**Week 1: Handle Imbalanced data**

**Data: Dorothea Dataset**

**Approach:**

* **Loading train data and test data**

Converting the indices to binary list.

* **Feature selection:**

1. Dimensionality reduction using truncatedSVD
2. Balancing Imbalance data using SMOTE:

SMOTE is a techique which introduces synthetic minority class examples through interpolation (kind=svm)

* **Cross-validation(Model-selection):**

Cross-validation using train\_test\_split for ideal model selection and for checking possible overfitting. Kfold cross validation was leading to data leakage.

* **Classifier**

Decision tree Classifier with class weights emphasizing on minority class.

**Methodology:**

* Load data:

The data consists of index of non-zero features with class label 1 or zero in the first column for the training set. Hence, initially data is transformed into binary matrix in order to have equal number of elements in each list.

* Feature Selection:

1. Dimensionality reduction :

For dimension reduction truncatedSVD is used, which when used affects the f1-score considerably.

**tsvd = TruncatedSVD(n\_components=1500,n\_iter=50)**

Using PCA makes the result dense and the input dataset is a form of binary sparse matrix hence PCA reduces the F1score.

1. Balancing imbalanced Data using SMOTE

There were two ways to balance data by oversampling or undersampling. Undersampling

reduced the efficiency

**sm = SMOTE(random\_state=42,kind='svm')**

SVM-SMOTE focuses on generating minority class instances near borderlines with SVM so as to build boundary between classes. Hence, using "kind='svm'" affects the efficiency of the algorithm considerably. Without this F1 score of 0.71 was seen.

* Cross-validation:

k-fold cross-validation using model\_selection package from sklearn resulted in over-fitting hence was giving high f1-score. In-order to avoid over-fitting and leakage, train\_test\_split was used before all the transformation and reduction technique. SMOTE used before dimension reduction also resulted in data leakage and reduced F1-score .Pipeline Method was also used to avoid over-fitting with k-fold Cross-validation but it makes no difference when used with train\_test\_split.

* Classifiers

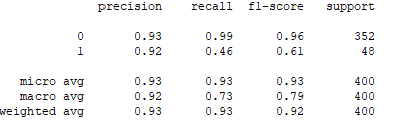
1. **Decision tree:**

Decision tree with class weights gave maximum F1-score

**clf = DecisionTreeClassifier(random\_state=53,class\_weight={0: 1, 1: 1.5})**

Class weights were given in such a way that the emphasis on the minority class would be more than the majority class. Without class weights F1-score was decreasing up to 0.67.

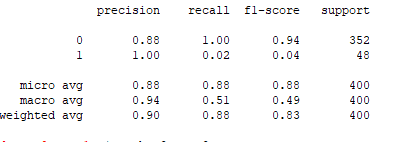
Even with split of 0.5 at cross-validation Decision tree classifier gave decent results. As test data increased, its F1-score increased



1. **Random Tree Classifier:**

Random Tree Classifier gave a F1-score of 0.67 on miner.

Cross-validation with split of 0.5 gave me this result:



As the test data increases Random Forest classifier started giving low scores.

1. **Naive-bayes:**

It was not possible to apply Naive-bayes from sklearn as it didn't handle negative values given by truncated svd.

**Conclusion:**

Data set gave maximum F1-score with combination of truncatedSVD, SMOTE-SVM and Decision tree classifier.

References:

Nguyen, Hien M., Eric W. Cooper, and Katsuari Kamei. “Borderline over-sampling for imbalanced data classification.” International Journal of Knowledge Engineering and Soft Data Paradigms 3.1 (2011): 4–21.