In [ ]:
```
#Logistic Regression and One versus One
#his dataset simulates a financial context, designed to reflect customer accou
#It comprises featuressuch as Age of Account (in years), Number of Transactior
#3. Your goal is to design an ML classifier using OVO with logistic regressior
```

In [2]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, cla
from sklearn.metrics import confusion_matrix
from sklearn.multiclass import OneVsOneClassifier
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
data = pd.read_csv('synthetic_FINANCE.csv')
data
```

Out[4]:

	Unnamed: 0	Age_of_Account_years	Number_of_Transactions_last_month	Average_Transactior
0	0	7	12	
1	1	20	55	
2	2	29	13	
3	3	15	23	
4	4	11	89	
...	...	...	...	
695	695	17	49	
696	696	7	71	
697	697	24	81	
698	698	23	84	
699	699	5	49	

700 rows × 7 columns

In [15]:
```
X = data[['Number_of_Transactions_last_month', 'Average_Transaction_Value', 'C
y = data['Risk_Class']
```

In [16]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, rando
```

In [17]:
```
logistic_regression_model = LogisticRegression()
ovo_classifier = OneVsOneClassifier(logistic_regression_model)
```

In [18]:
```
ovo_classifier.fit(X_train, y_train)
```

Out[18]:
```
OneVsOneClassifier(estimator=LogisticRegression())
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [19]:
```
y_pred = ovo_classifier.predict(X_test)
```

In [20]:
```
#Calculating the evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='macro')
recall = recall_score(y_test, y_pred, average='macro')
```

In [21]:
```
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
```

```
Accuracy: 0.5285714285714286
Precision: 0.49845540786388476
Recall: 0.4816339180561582
```

In [22]:
```
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

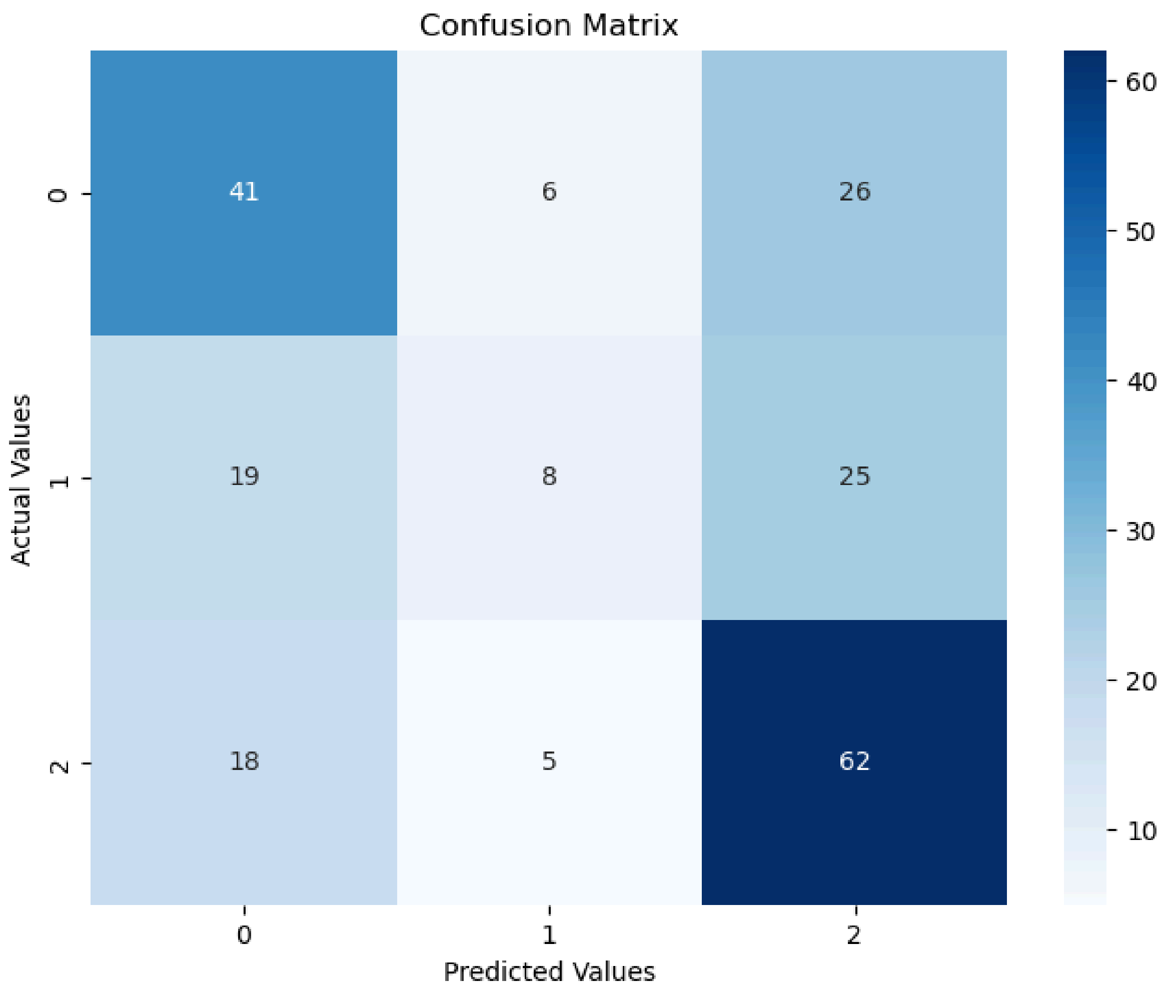
```

Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.56      0.54        73
           1       0.42      0.15      0.23        52
           2       0.55      0.73      0.63        85

    accuracy                           0.53       210
   macro avg       0.50      0.48      0.46       210
weighted avg       0.51      0.53      0.50       210
```

In [23]:
```
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d", cmap='Blues')
plt.title('Confusion Matrix')
plt.ylabel('Actual Values')
plt.xlabel('Predicted Values')
plt.show()
```



In [25]:
```
Number_of_Transactions_last_month = int(input("Enter the number of transactior
Average_Transaction_Value = float(input("Enter the average transaction value:
Account_Balance = int(input("Enter the credit score: "))
input_data = [[Number_of_Transactions_last_month, Average_Transaction_Value, A
outcome = ovo_classifier.predict(input_data)
print(f"Predicted risks:{outcome}")
```

```
Enter the number of transactions in last month: 56
Enter the average transaction value: 6785.67
Enter the credit score: 475
Predicted risks:[2]
```

In [ ]:
```
#From my point of view Precision looks very important for this analysis, becau
#In the live industries, false positives can lead to unnecessary problems like
#So here the high precision indicates that the model is accurately identifying

#Evaluating the result:

#Precision
#Classificiation 0 has the highest precision of 53% , hence the accounts corre
#Classificiation 2 has the precision of 55%, hence the accounts correctly pred
#Classification 1 has the lowest precision of 42%, hence only this percentage

#Recall
#Classification 2 has the highest recall of 73% of 2 accounts.
#Classification 0 has the recall of 56% of 0 accounts.
#Classification 1 has the lowest recall of 85% of 1 accounts.

#F1-score
#Classification 2 has the highest F1-score of 63%, which represents a balance
#Classification 0 has the F1-score of 54%, indicating a good balance between p
#Classification 1 has the lowest F1-score of 23%, indicating a poor balance be


#Classification 0 and Class 2 prediction shows that a balanced performance bet
#Classification 1 has lower precision, recall, and F1-score compared to the ot
```