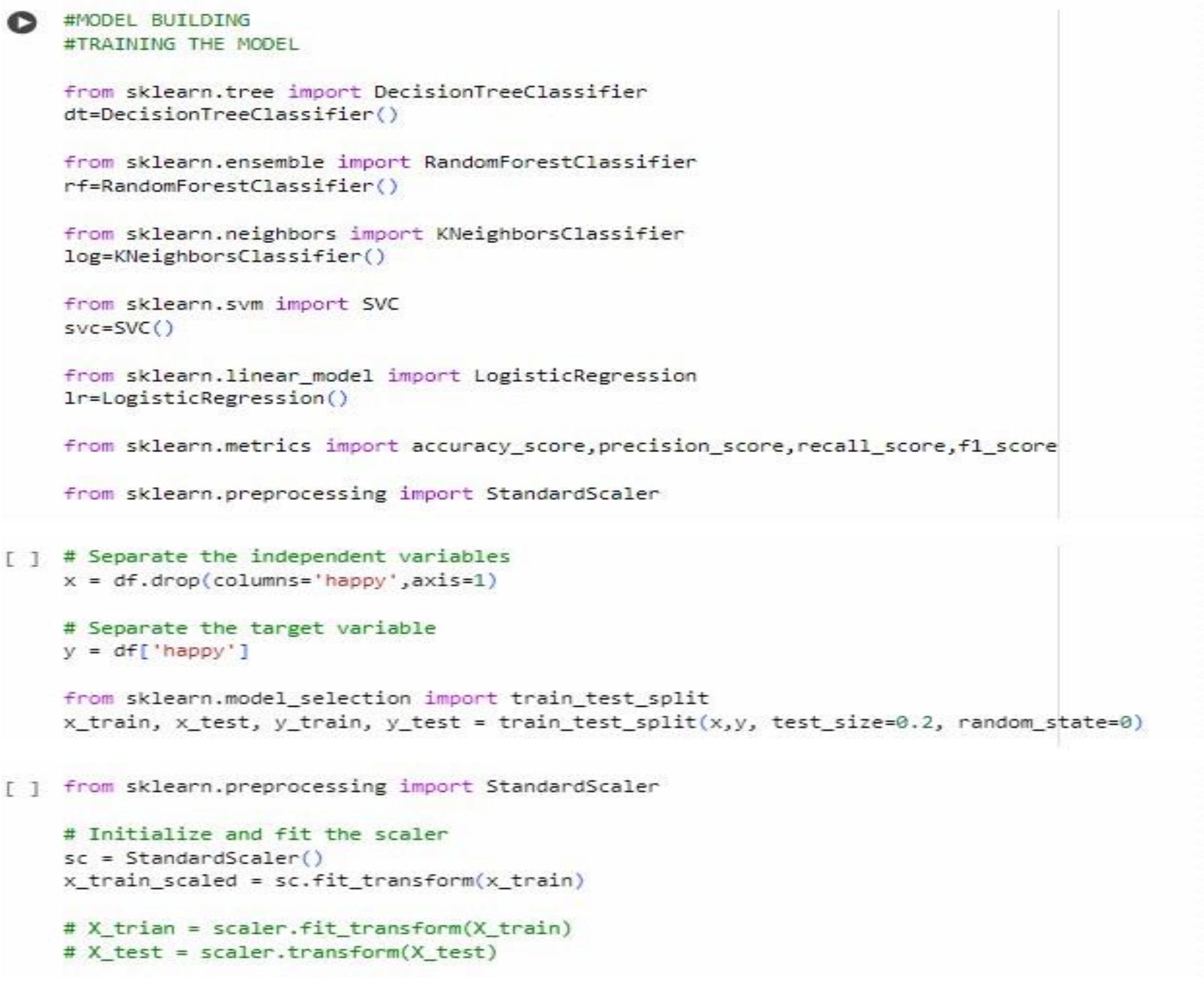
# Model Development Phase Template

| | |
|---|---|
| Date | 21 June 2024 |
| Team ID | 739998 |
| Project Title | Eudaimonia Engine: Machine Learning Delving into Happiness Classification |
| Maximum Marks | 4 Marks |

**Initial Model Training Code, Model Validation and Evaluation Report**

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

**Initial Model Training Code:**



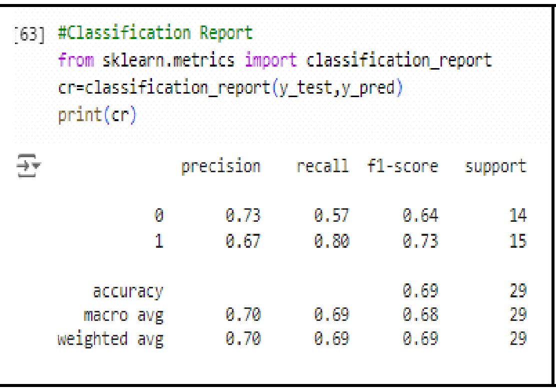
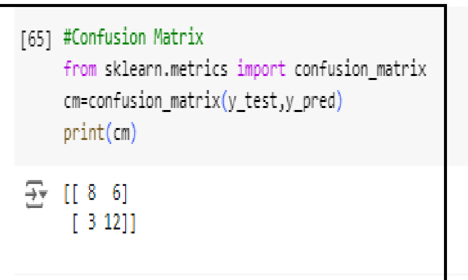
```
#MODEL BUILDING
#TRAINING THE MODEL

from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()

from sklearn.ensemble import RandomForestClassifier
rf=RandomForestClassifier()

from sklearn.neighbors import KNeighborsClassifier
log=KNeighborsClassifier()

from sklearn.svm import SVC
svc=SVC()

from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()

from sklearn.metrics import accuracy_score,precision_score,recall_score,f1_score

from sklearn.preprocessing import StandardScaler
```

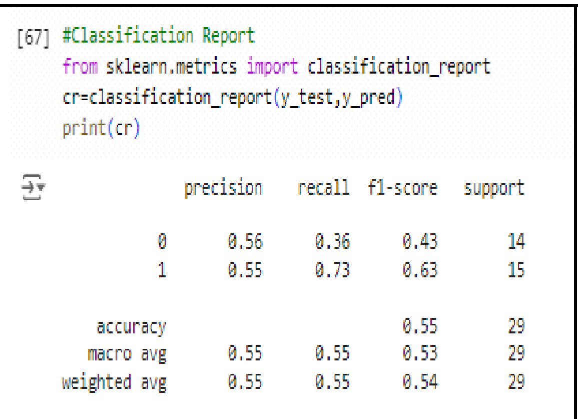
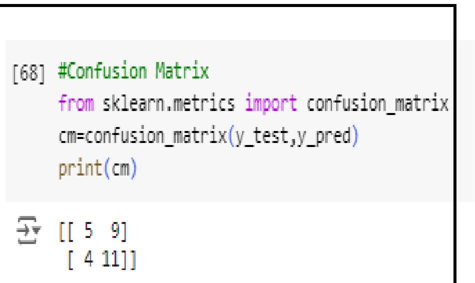
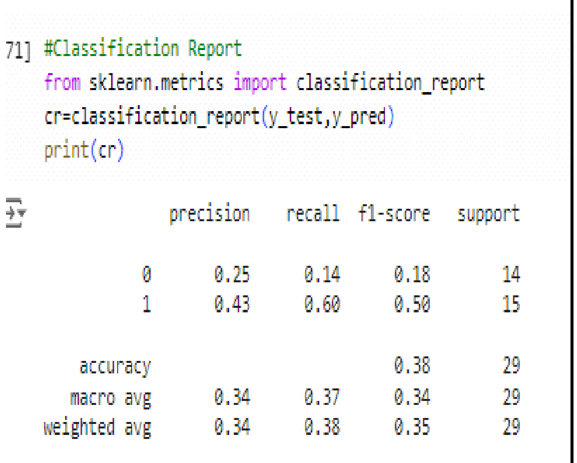
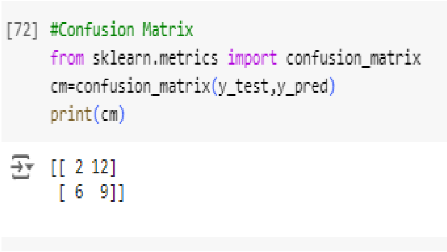
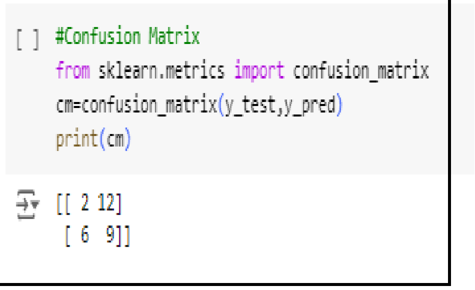
```
[ ] # Separate the independent variables
    x = df.drop(columns='happy',axis=1)

    # Separate the target variable
    y = df['happy']

    from sklearn.model_selection import train_test_split
    x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2, random_state=0)
```

```
[ ] from sklearn.preprocessing import StandardScaler

    # Initialize and fit the scaler
    sc = StandardScaler()
    x_train_scaled = sc.fit_transform(x_train)

    # X_trian = scaler.fit_transform(X_train)
    # X_test = scaler.transform(X_test)
```

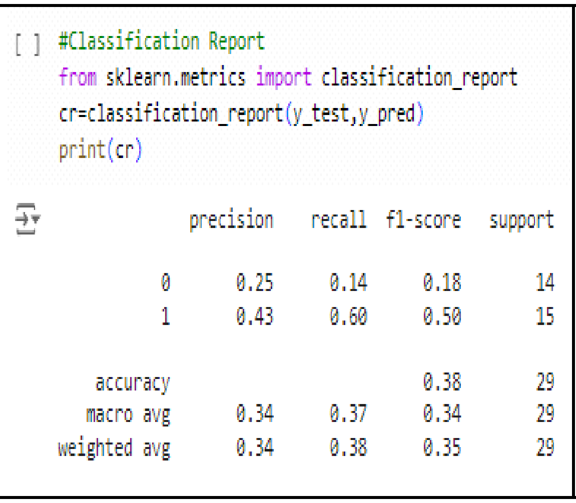
## Model Validation and Evaluation Report:

| Model | Classification Report | F1 Sco r e | Confusion Matrix |
|---|---|---|---|
| | | | |

|  |  |  |  |
|---|---|---|---|
| Decision Tree Model | <br>[63] #Classification Report<br>from sklearn.metrics import classification_report<br>cr=classification_report(y_test,y_pred)<br>print(cr)<br>precision recall f1-score support<br>0 0.73 0.57 0.64 14<br>1 0.67 0.80 0.73 15<br>accuracy 0.69 29<br>macro avg 0.70 0.69 0.68 29<br>weighted avg 0.70 0.69 0.69 29 | 69% | <br>[65] #Confusion Matrix<br>from sklearn.metrics import confusion_matrix<br>cm=confusion_matrix(y_test,y_pred)<br>print(cm)<br>[[ 8 6]<br>[ 3 12]] |

|  |  |  |  |
|---|---|---|---|
| Random Forest Model | <br>[67] #Classification Report<br>from sklearn.metrics import classification_report<br>cr=classification_report(y_test,y_pred)<br>print(cr)<br>precision recall f1-score support<br>0 0.56 0.36 0.43 14<br>1 0.55 0.73 0.63 15<br>accuracy 0.55 29<br>macro avg 0.55 0.55 0.53 29<br>weighted avg 0.55 0.55 0.54 29 | 55 % | <br>[68] #Confusion Matrix<br>from sklearn.metrics import confusion_matrix<br>cm=confusion_matrix(y_test,y_pred)<br>print(cm)<br>[[ 5 9]<br>[ 4 11]] |
| KNN | <br>71] #Classification Report<br>from sklearn.metrics import classification_report<br>cr=classification_report(y_test,y_pred)<br>print(cr)<br>precision recall f1-score support<br>0 0.25 0.14 0.18 14<br>1 0.43 0.60 0.50 15<br>accuracy 0.38 29<br>macro avg 0.34 0.37 0.34 29<br>weighted avg 0.34 0.38 0.35 29 | 38% | <br>[72] #Confusion Matrix<br>from sklearn.metrics import confusion_matrix<br>cm=confusion_matrix(y_test,y_pred)<br>print(cm)<br>[[ 2 12]<br>[ 6 9]] |
| SVC |  | 38% | <br>[ ] #Confusion Matrix<br>from sklearn.metrics import confusion_matrix<br>cm=confusion_matrix(y_test,y_pred)<br>print(cm)<br>[[ 2 12]<br>[ 6 9]] |

<table>
<tr>
<td></td>
<td>

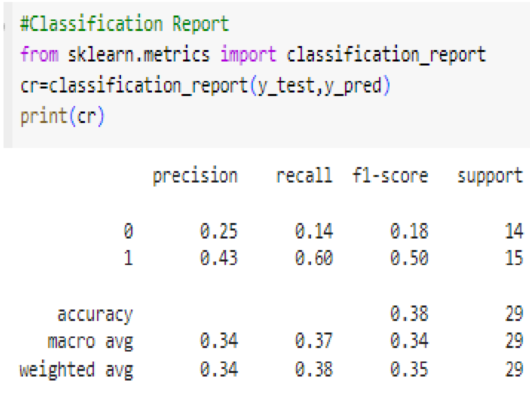


```
[ ] #Classification Report
    from sklearn.metrics import classification_report
    cr=classification_report(y_test,y_pred)
    print(cr)
```

```
              precision    recall  f1-score   support

           0       0.25      0.14      0.18        14
           1       0.43      0.60      0.50        15

    accuracy                           0.38        29
   macro avg       0.34      0.37      0.34        29
weighted avg       0.34      0.38      0.35        29
```

</td>
<td></td>
<td></td>
</tr>
</table>

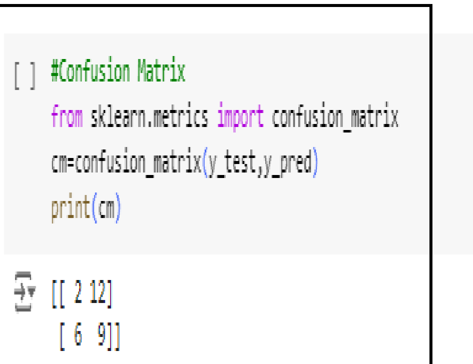
<table>
<tr>
<td>Logistic Regression Model</td>
<td>



```
#Classification Report
from sklearn.metrics import classification_report
cr=classification_report(y_test,y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       0.25      0.14      0.18        14
           1       0.43      0.60      0.50        15

    accuracy                           0.38        29
   macro avg       0.34      0.37      0.34        29
weighted avg       0.34      0.38      0.35        29
```

</td>
<td>38 %</td>
<td>



```
[ ] #Confusion Matrix
    from sklearn.metrics import confusion_matrix
    cm=confusion_matrix(y_test,y_pred)
    print(cm)
```

```
[[ 2 12]
 [ 6  9]]
```

</td>
</tr>
</table>