1. **Data Cleaning**

Change Values where it have false values

```
Basic,3.0,Single,3.0
```

```
,8.0,Freelancer,Male,3
```

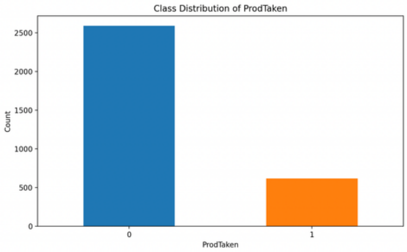
```
,Small Business,Female,3,4.0,
```

Fe Male → Female

Free Lancer → Freelancer

Unmarried → Single

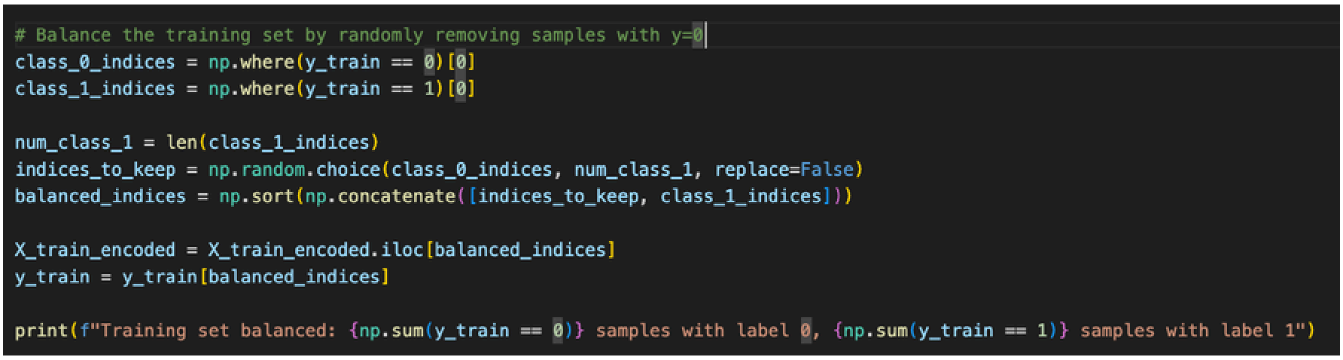
2. **Data Analysis**



We can see that the Class Imbalanced towards 0 creating biased

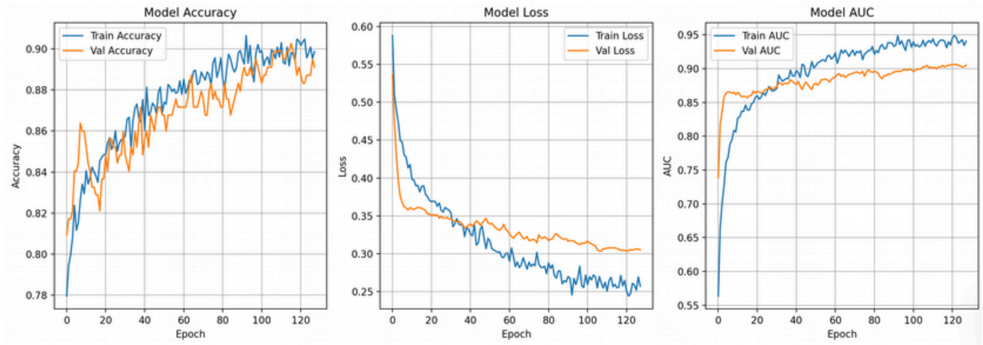
3. **Feature Engineering**

Randomly remove data values that have ProdTaken of 0 till the amount will match the len of Class 1

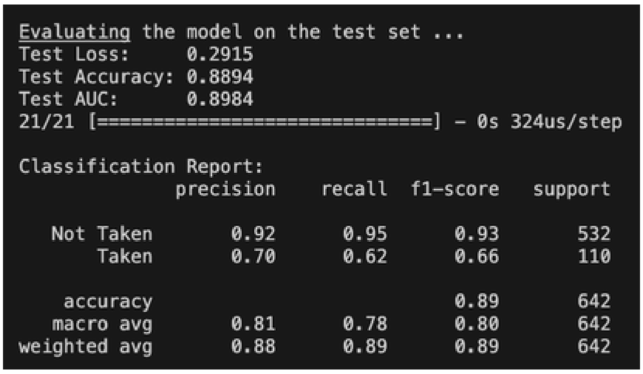


```
# Balance the training set by randomly removing samples with y=0
class_0_indices = np.where(y_train == 0)[0]
class_1_indices = np.where(y_train == 1)[0]

num_class_1 = len(class_1_indices)
indices_to_keep = np.random.choice(class_0_indices, num_class_1, replace=False)
balanced_indices = np.sort(np.concatenate([indices_to_keep, class_1_indices]))

X_train_encoded = X_train_encoded.iloc[balanced_indices]
y_train = y_train[balanced_indices]

print(f"Training set balanced: {np.sum(y_train == 0)} samples with label 0, {np.sum(y_train == 1)} samples with label 1")
```

4. **Train the Model**



5. **Inference for Eval**



```
Evaluating the model on the test set ...
Test Loss:     0.2915
Test Accuracy: 0.8894
Test AUC:      0.8984
21/21 [==============================] - 0s 324us/step

Classification Report:
              precision    recall  f1-score   support

   Not Taken       0.92      0.95      0.93       532
       Taken       0.70      0.62      0.66       110

    accuracy                           0.89       642
   macro avg       0.81      0.78      0.80       642
weighted avg       0.88      0.89      0.89       642
```