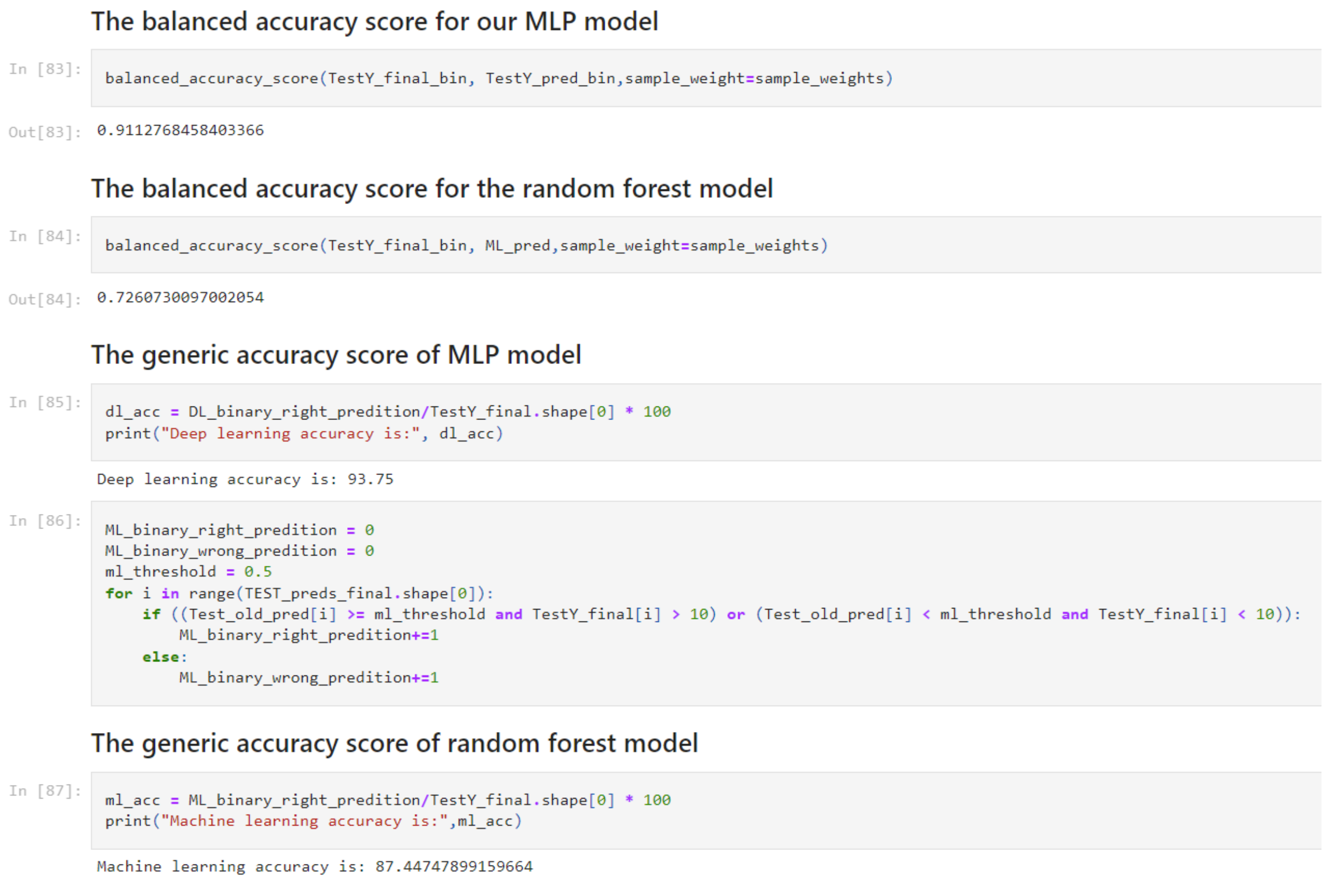
Spatio-temporal modelling of arsenic data

Experiment 1A – MLP model with oversampled TrainX dataset –

Results –



The balanced accuracy score for our MLP model

In [83]:

```
balanced_accuracy_score(TestY_final_bin, TestY_pred_bin,sample_weight=sample_weights)
```

Out[83]:

```
0.9112768458403366
```

The balanced accuracy score for the random forest model

In [84]:

```
balanced_accuracy_score(TestY_final_bin, ML_pred,sample_weight=sample_weights)
```

Out[84]:

```
0.7260730097002054
```

The generic accuracy score of MLP model

In [85]:

```
dl_acc = DL_binary_right_predition/TestY_final.shape[0] * 100
print("Deep learning accuracy is:", dl_acc)
```

```
Deep learning accuracy is: 93.75
```

In [86]:

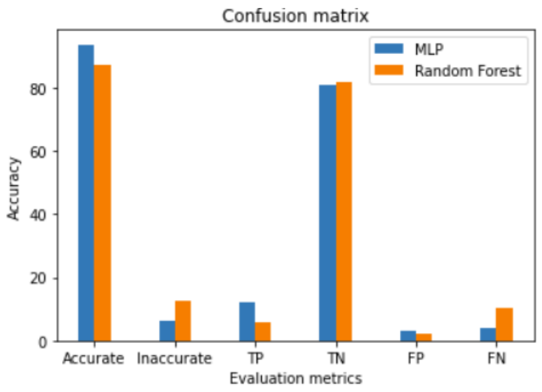
```
ML_binary_right_predition = 0
ML_binary_wrong_predition = 0
ml_threshold = 0.5
for i in range(TEST_preds_final.shape[0]):
    if ((Test_old_pred[i] >= ml_threshold and TestY_final[i] > 10) or (Test_old_pred[i] < ml_threshold and TestY_final[i] < 10)):
        ML_binary_right_predition+=1
    else:
        ML_binary_wrong_predition+=1
```

The generic accuracy score of random forest model

In [87]:

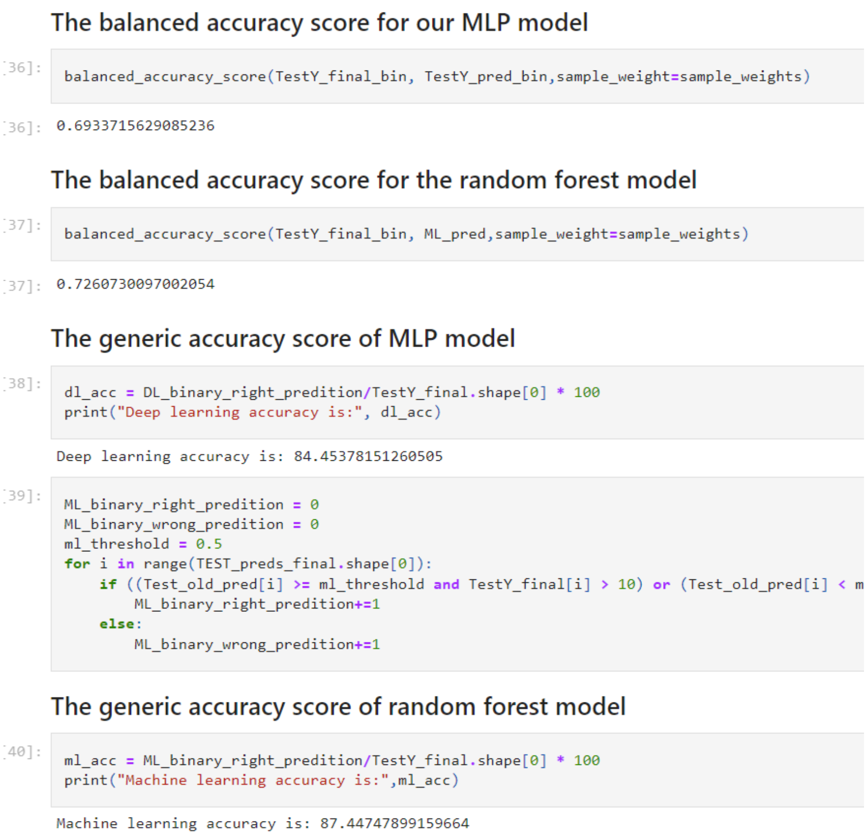
```
ml_acc = ML_binary_right_predition/TestY_final.shape[0] * 100
print("Machine learning accuracy is:",ml_acc)
```

```
Machine learning accuracy is: 87.44747899159664
```



Experiment 1B – MLP model without oversampling –

Results-



The balanced accuracy score for our MLP model

[36]:
```
balanced_accuracy_score(TestY_final_bin, TestY_pred_bin,sample_weight=sample_weights)
```

[36]:
```
0.6933715629085236
```

The balanced accuracy score for the random forest model

[37]:
```
balanced_accuracy_score(TestY_final_bin, ML_pred,sample_weight=sample_weights)
```

[37]:
```
0.7260730097002054
```

The generic accuracy score of MLP model

[38]:
```
dl_acc = DL_binary_right_predition/TestY_final.shape[0] * 100
print("Deep learning accuracy is:", dl_acc)
```

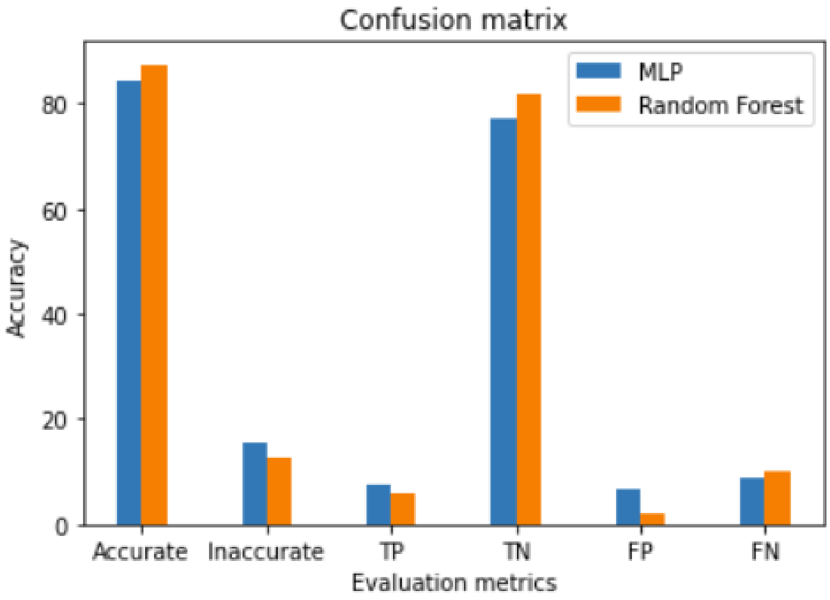
```
Deep learning accuracy is: 84.45378151260505
```

[39]:
```
ML_binary_right_predition = 0
ML_binary_wrong_predition = 0
ml_threshold = 0.5
for i in range(TEST_preds_final.shape[0]):
    if ((Test_old_pred[i] >= ml_threshold and TestY_final[i] > 10) or (Test_old_pred[i] < m
        ML_binary_right_predition+=1
    else:
        ML_binary_wrong_predition+=1
```

The generic accuracy score of random forest model

[40]:
```
ml_acc = ML_binary_right_predition/TestY_final.shape[0] * 100
print("Machine learning accuracy is:",ml_acc)
```

```
Machine learning accuracy is: 87.44747899159664
```



Experiment 2 – Graphical analysis of the dataset

	Euclidean cut-off	No. of edges	Average clustering Coeff	No. of edges in largest connected component
1	2500	8211	0.01	699
2	3500	14083	0.15	1893
3	5000	26975	0.29	10720
4	6000	30246	0.31	12526
5	7500	45866	0.40	21438
6	10000	61977	0.46	30861
7	15000	121963	0.57	64864

Balanced accuracy and Generic accuracy

Model name	Balanced Accuracy	Generic accuracy		
Random forest	69	87		
MLP without oversampling	68	84.4		
MLP with oversampling train	91	93.75		
MLP-GNN	70	85		
Embedding based GNN	69	69.2		
Binary classification model	NA	85		