a) General information on dataset

Name of dataset used: German Traffic Sign Benchmarks

Number of classes: 5 which are {'00005', '00004', '00010', '00012', '00038'}

Labels of classes: [5, 4, 10, 12, 38]

Total number of samples: 13320

Number of samples used in training: 10020

Size: 5: 630

4: 660

10: 660

12: 690

38: 690

Number of samples used in testing: 3330

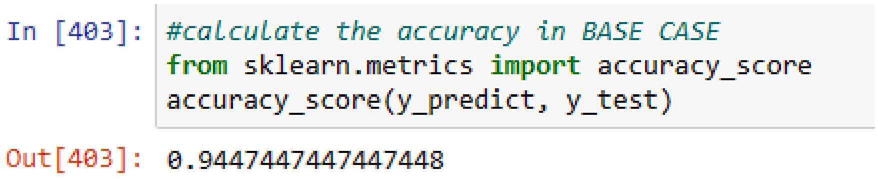
b) implementation details

Feature extraction method: HOG (Histogram of Oriented Gradients)

Number of HOG features for a single image: 144

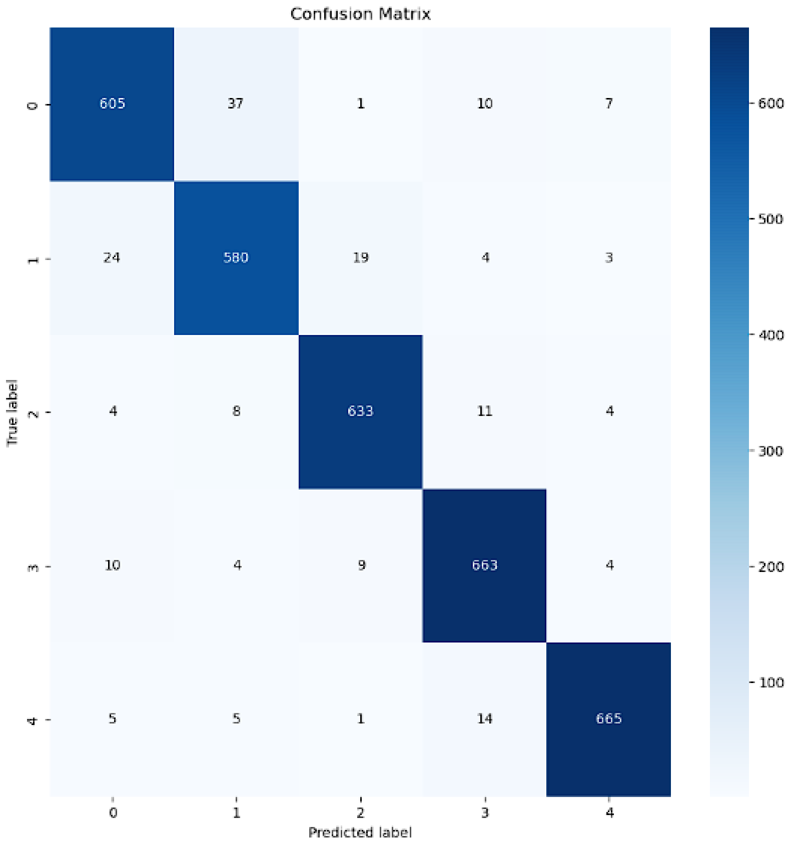
C)Result details

Logistic regression without future extraction(base case)

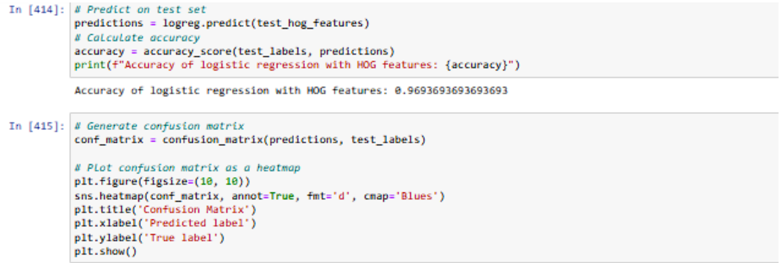


```
In [403]: #calculate the accuracy in BASE CASE
          from sklearn.metrics import accuracy_score
          accuracy_score(y_predict, y_test)

Out[403]: 0.9447447447447448
```



Logistic regression with future extraction

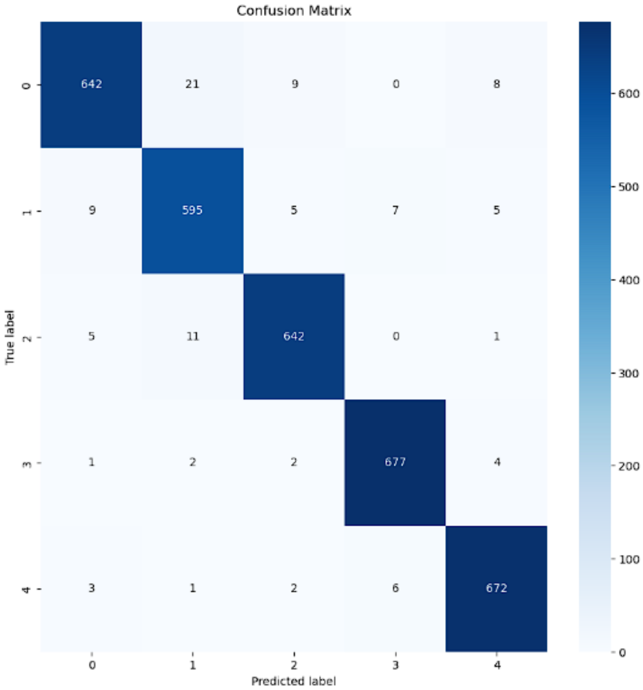


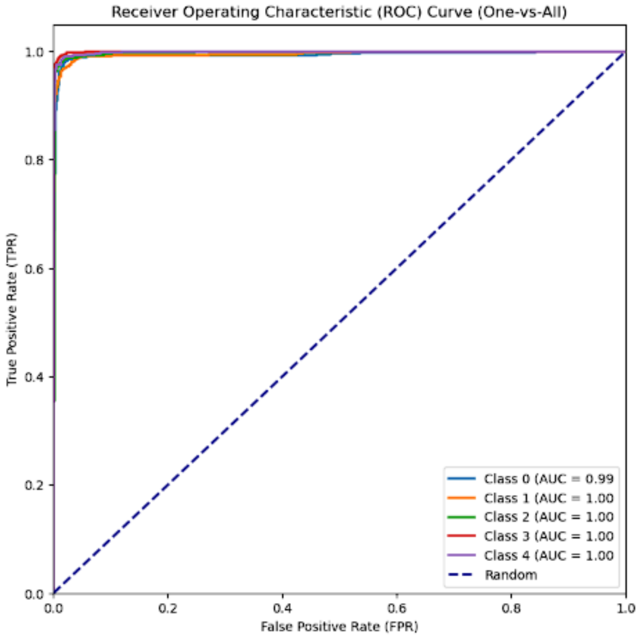
In [414]:
```
# Predict on test set
predictions = logreg.predict(test_hog_features)
# Calculate accuracy
accuracy = accuracy_score(test_labels, predictions)
print(f"Accuracy of logistic regression with HOG features: {accuracy}")
```

```
Accuracy of logistic regression with HOG features: 0.9693693693693693
```

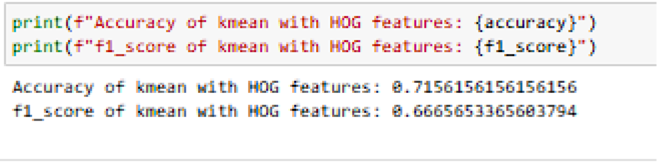
In [415]:
```
# Generate confusion matrix
conf_matrix = confusion_matrix(predictions, test_labels)

# Plot confusion matrix as a heatmap
plt.figure(figsize=(10, 10))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.show()
```





Kmean



```
print(f"Accuracy of kmean with HOG features: {accuracy}")
print(f"f1_score of kmean with HOG features: {f1_score}")
```

```
Accuracy of kmean with HOG features: 0.7156156156156156
f1_score of kmean with HOG features: 0.6665653365603794
```