# S&P Global Market Intelligence – Senior Data Scientist Take home assessment

- Sai Kasyap Kamaraju
- Code : Link

## Exercise 1 – Bankruptcy Dataset Solutions:

- Can you predict the class label of a sample based on its features?

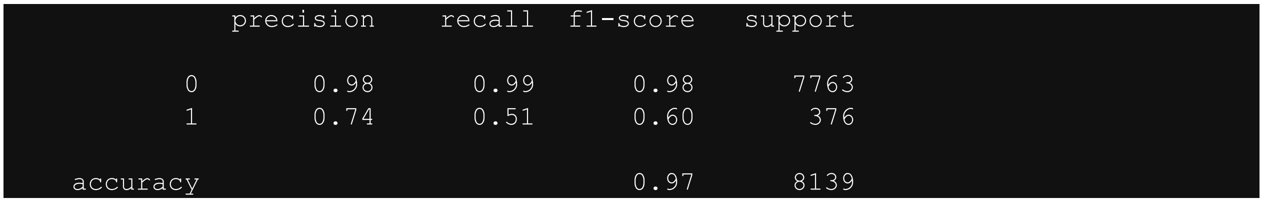
Yes, I was able to predict the class label of a sample based on its features using a supervised model.

- *What is the accuracy of the classifier on the test set?*

The accuracy of the XGB classifier on the test set is 0.97 (unbalanced) and for balanced it was found to be 0.98.

- *How does the classifier perform for each class label?*

The XBG classifier's performance for each class label using metrics such as precision, recall, and F1-score. Here, the dataset is imbalanced and most of the companies do not go bankrupt, the classifier does have high accuracy but low recall for the minority class.

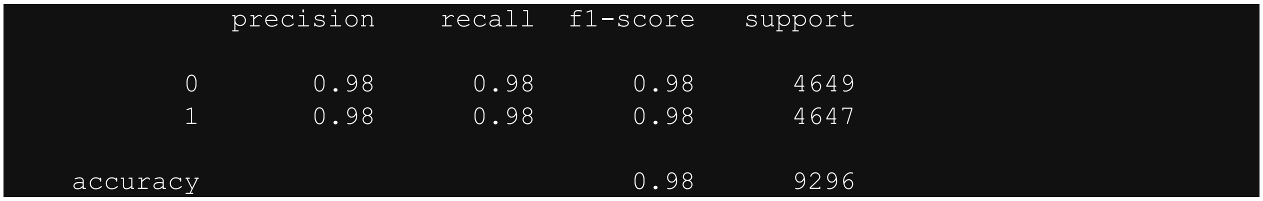


```
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      7763
           1       0.74      0.51      0.60       376

    accuracy                           0.97      8139
```

After, oversampling the data using SMOTE, the recall and F1-score improved significantly.



```
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      4649
           1       0.98      0.98      0.98      4647

    accuracy                           0.98      9296
```

- *Can you identify the most important features for the classifier?*
  The most important feature is feature 25, with a score of 0.07624 using XGBOOST.

- *Can you explain why the classifier misclassified a specific sample?*

  One possible reason for misclassification could be that the sample had feature values that were outliers compared to the rest of the training data, and the classifier was not able to generalize well to such data points.
- *How well does the classifier generalize to new, unseen data?*

We can access the generalization performance of the model using cross validation in general. This will help us how model estimates on new, unseen data.
If the model loss on train, test, validation is close enough then we could say that the model has low variance and can be generalized.

- *Can you compare the performance of the classifier with other models?*

  Yes, compared the performance of the XBG classifiers with other ensemble and other baseline models using metrics such as accuracy, recall.

- *How does the performance of the classifier change with different hyperparameters?*

  The hyperparameter tuning was done but it was done accurately due to the time constraint. The performance of the classifier can change with different hyperparameters such as the learning rate, the number of trees, depth etc. in XGB was tuned using methods like random search.

- *How does the classifier perform on imbalanced datasets?*

The performance of the classifier on imbalanced datasets is affected by the class distribution. In the case of imbalanced datasets, where one class has significantly fewer samples than the other, the classifier may tend to predict the majority (Not Bankrupt) class more often, leading to poor performance on the minority class (Bankrupt).

Hence, Recall is chosen as a preferred metric for evaluation of these models.

In the case of XGBoost, the 'scale_pos_weight' parameter can be used to adjust the balance of positive and negative weights in the dataset. This can help to increase the weight of the minority class and improve the performance of the classifier on the minority class.

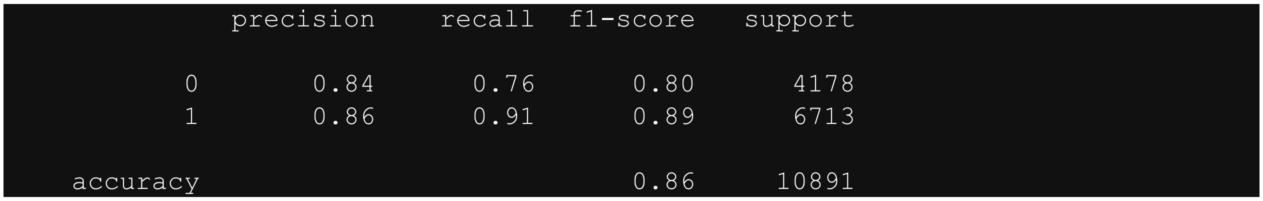
**Exercise 2 : Sentiment Analysis**

- *What is the accuracy of the classifier on the test set?*

The accuracy of the Naïve Bayes classifier with count vectorizer is found to be 0.86 or 86 % on the test set.

- *How does the classifier perform for each class label?*

The performance of the NB classifier for each class label, using metrics such as precision, recall, and F1-score for each class is as follows :



```
              precision    recall  f1-score   support

           0       0.84      0.76      0.80      4178
           1       0.86      0.91      0.89      6713

    accuracy                           0.86     10891
```

- *What model did you use and why?*

I had used Naïve Bayes with Count vectorizer and compared with XGB with TFIDF.
It was found that Naïve Bayes with count vectorizer slightly outperformed the others.
Due to computational resources, was not able use BERT.

- *What major challenges did you face while working on the task?*
  Handling noisy and ambiguous text data -- > Pre-processing of the reviews was done.
  selecting appropriate features - > Use count vectorizer was better in this case.
  Choosing suitable hyperparameters.
  Computational resource for using BERT was one of the other challenges.

- *Can you explain why the classifier misclassified a specific sample?*

Naive Bayes classifiers assume that the features are conditionally independent given the class label.
This assumption may not hold true in some cases, which can lead to misclassification.
Other factors include class distribution (imbalanced).

- *How does the performance of the classifier change with different hyperparameters?*
  This was done with XGB with TF-iDF. The performance of XBG improved significantly in this task after hyper parameter tuning with parameters like learning rate, depth, number of trees etc.

  Before Hyperparameter tuning :

```
           precision    recall  f1-score   support

         0       0.75      0.64      0.69      4178
         1       0.80      0.87      0.83      6713

  accuracy                           0.78     10891
```

  After Hyperparameter Tuning ( Best hyperparameters:  {'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 150} :

```
           precision    recall  f1-score   support

         0       0.81      0.78      0.80      4178
         1       0.87      0.89      0.88      6713

  accuracy                           0.85     10891
```

- Are there any other observations that you had while working on this task?

Yes, Count vectorizer out performed TF-IDF in feature selection.

Use of word embedding models like Glove and Transformer models can also be used to improve the model predictions.