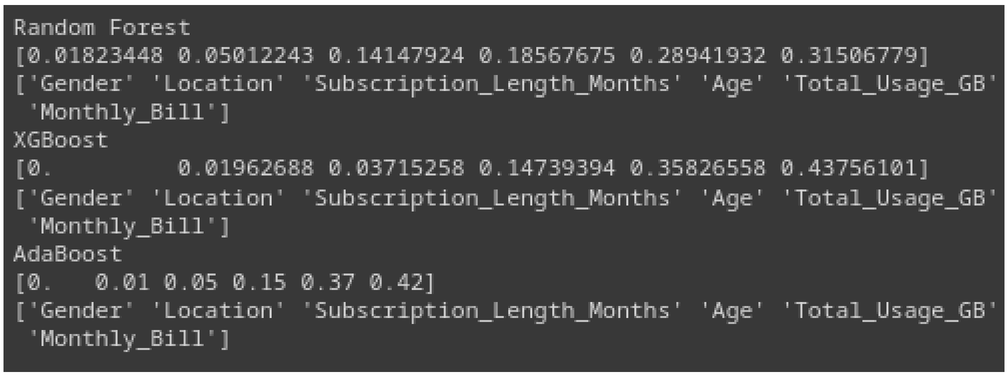
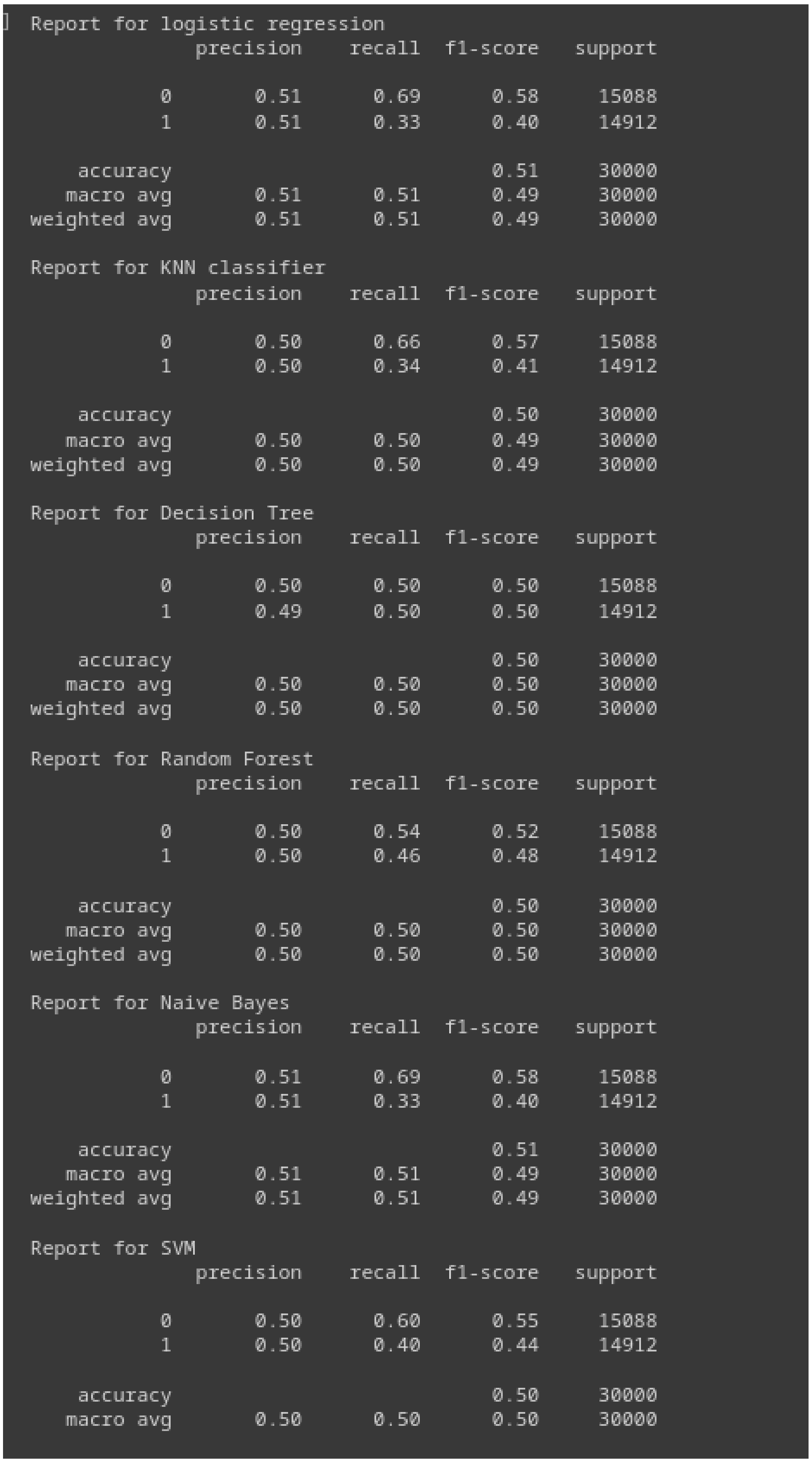
Churn Dataset Analysis

- Well balanced dataset with equal distribution of classification - no need to perform sampling
- No null values present
- No Duplicated data
- Datatypes are appropriate for all features
- No correlation between features and target variable
- Data is widely distributed - has to perform normalization or standardisation for Gradient descent algorithm to reach global minima gradually.
- No linear relation among features
- No outliers present - Boxplot
- Categorical Data - Label Encoder used for ‘Location’ and ‘Gender’ features
- ‘Name’ and ‘CustomerID’ columns has no effect on target prediction - so removing both features
- Training and Testing dataset split - 70 : 30
- Data distribution is not normal or gaussian, so all features are normalised - MinMaxScaler()
- Models tested
    - Decision Tree
    - Random Forest
    - Logistic Regression
    - KNN
    - Naive Bayes
    - SVC
    - ANN
- Cross Validation - Stratifiedkfold
- Hypertunning - RandomizedSearchCV
- Feature Importance
    - 

```
Random Forest
[0.01823448 0.05012243 0.14147924 0.18567675 0.28941932 0.31506779]
['Gender' 'Location' 'Subscription_Length_Months' 'Age' 'Total_Usage_GB'
 'Monthly_Bill']
XGBoost
[0.         0.01962688 0.03715258 0.14739394 0.35826558 0.43756101]
['Gender' 'Location' 'Subscription_Length_Months' 'Age' 'Total_Usage_GB'
 'Monthly_Bill']
AdaBoost
[0.   0.01 0.05 0.15 0.37 0.42]
['Gender' 'Location' 'Subscription_Length_Months' 'Age' 'Total_Usage_GB'
 'Monthly_Bill']
```

- Model Results



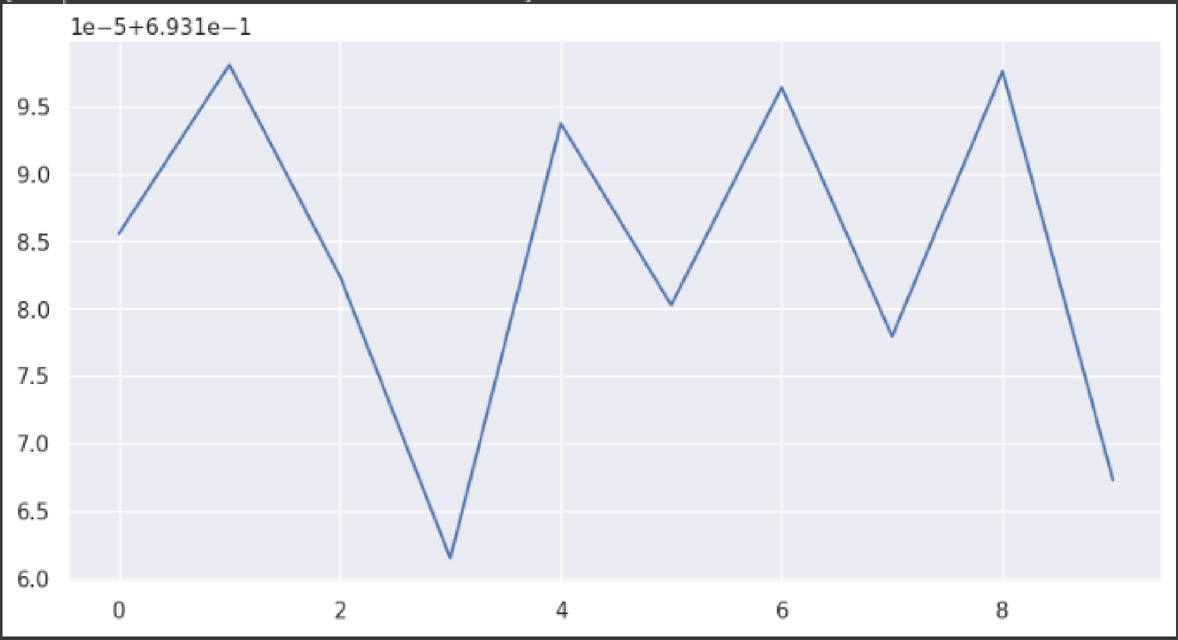
```
] Report for logistic regression
              precision    recall  f1-score   support

           0       0.51      0.69      0.58     15088
           1       0.51      0.33      0.40     14912

    accuracy                           0.51     30000
   macro avg       0.51      0.51      0.49     30000
weighted avg       0.51      0.51      0.49     30000

Report for KNN classifier
              precision    recall  f1-score   support

           0       0.50      0.66      0.57     15088
           1       0.50      0.34      0.41     14912

    accuracy                           0.50     30000
   macro avg       0.50      0.50      0.49     30000
weighted avg       0.50      0.50      0.49     30000

Report for Decision Tree
              precision    recall  f1-score   support

           0       0.50      0.50      0.50     15088
           1       0.49      0.50      0.50     14912

    accuracy                           0.50     30000
   macro avg       0.50      0.50      0.50     30000
weighted avg       0.50      0.50      0.50     30000

Report for Random Forest
              precision    recall  f1-score   support

           0       0.50      0.54      0.52     15088
           1       0.50      0.46      0.48     14912

    accuracy                           0.50     30000
   macro avg       0.50      0.50      0.50     30000
weighted avg       0.50      0.50      0.50     30000

Report for Naive Bayes
              precision    recall  f1-score   support

           0       0.51      0.69      0.58     15088
           1       0.51      0.33      0.40     14912

    accuracy                           0.51     30000
   macro avg       0.51      0.51      0.49     30000
weighted avg       0.51      0.51      0.49     30000

Report for SVM
              precision    recall  f1-score   support

           0       0.50      0.60      0.55     15088
           1       0.50      0.40      0.44     14912

    accuracy                           0.50     30000
   macro avg       0.50      0.50      0.50     30000
```

- ANN Result

```
Epoch 1/10
7000/7000 [==============================] - 16s 2ms/step - loss: 0.6932 - accuracy: 0.4997
Epoch 2/10
7000/7000 [==============================] - 16s 2ms/step - loss: 0.6932 - accuracy: 0.4977
Epoch 3/10
7000/7000 [==============================] - 18s 3ms/step - loss: 0.6932 - accuracy: 0.4993
Epoch 4/10
7000/7000 [==============================] - 14s 2ms/step - loss: 0.6932 - accuracy: 0.5010
Epoch 5/10
7000/7000 [==============================] - 14s 2ms/step - loss: 0.6932 - accuracy: 0.5009
Epoch 6/10
7000/7000 [==============================] - 14s 2ms/step - loss: 0.6932 - accuracy: 0.4998
Epoch 7/10
7000/7000 [==============================] - 14s 2ms/step - loss: 0.6932 - accuracy: 0.4999
Epoch 8/10
7000/7000 [==============================] - 14s 2ms/step - loss: 0.6932 - accuracy: 0.4981
Epoch 9/10
7000/7000 [==============================] - 14s 2ms/step - loss: 0.6932 - accuracy: 0.4976
Epoch 10/10
7000/7000 [==============================] - 16s 2ms/step - loss: 0.6932 - accuracy: 0.5002
```

```
# predicting the test data
y_pred = classifier.predict(xtest)
y_pred = (y_pred>0.5)
plt.plot(ann_model.history['loss'])
```

```
938/938 [==============================] - 3s 3ms/step
[<matplotlib.lines.Line2D at 0x7e1e30d547f0>]
```



○

- Hypertuning for Logistic and Random Forest

```
Parameters: {'solver': 'newton-cg', 'penalty': 'l2', 'C': 0.01}
Best score is 0.5005428571428572
Parameters: {'criterion': 'gini', 'max_depth': 3, 'max_features': 2, 'min_samples_leaf': 1}
Best score is 0.5032285714285715
```

○