print classification report
print("Classification Report:")
print(classification_report(ytest, predictions))

# Check the accuracy on the training set
training_accuracy = svm_model.score(xtrain, ytrain) * 100
testing_accuracy = percentage * 100

print('Training Accuracy = {:.2f}%'.format(training_accuracy))
print('Testing Accuracy = {:.2f}%'.format(testing_accuracy))
```

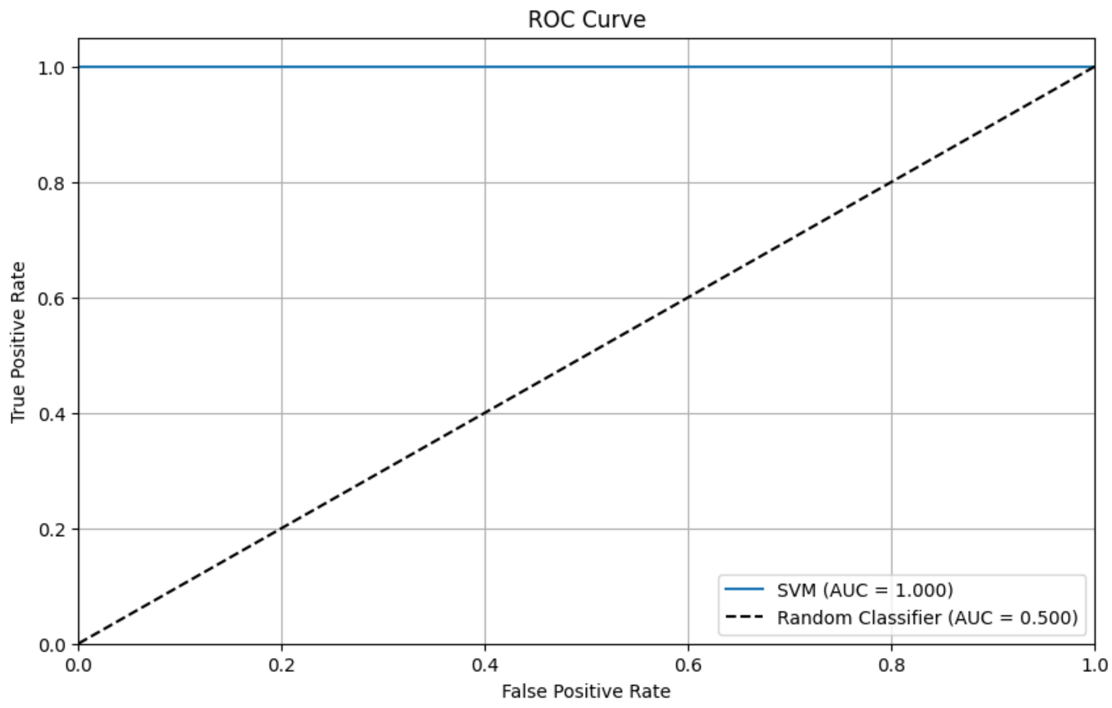
```
Validation Confusion Matrix:
[[ 25   4]
 [  2 347]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.86      0.89        29
           1       0.99      0.99      0.99       349

    accuracy                           0.98       378
   macro avg       0.96      0.93      0.94       378
weighted avg       0.98      0.98      0.98       378

Training Accuracy = 100.00%
Testing Accuracy = 98.41%
```

Since the model has performed well on the test set, it doesnt require any hyperparameter tuning.

[15]:

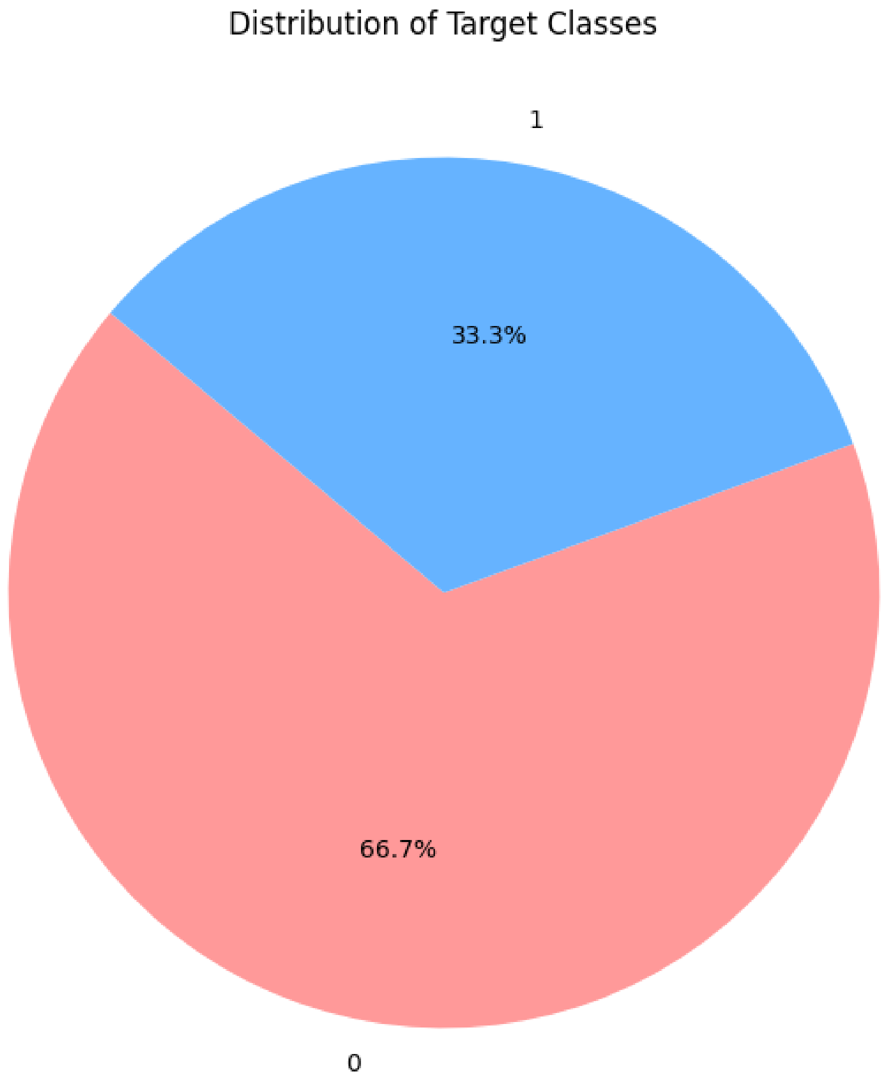
```python
from sklearn import svm
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

# Load a sample dataset (e.g., Iris)
data = load_iris()
X = data.data
y = (data.target == 2).astype(int)  # Convert to binary classification (e.g.,␣
 ↪class 2 vs rest)

# Split the dataset into training and test sets
Xtrain, xtest, ytrain, ytest = train_test_split(X, y, test_size=0.3,␣
 ↪random_state=42)

# Create and train an SVM model with probability estimates enabled
svm_model = make_pipeline(StandardScaler(), svm.SVC(probability=True))
```

```
svm_model.fit(Xtrain, ytrain)

# Now you can use the predict_proba method as in your original code
y_pred_proba = svm_model.predict_proba(xtest)[:, 1]

# Calculate FPR, TPR and threshold values
fpr, tpr, thresholds = roc_curve(ytest, y_pred_proba)

# Plot ROC curve
plt.figure(figsize=(10, 6))
plt.plot(fpr, tpr, label='SVM (AUC = {:.3f})'.format(roc_auc_score(ytest,␣
 ↪y_pred_proba)))
plt.plot([0, 1], [0, 1], 'k--', label='Random Classifier (AUC = 0.500)')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.grid(True)
plt.show()
```



[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Convert y to a Pandas Series
y_series = pd.Series(y)

# Get the class distribution
class_counts = y_series.value_counts()

# Plot a pie chart of the target class distribution
plt.figure(figsize=(8, 8))
plt.pie(class_counts, labels=class_counts.index, autopct='%1.1f%%', startangle=140, colors=['#ff9999','#66b3ff'])
plt.title('Distribution of Target Classes')
plt.show()
```


Distribution of Target Classes