

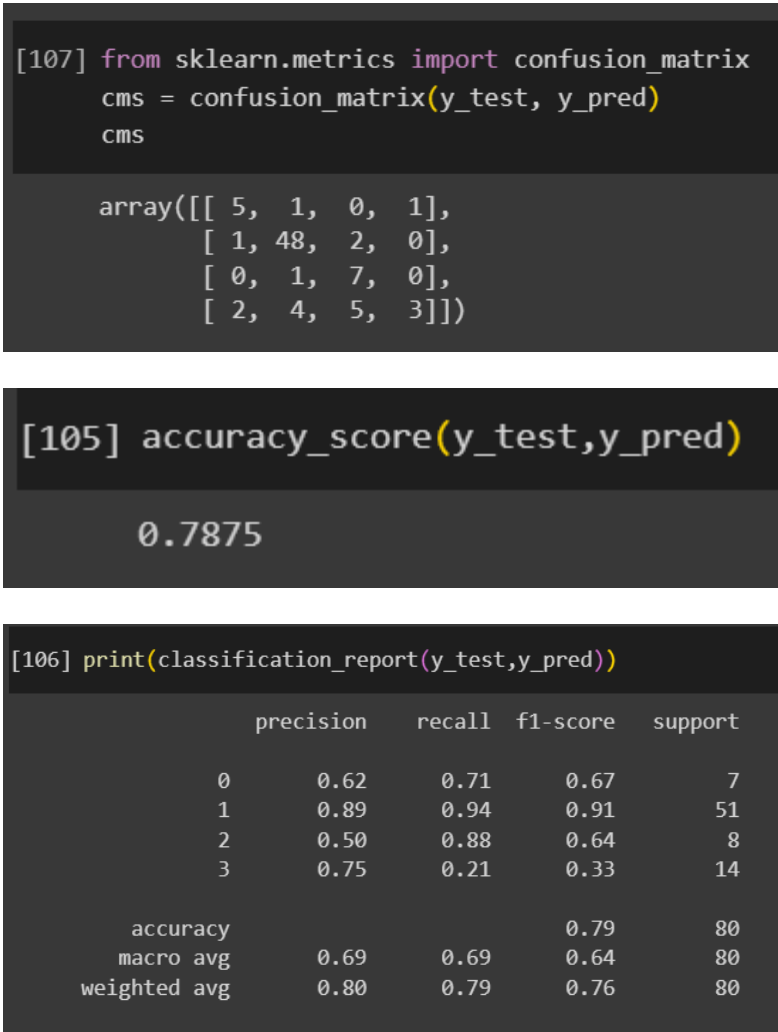
## Project Development Phase Model Performance Test

Date	21 <sup>st</sup> November 2023
Team ID	Team-593183
Project Name	Car Purchase Prediction using ML
Maximum Marks	10 Marks

### Model Performance Testing:

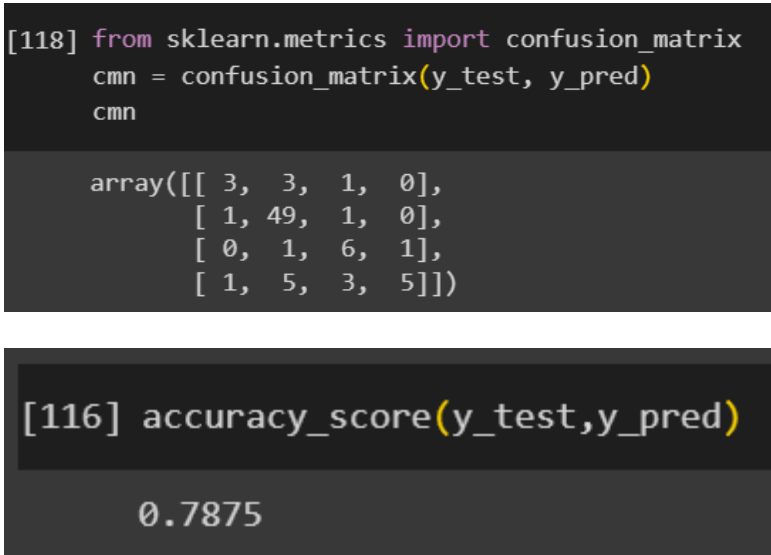
In this project we have obtained highest accuracy of 78.75% using two different classification models; i.e.; **Support Vector Classifier** and **Naive Bayes Classifier**.

### Support Vector Classifier:

Sl. No.	Parameter	Values	Screenshot
1.	Metrics	<b>Classification Model:</b>  <b>Confusion Matrix</b>  <b>Accuracy Score</b>  <b>Classification Report</b>	 <pre>[107] from sklearn.metrics import confusion_matrix       cms = confusion_matrix(y_test, y_pred)       cms</pre> <pre>array([[ 5,  1,  0,  1],        [ 1, 48,  2,  0],        [ 0,  1,  7,  0],        [ 2,  4,  5,  3]])</pre> <pre>[105] accuracy_score(y_test,y_pred)</pre> <pre>0.7875</pre> <pre>[106] print(classification_report(y_test,y_pred))</pre> <pre>               precision    recall  f1-score   support       0       0.62       0.71       0.67         7      1       0.89       0.94       0.91        51      2       0.50       0.88       0.64         8      3       0.75       0.21       0.33        14   accuracy          0.79  macro avg         0.69  weighted avg      0.80</pre>

2.	Tune the Model	<div data-bbox="606 297 1249 683"> <h3>4. Support Vector Classification</h3> <pre>[100] from sklearn.svm import SVC       SVC_model=SVC(kernel='rbf',random_state=0)       SVC_model.fit(x_train,y_train)</pre> <div data-bbox="681 394 874 456"> <div>SVC</div> <div>SVC(random_state=0)</div> </div> <pre> y_pred=SVC_model.predict(x_test) y_pred array([0, 2, 1, 2, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 3, 1, 1,        0, 3, 1, 0, 1, 1, 2, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 0, 1, 1, 0,        1, 2, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 1, 2, 1, 1, 2,        1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 0, 1, 1])</pre> </div> <div data-bbox="606 719 1249 943"> <pre>[102] y_test       209    1       290    3        33    1       210    2        93    1         --       246    1       227    3       369    0       176    1       289    3       Name: purchased_car, length: 80, dtype: int64  [103] SVC_model.predict([[1,1,23000]])  /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but SVC was fitted with feature names   warnings.warn( array([1])  [104] from sklearn.metrics import accuracy_score,classification_report  [105] accuracy_score(y_test,y_pred)  0.7875</pre> </div>
----	----------------	--

## Naive Bayes Classifier:

Sl. No.	Parameter	Values	Screenshot
1.	Metrics	<b>Classification Model:</b>  Confusion Matrix         Accuracy Score	 <pre>[118] from sklearn.metrics import confusion_matrix       cmn = confusion_matrix(y_test, y_pred)       cmn  array([[ 3,  3,  1,  0],        [ 1, 49,  1,  0],        [ 0,  1,  6,  1],        [ 1,  5,  3,  5]])</pre> <pre>[116] accuracy_score(y_test,y_pred)  0.7875</pre>

		Classification Report	<pre>[117] print(classification_report(y_test,y_pred))</pre> <table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.60</td><td>0.43</td><td>0.50</td><td>7</td></tr><tr><td>1</td><td>0.84</td><td>0.96</td><td>0.90</td><td>51</td></tr><tr><td>2</td><td>0.55</td><td>0.75</td><td>0.63</td><td>8</td></tr><tr><td>3</td><td>0.83</td><td>0.36</td><td>0.50</td><td>14</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.79</td><td>80</td></tr><tr><td>macro avg</td><td>0.71</td><td>0.62</td><td>0.63</td><td>80</td></tr><tr><td>weighted avg</td><td>0.79</td><td>0.79</td><td>0.77</td><td>80</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.60	0.43	0.50	7	1	0.84	0.96	0.90	51	2	0.55	0.75	0.63	8	3	0.83	0.36	0.50	14	accuracy			0.79	80	macro avg	0.71	0.62	0.63	80	weighted avg	0.79	0.79	0.77	80
	precision	recall	f1-score	support																																							
0	0.60	0.43	0.50	7																																							
1	0.84	0.96	0.90	51																																							
2	0.55	0.75	0.63	8																																							
3	0.83	0.36	0.50	14																																							
accuracy			0.79	80																																							
macro avg	0.71	0.62	0.63	80																																							
weighted avg	0.79	0.79	0.77	80																																							
2.	Tune the Model	Validation Method	<div><h3>5. Naive Bayes</h3><pre>[112] from sklearn.naive_bayes import GaussianNB NB_model = GaussianNB() NB_model.fit(x_train,y_train)</pre><div><div>▼ GaussianNB</div><div>GaussianNB()</div></div><pre>y_pred=NB_model.predict(x_test) y_pred</pre><pre>array([0, 3, 1, 2, 1, 1, 2, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 3, 1, 1,         1, 3, 1, 0, 1, 1, 2, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1,         1, 2, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 1, 1, 2,         1, 2, 1, 2, 2, 1, 1, 1, 1, 1, 2, 0, 1, 1])</pre><div><div>y_test</div><div><pre>209 1 208 3 33 1 210 2 93 1 ... 246 1 227 3 369 0 176 1 289 3 Name: purchased_car, length: 80, dtype: int64</pre></div></div><pre>[115] NB_model.predict([[1,51,23000]])</pre><div><div>/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but GaussianNB was fitted with feature names warnings.warn( array([3])</div><pre>[116] accuracy_score(y_test,y_pred)</pre><pre>0.7875</pre></div></div>																																								