TOXICITY PREDICTION CHALLENGE -FINAL REPORT
Team Scorpion

Data Preprocessing:

- Initialized train.csv, test.csv, feamat.csv to df_train, df_test, df_feamat respectively.
- ‘Id’ column on train.csv and ‘x’ column on test.csv were split into ‘ChemId’ and ‘AssayId’ by using delimiter ‘;’.

```
df_train[['ChemId','AssayId']]=df_train.Id.str.split(";",expand=True)
    df_test[['ChemId','AssayId']]=df_test.x.str.split(";",expand=True)
```

- We found that there is an infinite value in Column in V15 and we handled that infinite value by replacing it with the mean value.

```
 df_feamat.replace([np.inf, -np.inf], np.nan, inplace =True)
df_feamat['V15'].fillna(value=df_feamat['V15'].mean(), inplace=True)
```

- For the Feature selection, we’ve dropped columns with only 0’s and 1’s and after dropping those columns we were left with 29 columns because the classifier might give biased results due to more number categorical values.

```
df_feamat = df_feamat.loc[:, (~df_feamat.isin([0,1])).any(axis =0)
```

- Merged train data and test data with df_Feamat.

```
df_train = df_train.merge(df_feamat, on="ChemId", how="left")
df_test = df_test.merge(df_feamat, on="ChemId", how="left")
```

MODEL TRAINING AND TESTING:

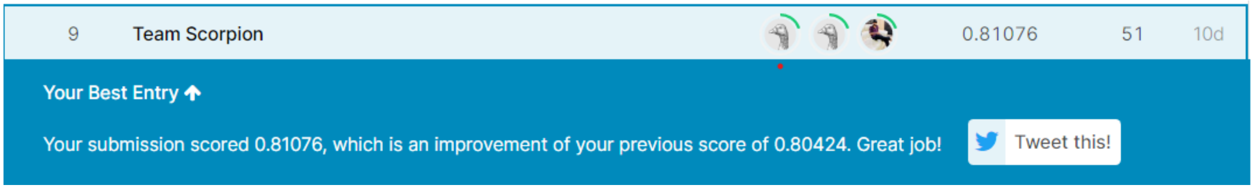
- For validation, we used StratifiedKFold to split the dataset into a training dataset and testing dataset.
- The parameters we used in StratifiedKFold are n_splits=10, random_state=False, shuffle= False. We did not shuffle the dataset randomly.
- For modelling, we used a boosting classifier, XGBClassifier in our final program.
- The parameters that we used for XGBClassifier are n_estimator=400, max_depth=8. We tried to use GridSearchCV for hyperparameter tuning but it was taking a lot of time to execute. So we have chosen parameters based on our internal evaluation scores.
- For Internal evaluation, we have used correlation and F1_score with the average = 'macro'.

Our Best score(According to the private leaderboard):

- Model used: XGB Classifier with parameters n_estimator=400, max_depth=8
- F1 Score: 0.799
- Private leaderboard score: 0.79867
- Public leaderboard score: 0.80424

Leaderboard Ranking:

- **Public leaderboard:**

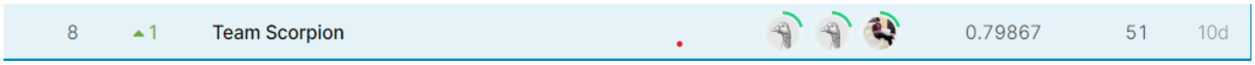


9 Team Scorpion 0.81076 51 10d

Your Best Entry

Your submission scored 0.81076, which is an improvement of your previous score of 0.80424. Great job!

- **Private leaderboard:**



8 ▲1 Team Scorpion 0.79867 51 10d

The below table lists the various attempts made by us on Kaggle:

Model	Parameters	F1 - Score	Public and private Leaderboard Scores	Features
GradientBoosting using XGBClassifier	n_estimator=400, max_depth=8.	0.79954	Public: 0.80424 Private: 0.79867	Used columns till V29
BaggingClassifier with DecisionTree	base_estimator= DecisionTree, n_estimators =100, random_state = 7.	0.7591	Public:0.77151 Private: 0.76557	SelectKBest with Parameter Tuning
Decision tree Classifier	Used GridSearchCV for parameter tuning with the parameters: criteria='gini'and random_state=1.	0.73106	Public: 0.74015 Private: 0.75104	Recursive Feature Elimination
Voting Classifier (Decision Tree and SVM)	Default	0.74214	Public:0.75308 Private: 0.74414	Used SelectKBest and selected the top 15 columns
Decision tree Classifier	Used GridSearchCV for parameter tuning and the parameters are: criteria='entropy' and random_state=1.	0.72623	Public: 0.73857 Private: 0.72609	SelectKBest with the first 6 columns

- We have tried the dimensionality reduction technique, Principal Component Analysis (PCA) but did not get a good result.
- We tried using the AdaBoost ensemble model, but it did not yield us a good internal evaluation score. So we did not do the submission.

Challenges:

Feature selection was one of the challenges that we faced during this competition. We tried SelectKBest, Feature Importance, Correlation, and Recursive Feature elimination. But even though some of these techniques gave us good internal scores, they did not yield good scores after submission, it was overfitting.
We also faced overfitting with most of the models that we have used.

What did we acquire from this project:

We got familiarized with Kaggle competitions and how important these competitions are in machine learning. Through the Kaggle competitions, We got lots of practical and hands-on experience on the topics that were covered in the class.

We learned about multiple prediction techniques which we were not familiar with earlier.
We got hands-on experience on different classifiers, like DecisionTree, RandomForest, Boosting, and Bagging classifiers. We learned about various preprocessing techniques such as correlation, SelectKBest, feature importance, RFE, PCA, etc, and parameter tuning techniques such as GridSearchCV. We also got familiarised with different ensemble models and pipeline models.

Through this course, we also got the opportunity to learn from our fellow teammates. We were motivated to try different feature selection techniques and different modelling algorithms after every presentation.