Guided Project

Project Workspace (https://naanmudhalvan.smartinternz.com/Student/guided_project_workspace/25718)

Chat with Mentor (https://naanmudhalvan.smartinternz.com/student-chat-with-mentor/25718)

Project Manual

(https://smartinternz.s3.amazonaws.com/Thyroid_disease_classification_using_ML.pdf)

Converting The Data Type

Handling Categorical Values

Splitting Data Into Train And Test

Handling Imbalanced Data

Applying StandardScaler

Performing Feature Importance

Selecting Output Columns

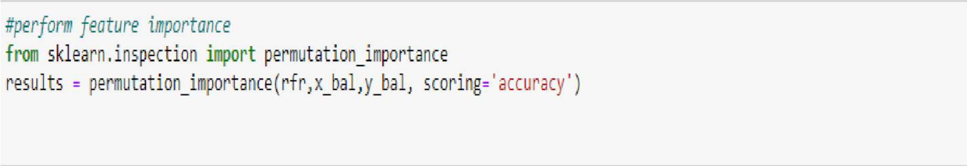
⊕ Exploratory Data Analysis

⊕ Model Building

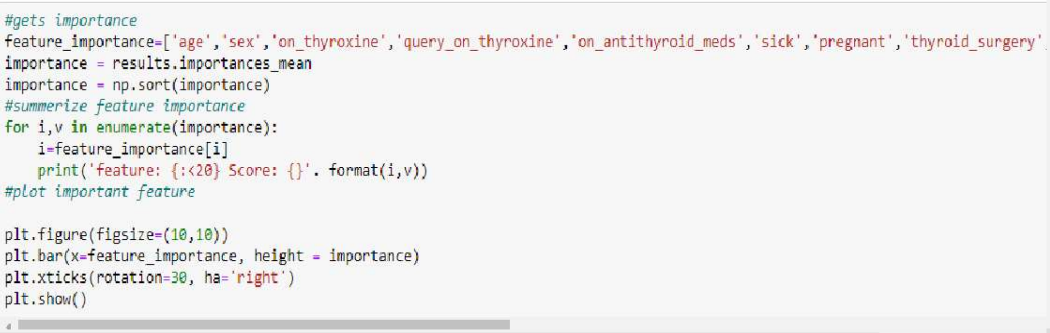
⊕ Performance Testing & Hyperparameter Tuning

Performing Feature Importance

- The idea behind permutation feature importance is simple. The feature importance is calculated by noticing the increase or decrease in error when we permute the values of a feature.
- If permuting the values causes a huge change in the error, it means the feature is important for our model.



```
#perform feature importance
from sklearn.inspection import permutation_importance
results = permutation_importance(rfr,x_bal,y_bal, scoring='accuracy')
```



```
#gets importance
feature_importance=['age','sex','on_thyroxine','query_on_thyroxine','on_antithyroid_meds','sick','pregnant','thyroid_surgery',
importance = results.importances_mean
importance = np.sort(importance)
#summerize feature importance
for i,v in enumerate(importance):
    i=feature_importance[i]
    print('feature: {:<20} Score: {}'. format(i,v))
#plot important feature

plt.figure(figsize=(10,10))
plt.bar(x=feature_importance, height = importance)
plt.xticks(rotation=30, ha='right')
plt.show()
```