**LAPORAN PRAKTIKUM PEMBELAJARAN MESIN C**

**Tugas 2**

**Dosen Pengampu :**

**Drs.Hari Suparwito, S.J. M.App.IT**



**GABRIEL BAYU HANDRIYANDANAR**

**225314095**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA**

**YOGYAKARTA**

**2024**

1. **Listing Code**

* Melihat Data

| Python Code | Penjelasan |
| --- | --- |
|  | Melakukan import library yang diperlukan seperti numpy, matplotlib, seaborn, dan pandas untuk analisis data, serta sklearn untuk pemodelan. Warnings diabaikan untuk meminimalkan gangguan. Fungsi-fungsi seperti confusion\_matrix, roc\_curve, dan auc diimport untuk evaluasi model nantinya. |
|  | Membaca file CSV yang berisi data tentang ukuran penguin. Data tersebut dimuat ke dalam dataframe df menggunakan library pandas. |
|  | Mencetak 5 data pertama dari dataframe df |
|  | Mencetak ukuran baris dan kolom dari dataframe df |
|  | Menampilkan informasi tentang dataframe df, seperti jumlah baris dan kolom, nama kolom beserta tipe datanya, serta jumlah nilai non-null dan penggunaan memori. |
|  | Menghitung jumlah nilai null (NaN) dalam setiap kolom dataframe df |
|  | Menampilkan baris-baris dalam dataframe df yang mengandung setidaknya satu nilai null (NaN) |

* Handling Missing Values And EDA

| Python Code | Penjelasan |
| --- | --- |
|  | Menghapus baris ke-3 dan ke-339 dari dataframe df menggunakan metode .drop(). Setelah baris-baris tersebut dihapus, kita menggunakan atribut .shape untuk melihat dimensi dataframe df, yaitu jumlah baris dan kolom setelah penghapusan. |
|  | Menghitung jumlah nilai null (NaN) dalam setiap kolom dataframe df |
|  | Menghitung jumlah kemunculan setiap nilai unik dalam kolom 'species' dari dataframe df, dan hasilnya disimpan dalam variabel Species |
|  | Memberikan gambaran mengenai berbagai nilai unik yang ada dalam setiap kolom dari dataframe df |
|  | Menghapus baris-baris dalam dataframe df yang mengandung nilai NaN (Not a Number) atau nilai yang hilang. |
|  | Membersihkan dataframe df dari baris yang memiliki nilai '.' pada kolom sex |
|  | Membuat palet warna menggunakan palet crayon dari library seaborn. Palet tersebut terdiri dari warna merah, kuning, dan biru |
|  | Membuat sebuah diagram pie yang menunjukkan persentase dari setiap spesies dalam data menggunakan matplotlib. |
|  | Membuat sebuah countplot yang menampilkan jumlah setiap jenis kelamin penguin dalam data menggunakan seaborn. |
|  | Membuat countplot dengan hue berdasarkan kolom 'sex' dan sumbu x berdasarkan kolom 'species' dari dataframe dfmenggunakan seaborn. |
|  | Mengelompokkan data berdasarkan spesies dan jenis kelamin, kemudian menghitung jumlah setiap kombinasi spesies dan jenis kelamin. |
|  | Membuat countplot dengan menggunakan sumbu x berdasarkan kolom 'island' dan hue berdasarkan kolom 'species' dari dataframe df |
|  | membuat boxplot dengan menggunakan data dari dataframe df. Sumbu x adalah 'species', sumbu y adalah 'culmen\_length\_mm', dan hue adalah 'sex'. |
|  | Mengelompokkan data berdasarkan spesies dan jenis kelamin, kemudian menghitung beberapa statistik deskriptif untuk panjang culmen (culmen\_length\_mm) dari setiap kelompok. |
|  | Membuat swarmplot dengan menggunakan data dari dataframe df. Sumur x adalah 'species', sumbu y adalah 'culmen\_depth\_mm', dan hue adalah 'sex'. |
|  | Mengelompokkan data berdasarkan spesies dan jenis kelamin, kemudian menghitung rata-rata kedalaman culmen (culmen\_depth\_mm) untuk setiap kelompok. |
|  | Membuat stripplot dengan menggunakan data dari dataframe df. Sumbu y adalah 'flipper\_length\_mm', sumbu x adalah 'island', dan hue adalah 'species'. |
|  | Menggunakan FacetGrid dari seaborn untuk membuat grid plot yang membandingkan distribusi massa tubuh (body\_mass\_g) penguin berdasarkan spesies dengan membaginya berdasarkan jenis kelamin. |
|  | Mengelompokkan data berdasarkan spesies dan jenis kelamin, kemudian menghitung standar deviasi dari massa tubuh (body\_mass\_g) untuk setiap kelompok. |
|  | Membuat countplot untuk menghitung jumlah pengamatan untuk setiap spesies penguin. |
|  | Membuat countplot untuk menghitung jumlah pengamatan untuk setiap pulau tempat tinggal penguin. |
|  | Membuat countplot dengan menggunakan data dari dataframe df, dengan sumbu x sebagai 'species' dan menggunakan hue berdasarkan 'island'. |
|  | Membuat dataframe baru dengan nama new\_df yang berisi salinan dari df yang sudah di preprocessing |
|  | menghapus kolom 'species' dan 'island' dari DataFrame new\_df |

* Supervised Learning
  + Cross Validation

| Python Code | Penjelasan |
| --- | --- |
|  | Mengimpor library yang diperlukan dari scikit-learn untuk melakukan pemodelan dan evaluasi model. |
|  | Membuat objek k\_fold dari KFold dengan jumlah lipatan (n\_splits) sebanyak 10, melakukan pengacakan (shuffle) |
|  | Menginisialisasi variabel train dan target yang akan digunakan dalam pemodelan. train berisi fitur-fitur yang akan digunakan untuk melatih model, sementara target berisi label yang akan diprediksi oleh model. |

* + Random Forest Classifier

| Python Code | Penjelasan |
| --- | --- |
|  | Menggunakan Hyperparameter Tuning algoritma pencarian hiperparameter untuk mencari kombinasi terbaik dari nilai-nilai ini yang menghasilkan kinerja model yang optimal. |
|  | Melakukan pencarian parameter secara efisien dengan menggunakan modul BayesSearchCV dari library scikit-Optimize |
|  | membuat objek RandomForestClassifier() dengan menggunakan parameter terbaik yang telah ditemukan melalui pencarian hiperparameter sebelumnya. Parameter-parameter tersebut adalah max\_depth=4, max\_features='sqrt', max\_leaf\_nodes=5, dan n\_estimators=43. Objek model yang telah dibuat disimpan dalam variabel rf\_model. |
|  | Menggunakan cross\_val\_predict untuk melakukan cross-validation terhadap model model dengan menggunakan data train dan target. cv=k\_fold mengatur penggunaan KFold sebagai metode cross-validation dengan 10 lipatan. n\_jobs=5 mengatur penggunaan 5 thread untuk proses paralel, dan scoring='accuracy' mengatur metrik evaluasi yang digunakan menjadi akurasi. |
|  | Menghitung dan memvisualisasikan confusion matrix dari model klasifikasi Random Forest yang telah dibuat sebelumnya dengan menggunakan heatmap dari library seaborn. Heatmap ini menggunakan warna untuk menunjukkan jumlah prediksi yang benar dan salah dengan lebih jelas. |
|  | mencetak laporan klasifikasi yang mengandung beberapa metrik evaluasi untuk model klasifikasi Random Forest yang telah dibuat sebelumnya. Laporan klasifikasi ini mencakup metrik seperti presisi, recall, dan f1-score untuk setiap kelas, serta rata-rata dari metrik-metrik tersebut. |
|  | Menghitung dan mencetak akurasi dari model klasifikasi Random Forest yang telah dibuat sebelumnya. |
|  | Menggunakan LabelEncoder dari scikit-learn untuk mengkodekan label kelas menjadi nilai numerik. Hasilnya disimpan dalam variabel target\_encoded. |
|  | model Random Forest yang telah dibuat (rf\_model) dilatih menggunakan data latih (train) dan target yang telah dikodekan (target\_encoded). Setelah model dilatih, program berjalan menggunakan metode cross-validation untuk membuat prediksi probabilitas menggunakan fungsi cross\_val\_predict(). |
|  | Menghitung nilai False Positive Rate (FPR), True Positive Rate (TPR), dan thresholds untuk kurva ROC (Receiver Operating Characteristic) menggunakan fungsi roc\_curve() dari scikit-learn. Dan menghitung nilai Area Under Curve (AUC) untuk kurva ROC menggunakan fungsi roc\_auc\_score(). |
|  | Memvisualisasikan plot kurva ROC (Receiver Operating Characteristic) dari model klasifikasi Random Forest yang telah dibuat sebelumnya. |

* + Support Vector Classifier

| Python Code | Penjelasan |
| --- | --- |
|  | Mendefinisikan grid parameter untuk melakukan grid search pada model Support Vector Classifier (SVC).Parameter C dan gamma didefinisikan sebagai rentang nilai yang akan diuji dalam pencarian hiperparameter. C adalah parameter penalti error, sedangkan gamma adalah koefisien kernel untuk 'rbf'. Dan juga menggunakan Real untuk menentukan rentang nilai bilangan real dan Categorical untuk parameter diskrit. |
|  | Membuat objek model SVC dan objek bayes\_search. param\_grid adalah grid parameter yang akan diuji, cv adalah metode cross-validation yang akan digunakan, n\_jobs adalah jumlah thread yang akan digunakan untuk proses paralel, dan scoring adalah metrik evaluasi yang akan digunakan.Lalu, menjalankan grid search untuk mencari parameter terbaik dengan menggunakan data train dan target. |
|  | Mencetak hasil terbaik dari bayes search, yaitu hyperparameter terbaik dan akurasi terbaik. |
|  | Membuat objek model SVC dengan menggunakan hyperparameter terbaik yang ditemukan dari grid search. Hyperparameter terbaik adalah C=0.1, gamma=1, dan kernel='poly'. |
|  | Menggunakan cross\_val\_predict untuk melakukan cross-validation terhadap model svc\_model dengan menggunakan data train dan target. cv=k\_fold mengatur penggunaan KFold sebagai metode cross-validation dengan 10 lipatan. n\_jobs=5 mengatur penggunaan 5 thread untuk proses paralel, dan scoring='accuracy' mengatur metrik evaluasi yang digunakan menjadi akurasi. |
|  | Menghitung dan memvisualisasikan confusion matrix dari model klasifikasi Support Vector yang telah dibuat sebelumnya dengan menggunakan heatmap dari library seaborn. Heatmap ini menggunakan warna untuk menunjukkan jumlah prediksi yang benar dan salah dengan lebih jelas. |
|  | mencetak laporan klasifikasi yang mengandung beberapa metrik evaluasi untuk model klasifikasi Support Vector yang telah dibuat sebelumnya. Laporan klasifikasi ini mencakup metrik seperti presisi, recall, dan f1-score untuk setiap kelas, serta rata-rata dari metrik-metrik tersebut. |
|  | Mencetak akurasi dari klasifikasi Support Vector |
|  | Melatih model SVC menggunakan data latih dan target yang telah dikodekan, kemudian menggunakan metode cross-validation untuk membuat prediksi probabilitas menggunakan fungsi cross\_val\_predict() dengan metode decision\_function. |
|  | Menggunakan nilai probabilitas yang dihasilkan dari metode decision\_function untuk menghitung False Positive Rate (FPR), True Positive Rate (TPR), dan thresholds untuk kurva ROC (Receiver Operating Characteristic) menggunakan fungsi roc\_curve() dari scikit-learn. Kemudian, nilai Area Under Curve (AUC) untuk kurva ROC dihitung menggunakan fungsi roc\_auc\_score() |
|  | memvisualisasikan plot kurva ROC (Receiver Operating Characteristic) dari model Support Vector Classifier (SVC) yang telah dilatih sebelumnya. |

* + Naive Bayes

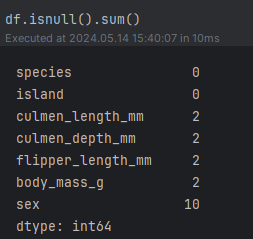
| Python Code | Penjelasan |
| --- | --- |
|  | Mengimpor Library untuk digunakan nantinya dalam metode Naive Bayes |
|  | Melakukan pencarian parameter secara efisien pada model Gaussian Naive Bayes (GaussianNB) menggunakan Bayes Search |
|  | Mencetak Hyperparameter terbaik beserta akurasi terbaik dari parameter tersebut |
|  | Membuat objek model GaussianNB dengan parameter yang sudah dituning menggunakan Bayes Search |
|  | Menggunakan metode cross\_val\_predict() dari scikit-learn untuk membuat prediksi menggunakan model Gaussian Naive Bayes (NB\_model) dengan menggunakan metode cross-validation. |
|  | Menggunakan metode cross\_val\_predict() dari scikit-learn untuk membuat prediksi menggunakan model Gaussian Naive Bayes (NB\_model) dengan menggunakan metode cross-validation. |
|  | Menghitung dan memvisualisasikan confusion matrix dari model Gaussian Naive Bayes yang telah dibuat sebelumnya. |
|  | Mencetak laporan klasifikasi yang berisi beberapa metrik evaluasi untuk model Gaussian Naive Bayes yang telah dibuat. |
|  | Menghitung dan mencetak akurasi dari model Gaussian Naive Bayes yang telah dibuat sebelumnya. |
|  | Model Gaussian Naive Bayes (NB\_model) dilatih menggunakan data latih (train) dan target yang telah dikodekan (target\_encoded). Kemudian, menggunakan metode cross-validation, kita membuat prediksi probabilitas menggunakan fungsi cross\_val\_predict() dengan metode 'predict\_proba'. |
|  | Menggunakan nilai probabilitas yang dihasilkan dari metode predict\_proba() untuk menghitung False Positive Rate (FPR), True Positive Rate (TPR), dan thresholds untuk kurva ROC (Receiver Operating Characteristic) menggunakan fungsi roc\_curve() dari scikit-learn.  Kemudian, nilai Area Under Curve (AUC) untuk kurva ROC dihitung menggunakan fungsi roc\_auc\_score(). |
|  | memvisualisasikan plot kurva ROC (Receiver Operating Characteristic) dari model Gaussian Naive Bayes yang telah dilatih sebelumnya. |

* + Artificial Neural Network

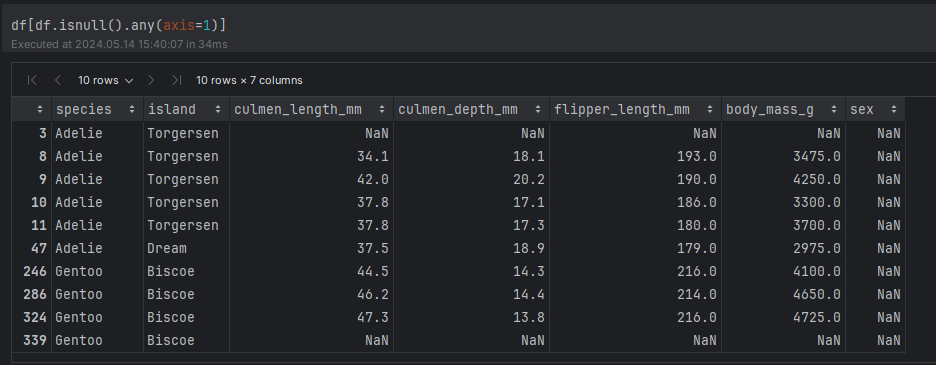
| Python Code | Penjelasan |
| --- | --- |
|  | Mengimport Library yang akan digunakan pada model Artificial Neural Network |
|  | Model ini memiliki tiga lapisan: lapisan masukan, dua lapisan tersembunyi, dan lapisan keluaran. Lapisan masukan memiliki 10 neuron dengan aktivasi ReLU, lapisan tersembunyi kedua memiliki 5 neuron dengan aktivasi ReLU, dan lapisan keluaran memiliki 1 neuron dengan aktivasi sigmoid. Jumlah neuron dalam lapisan masukan sesuai dengan jumlah fitur dalam data latih. |
|  | Mengompilasi model jaringan saraf tiruan (Artificial Neural Network/ANN) menggunakan optimizer 'adam', fungsi loss 'binary\_crossentropy', dan metrik evaluasi 'accuracy'. |
|  | Melatih model jaringan saraf tiruan (Artificial Neural Network/ANN) menggunakan metode fit(). |
|  | menggunakan model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih untuk membuat prediksi menggunakan data latih. |
|  | Menggunakan fungsi confusion\_matrix() dari scikit-learn untuk menghitung confusion matrix dari model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih sebelumnya. |
|  | Memvisualisasikan confusion matrix dari model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih sebelumnya. |
|  | menggunakan fungsi classification\_report() dari scikit-learn untuk mencetak laporan klasifikasi yang berisi beberapa metrik evaluasi untuk model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih sebelumnya. |
|  | Menggunakan fungsi accuracy\_score() dari scikit-learn untuk menghitung dan mencetak akurasi dari model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih sebelumnya. |
|  | Menggunakan model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih sebelumnya untuk membuat prediksi probabilitas menggunakan metode predict(). Kemudian, Anda menghitung Area Under Curve (AUC) untuk kurva ROC menggunakan fungsi roc\_auc\_score() dari scikit-learn. |
|  | Menggunakan nilai probabilitas yang dihasilkan dari metode predict() untuk menghitung False Positive Rate (FPR), True Positive Rate (TPR), dan thresholds untuk kurva ROC (Receiver Operating Characteristic) menggunakan fungsi roc\_curve() dari scikit-learn. |
|  | memvisualisasikan plot kurva ROC (Receiver Operating Characteristic) dari model jaringan saraf tiruan (Artificial Neural Network/ANN) yang telah dilatih sebelumnya. |

1. **Analisa**

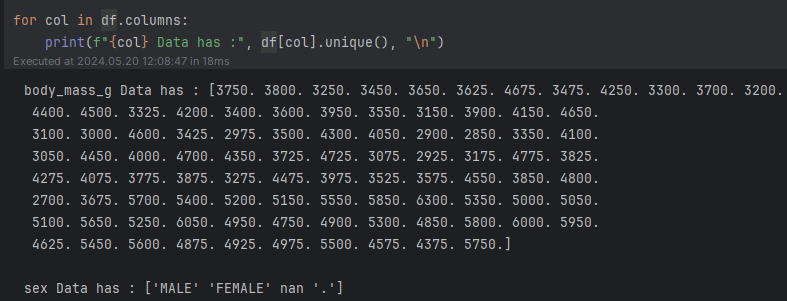
Data Penguin diawal menunjukkan adanya Missing Values yang berada di features culmen\_length\_mm dengan 2 data hilang, culmen\_depth\_mm dengan 2 data hilang, lalu flipper\_length\_mm dengan 2 data yang hilang, body\_mass\_g dengan 2 data yang hilang dan sex dengan 10 data yang hilang.



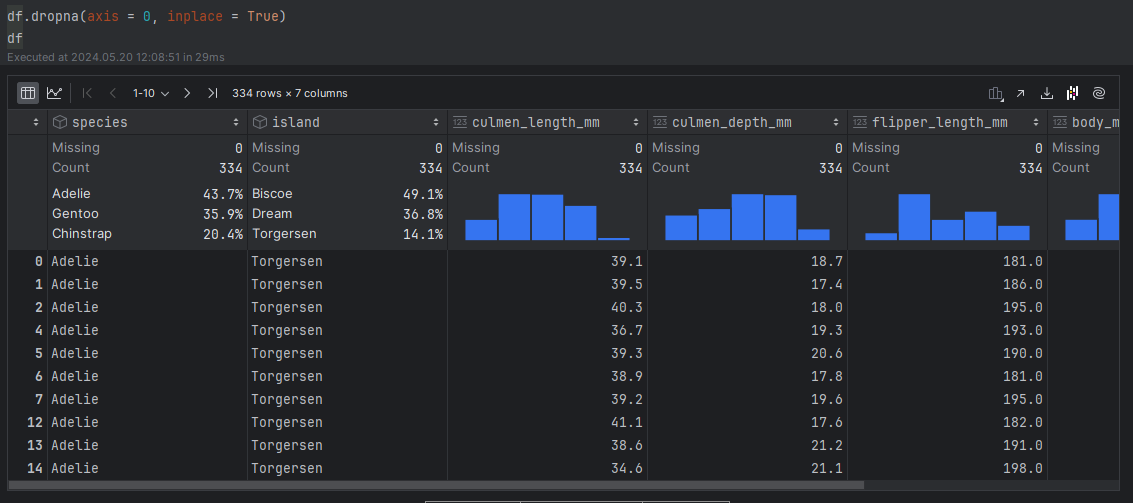
Data berikut menunjukkan baris mana yang mempunyai data yang bukan berupa angka (NaN)



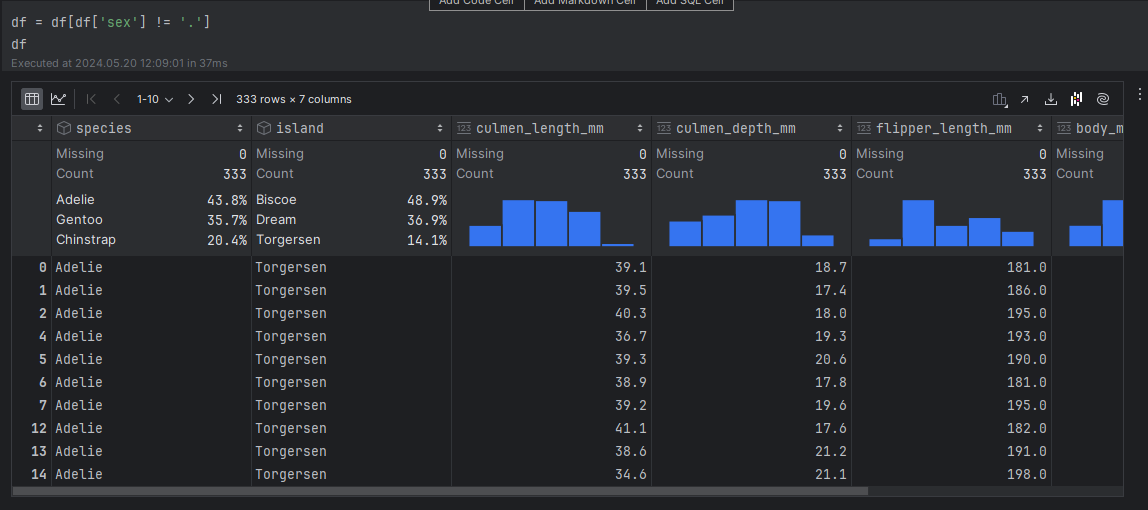
Pada data palmer penguin ini, setelah dilakukan pengecekan kolom, terdapat beberapa data di kolom sex yang datanya adalah NaN dan beranomali ‘.’. Berikut adalah isi datanya.



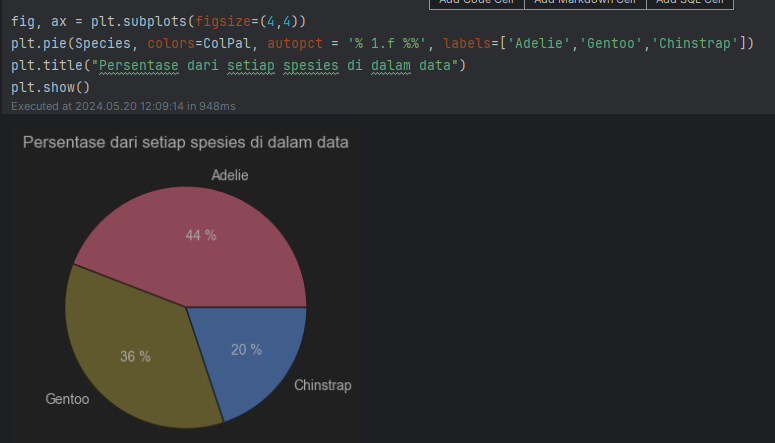
Untuk menghandle missing values, saya menghapus data tersebut karena hanya sebagian kecil yang merupakan NaN.



Lalu terdapat anomali yang mana di kolom sex, terdapat data yang berisi “.” lalu saya menghapusnya karena anomali nya hanya terdapat di 1 data saja

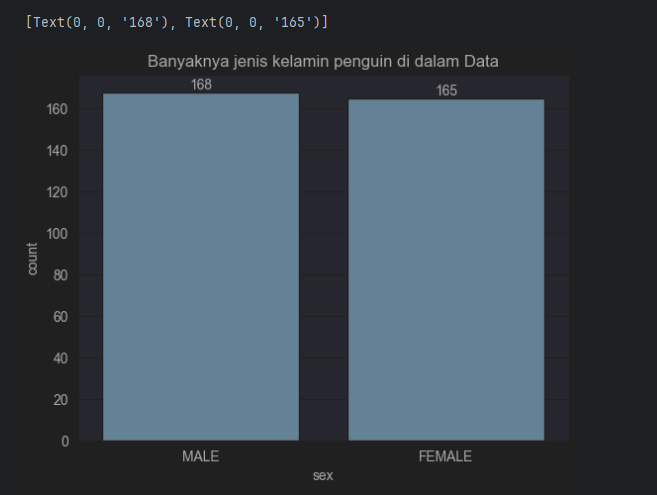


Untuk visualisasi nya, saya menggunakan Pie chart dari library matplotlib untuk memvisualisasi setiap spesies. Berikut adalah visualisasinya



Berdasarkan gambar diatas, Populasi spesies penguin terbesar adalah Penguin Adelie yang mana sebanyak 44% adalah Penguin Adelie dalam data tersebut, 36% Penguin Gentoo, dan 20% Penguin Chinstrap.

Visualisasi dibawah ini adalah Data banyak nya Jenis kelamin dalam data penguin, berikut adalah visualisasi nya



Dalam gambar diatas, sebanyak 168 penguin adalah berjenis kelamin Jantan, sedangkan 165 penguin berjenis kelamin Betina, Visualisasi ini merupakan gabungan dari semua jenis penguin, baik itu Adelie, Gentoo, dan Chinstrap.

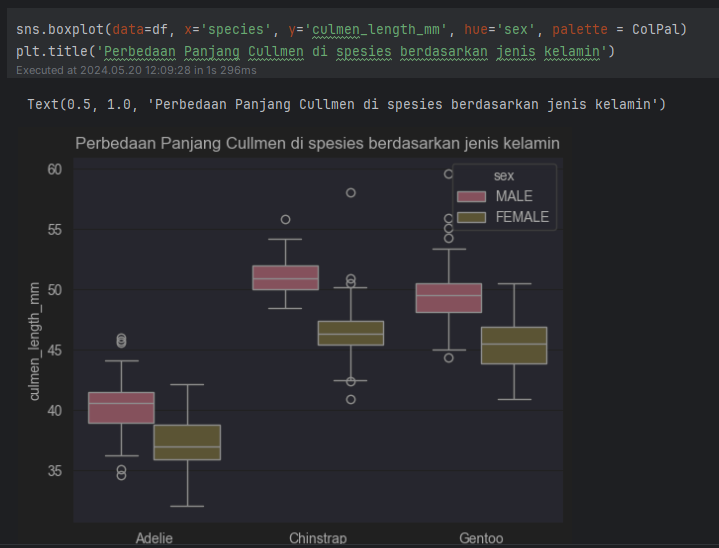
Lalu untuk visualisasi dari setiap jenis kelamin dari setiap spesies penguin sebagai berikut.

  
  
Untuk spesies Penguin Adelie dan Chinstrap, jumlah jenis kelamin antara jantan dan betina seimbang, yaitu dari Spesies Adelie sebanyak 73 Jantan dan 73 Betina, sedangkan dari spesies Chinstrap adalah 34 Jantan dan 34 Betina, sedangkan dari spesies Gentoo, sebanyak 61 adalah berjenis kelamin Jantan, dan 58 adalah berjenis kelamin Betina.

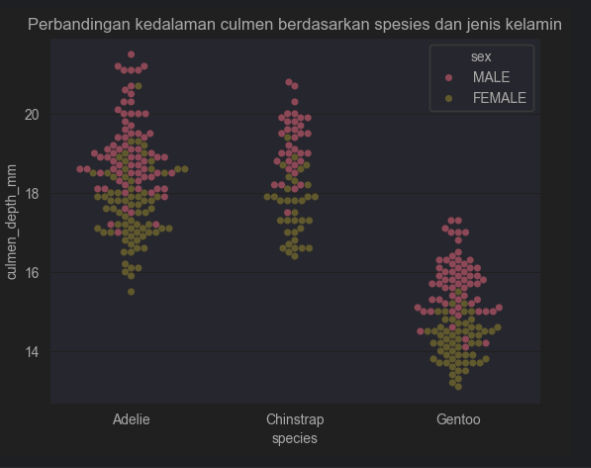
Untuk Visualisasi tempat tinggal penguin berdasarkan spesies nya adalah sebagai berikut



Dari Visualisasi diatas, menunjukkan bahwa Penguin Adelie bertempat tinggal di 3 pulau besar yaitu di pulau Dream sebanyak 55 penguin, Pulau Torgersen sebanyak 47 penguin, dan Pulau Biscoe sebanyak 44 penguin. Lalu untuk Penguin Chinstrap hanya ada di Pulau Dream dengan populasi sebanyak 68 penguin. Dan pulau Biscoe menjadi tempat tinggal Penguin Gentoo dengan populasi 119, lebih banyak dari penguin Adelie yang tinggal di Pulau Biscoe.



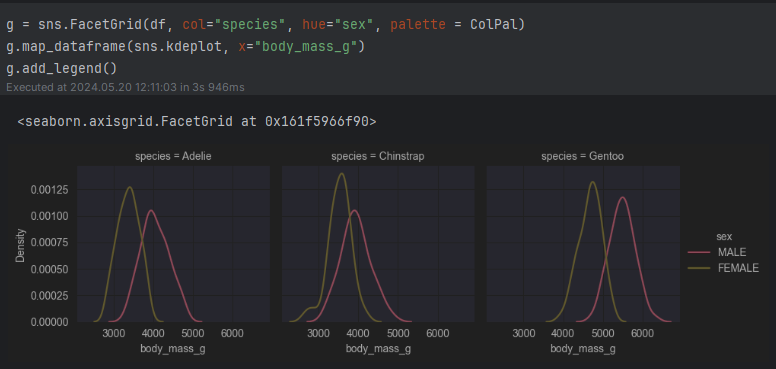
Plot ini menunjukkan adanya perbedaan yang signifikan dalam panjang culmen antara male dan female di semua spesies penguin yang diteliti. Male cenderung memiliki panjang culmen yang lebih besar dibandingkan female dalam spesies yang sama.



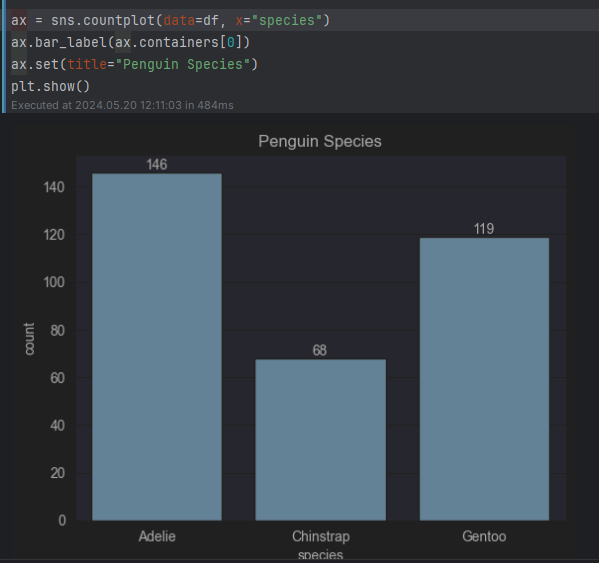
Plot ini menunjukkan bahwa ada beberapa perbedaan dalam distribusi kedalaman culmen di antara spesies penguin yang diteliti, serta perbedaan berdasarkan jenis kelamin. Meskipun ada variasi dalam distribusi, perbedaan berdasarkan jenis kelamin tidak terlalu mencolok untuk setiap spesies, kecuali mungkin sedikit pada spesies Gentoo.



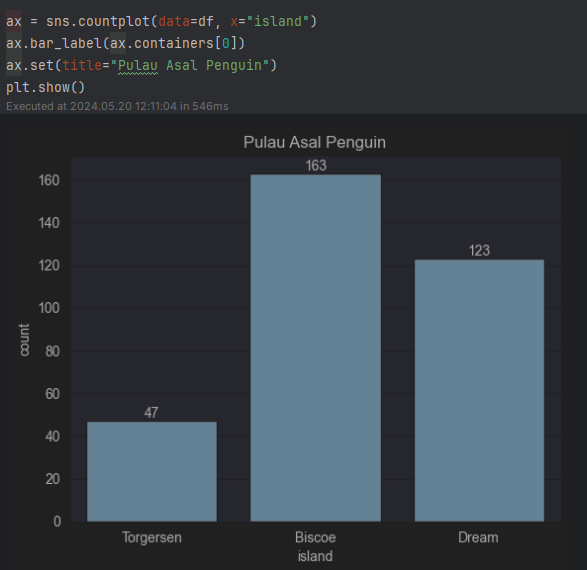
Plot ini menunjukkan perbedaan dalam distribusi panjang sirip penguin berdasarkan pulau dan spesies. Spesies Gentoo cenderung memiliki panjang sirip yang lebih besar dibandingkan spesies Adelie dan Chinstrap, terlepas dari pulau tempat mereka ditemukan. Pulau Biscoe dan Dream memiliki variasi spesies yang lebih besar dibandingkan pulau Torgersen, yang hanya memiliki spesies Adelie.



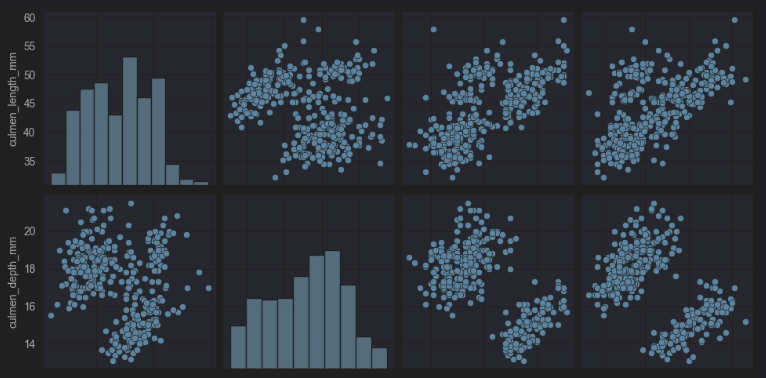
Plot ini menunjukkan perbandingan dari distribusi massa tubuh ketiga spesies penguin, Dari visualisasi diatas, kita bisa mengambil kesimpulan bahwa distribusi massa tubuh Betina dari setiap spesies penguin lebih tinggi daripada distribusi Jantan. Maka dari itu, spesies Betina dari semua penguin memiliki berat yang lebih … dari spesies Jantan

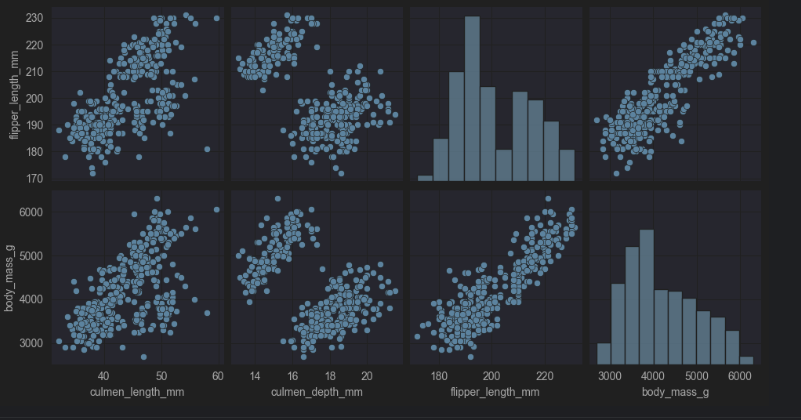


Visualisasi diatas menunjukkan banyak nya spesies dari setiap penguin yang ada didalam Data. Peringkat 1 dari populasi penguin terbanyak diraih oleh penguin Adelie dengan 146 penguin, Sedangkan Gentoo ada di urutan kedua dengan 119 Penguin, Dan Chinstrap adalah penguin dengan populasi yang paling sedikit dengan 68 penguin.



Visualisasi diatas ini menunjukkan keseluruhan tempat tinggal spesies penguin. Pulau Biscoe adalah Pulau dengan populasi spesies penguin terbanyak yang mencapai 163 penguin. Diikuti oleh pulau Dream dengan populasi sebanyak 123 penguin dan Pulau Torgersen berada di urutan terakhir dengan 47 penguin.





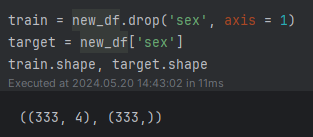
Data ini menunjukkan Distribusi dan hubungan antar variabel features, yang mana visualisasi ini adalah campuran dari scatterplot dan barplot. untuk Barplot, features yang digunakan adalah features yang sama dari sumbu X dan sumbu Y.

**Supervised Learning**

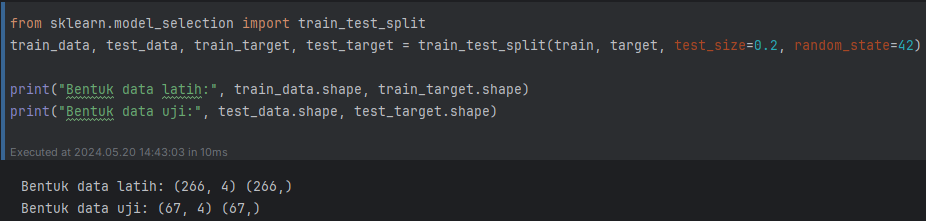
* Inisiasi Train dan Target

Kami menggunakan Cross Validation karena jumlah data yang terlalu sedikit. Berikut adalah inisialisasi nya

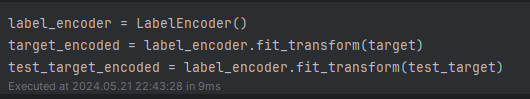
   
Lipatan yang dipakai adalah 10 lipatan, dengan begitu data yang didapat untuk diolah lebih banyak dari data yang asli. Lalu untuk train dan target yang akan diolah, memakai features ‘Sex’ agar bisa dibuat ROC nya



Data train dan target dibagi 2 dengan menggunakan train\_test\_split



Tujuan data dibagi adalah untuk menilai seberapa baik model yang telah dilatih akan bekerja pada data yang belum pernah dilihat sebelumnya. Dan untuk mengidentifikasi dan mencegah overfitting di mana model menjadi terlalu sesuai dengan data latih dan kehilangan kemampuan untuk generalisasi. Dan kami juga melakukan Label Encoder pada target agar bisa dievaluasi dengan menggunakan ROC dan Confusion Matrix nantinya.



* Random Forest Classifier

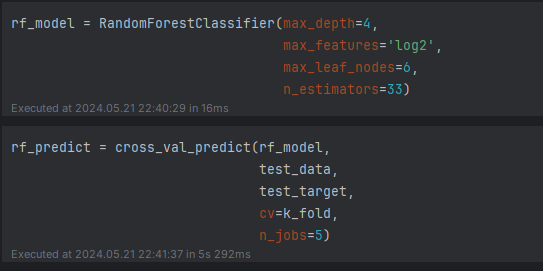
Pertama kami Menginisiasi parameter yang akan dicari dan digunakan berdasarkan kinerja akurasi terbaik nya. Kami menggunakan BayesSearchCV agar lebih cepat dalam pencarian parameter dengan akurasi terbaiknya.

Berikut adalah Hyperparameter yang akan dicari

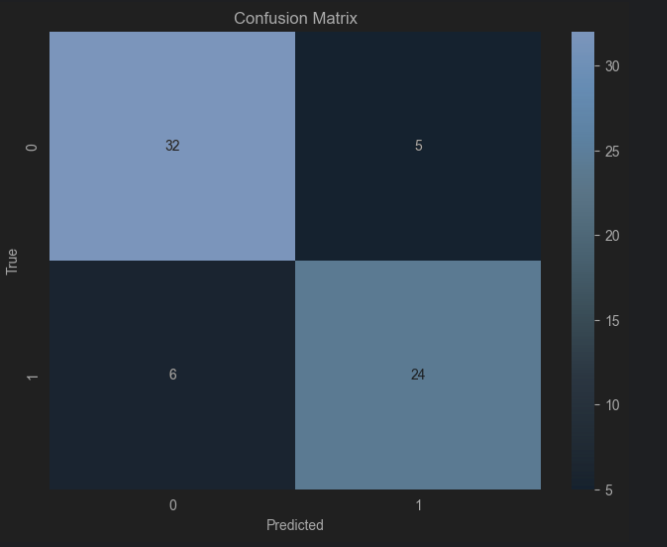
* n\_estimators = Jumlah pohon (trees) dalam ensemble.
* max\_features = Jumlah fitur yang akan dipertimbangkan untuk setiap pembagian (split) di pohon.
* max\_depth = Kedalaman maksimum pohon.
* max\_leaf\_nodes = Jumlah maksimum node daun pada setiap pohon.



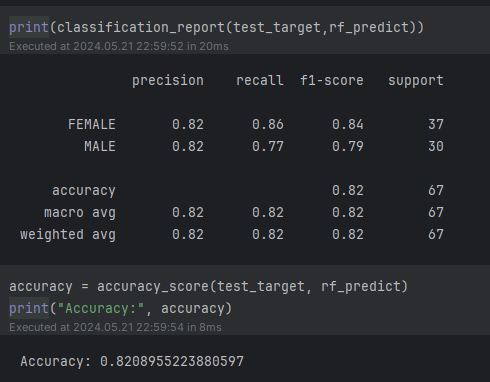
Lalu kami menginisiasi modeling dari Random Forest menggunakan parameter yang sudah dicari dan ditemukan berdasarkan akurasi terbaiknya, Lalu kami juga membuat prediksi berdasarkan modelling yang dibuat dengan menggunakan cross validation



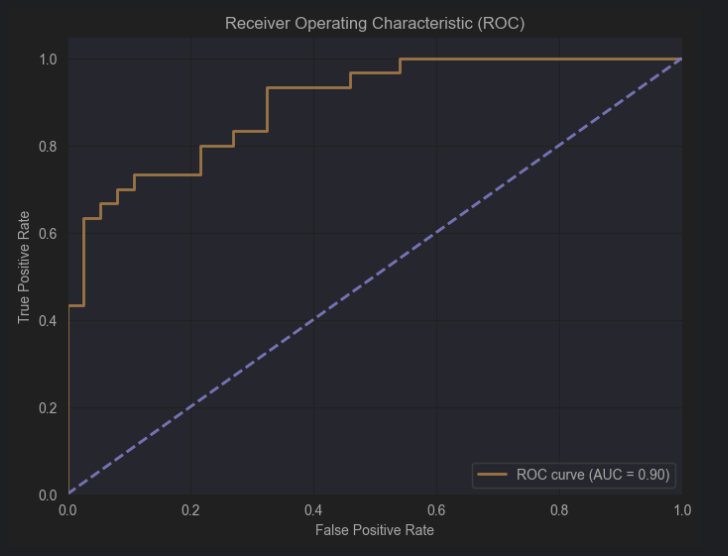
Lalu kami memvisualisasikan hasil prediksi dengan menggunakan Confusion Matrix



Yang bisa diambil dari visualisasi diatas adalah Nilai True Positive (0,0) dan True Negative (1,1) yang mana nilai True Positive harus lebih besar dari False Positive (1,0) dan True Negative harus lebih besar dari False Negative (0,1).



Lalu untuk skor akurasi, menggunakan method classification\_report untuk menampilkan Precision, Recall dan f1\_score. Menggunakan Method data Target dan prediksi dari modeling. Dan kami juga menampilkan Skor Akurasi

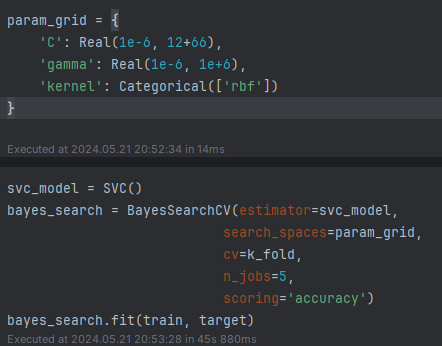


Untuk Visualisasi ROC nya seperti diatas dan Akurasi AUC nya adalah 0.90

* Support Vector Classifier

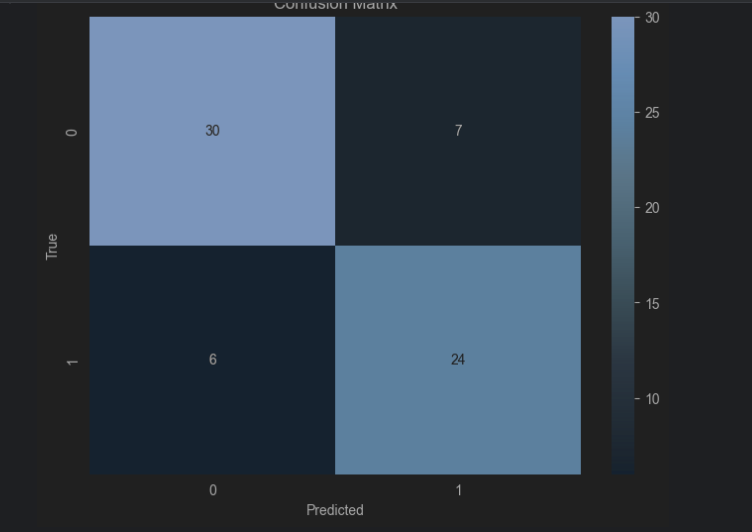
Parameter yang akan dicari adalah

1. C = Parameter regularisasi
2. gamma = Parameter kernel koefisien untuk kernel RBF
3. kernel = Jenis kernel yang digunakan dalam algoritma SVM.

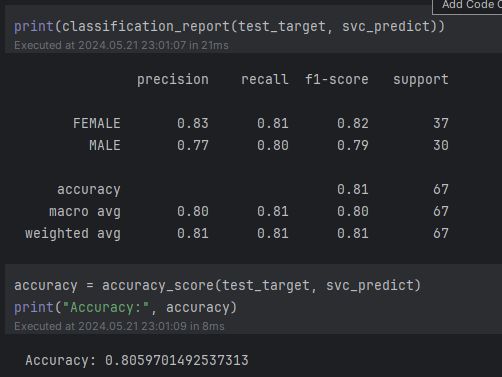


Lalu ditemukan Hyperparameter terbaik nya, berikut adalah hasilnya

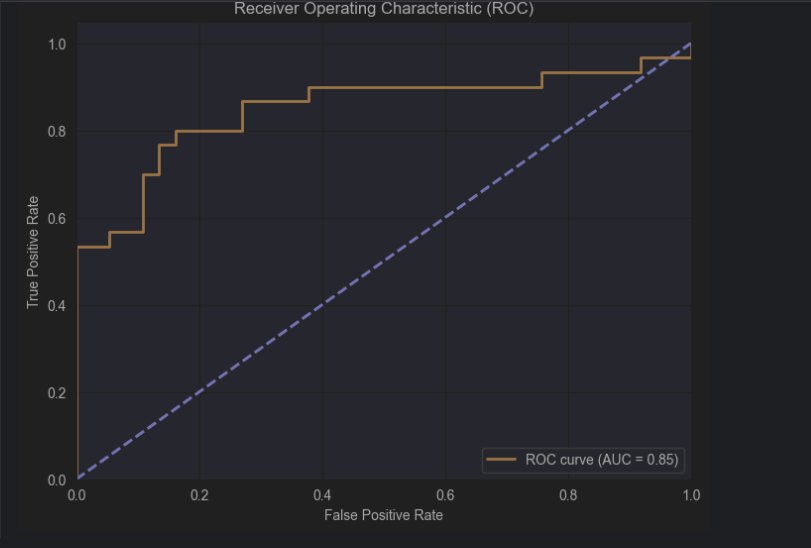
 Lalu kami melakukan permodelan dengan Support Vector Classifier menghasilkan Confusion Matrix sebagai berikut



Lalu melakukan evaluasi skor dari permodelan SVC

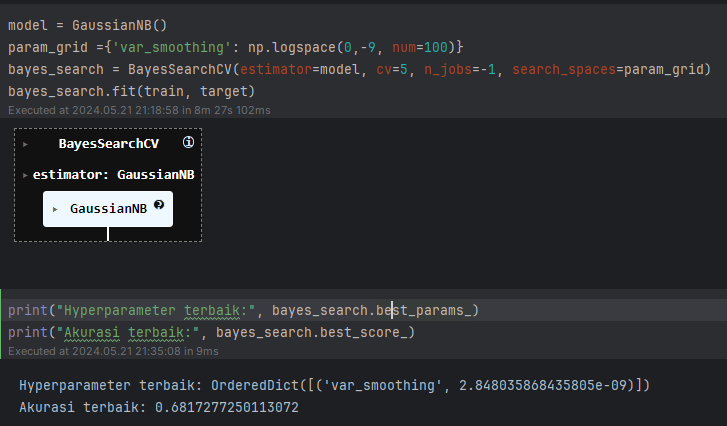


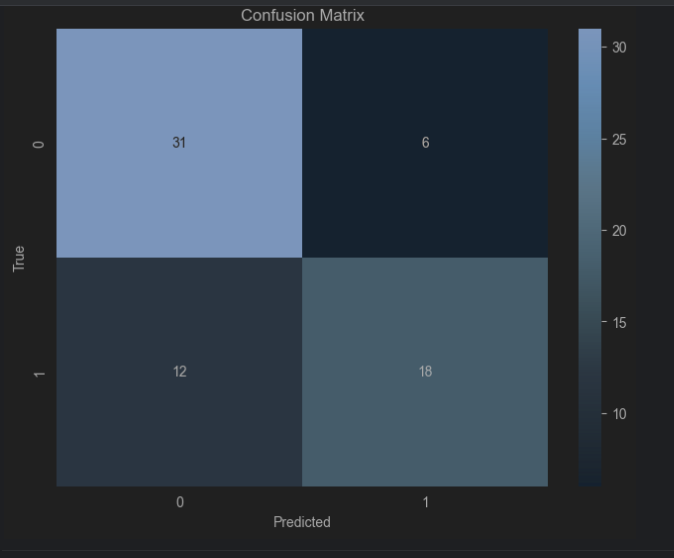
Lalu untuk visualisasi hasil evaluasi dengan menggunakan ROC dan AUC adalah sebagai berikut



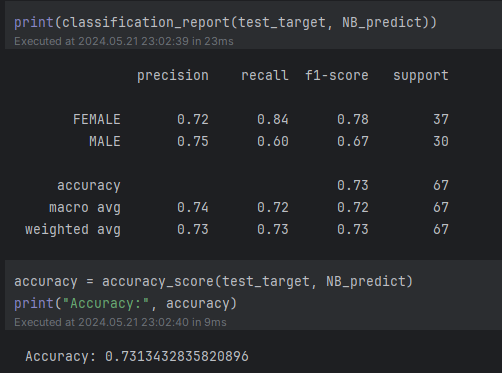
* Naive Bayes

Pencarian Parameter terbaik Naive Bayes adalah var\_smoothing yang mana var\_smoothing adalah menambahkan jumlah kecil ke varians setiap fitur, yang dapat membantu dengan stabilitas numerik.

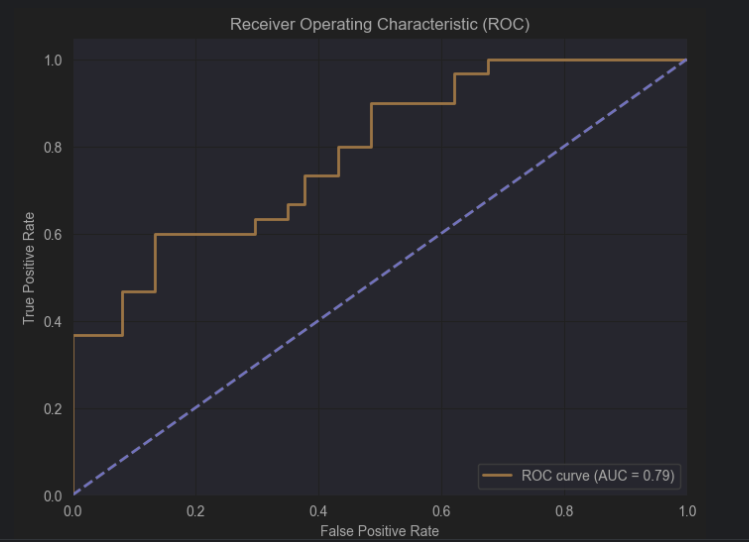




Lalu untuk Confusion nya seperti diatas, dan untuk Akurasi dari evaluasi model nya adalah sebagai berikut



Untuk ROC dan AUC nya adalah sebagai berikut



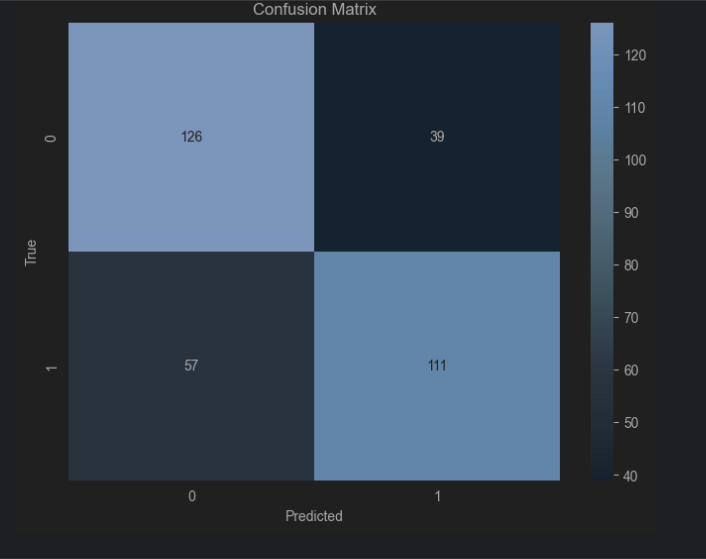
Dengan nilai AUC = 0.82

* Artificial Neural Network

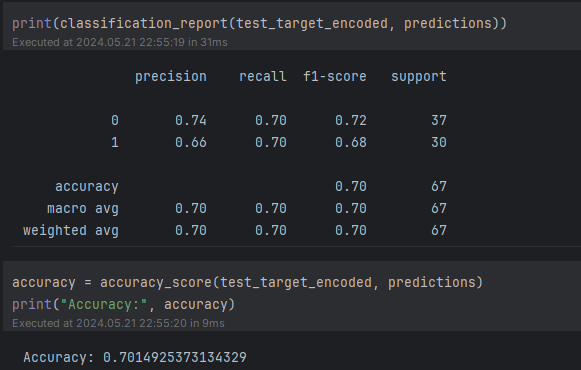
Dalam menggunakan ANN, Kami membuat Neural network sebanyak 600 pada inputan nya, lalu 300 untuk layer kedua dan 100 untuk layer ketiga dengan menggunakan activation Relu, Lalu untuk outputnya menggunakan sigmoid karena outputnya berupa biner. Lalu model compile nya menggunakan Optimizer ADAM lalu membuat evaluasi modelnya



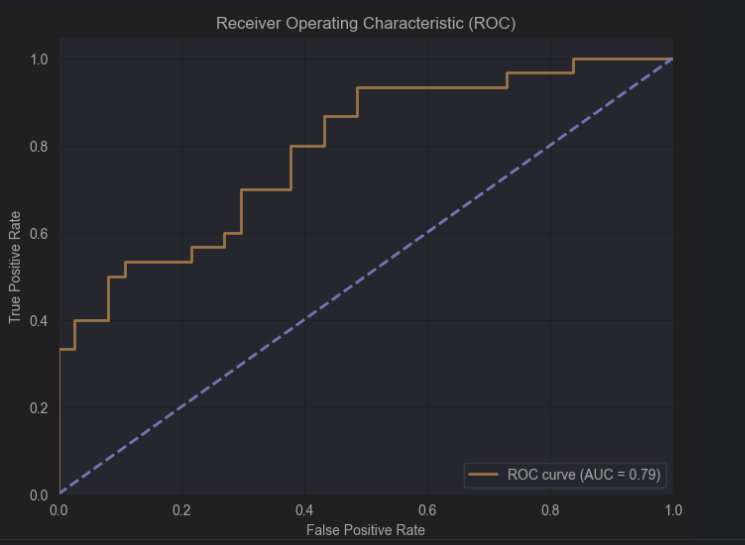
Lalu untuk Confusion Matrix sebagai berikut



Dan berikut adalah Akurasi Skor untuk evaluasi model ANN



Dan berikut adalah ROC dan AUC nya



AUC dari ANN adalah 0.77