Toxicity Prediction Challenge

Team Name: FALCONS

Team Members: Sahil Aneja, Rahul Ananda Bijai

Notebook link -
https://colab.research.google.com/drive/16dc0clcTqyQ3BGIV-cckkeb_6nj7dz97?usp=sharing

OS/Platform - Google Colab

Run Steps - refer README.txt

Language - Python

Libraries Used - Numpy, pandas, sklearn, xgboost, lightgbm, imblearn

Data Preparation:

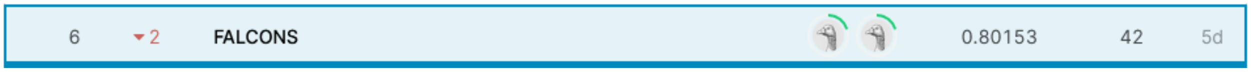
- In Train and Test Data, split the ‘Id’ and ‘x’ fields to extract the ‘chemical id’ and ‘assay id’ respectively.
- Merged the test and train datasets with the feature’s dataset: - join on ‘chemical id’ and ‘V1’ to obtain the complete datasets.
- Dropped the columns from train and test datasets that have the same value in all the rows. For Example, columns having the value 0 for all rows. After performing this task, we were left with 859 columns.
- Replaced the infinite values with max values. For example, for the V15 column the value ‘Inf’ was replaced with the maximum value in that column.
- Normalized the data using the MinMaxScaler.
- Features were further reduced by using correlation with a threshold value of 0.7. Thereafter, we were left with 366 features.
- Performed the test train split using the test_size=0.20.
- Over-sampling using Smote. Explored several Smote oversampling techniques such as Borderline-Smote, ADASYN and hybrid techniques such as SMOTEENN and SmoteTomek.
- Deleted the unused data frames to clean up some memory.

Model Training/Testing:

- We tried the Random Forest, Decision Tree, Logistic Regression models and ensemble models such as XGBoost and Light GBM.

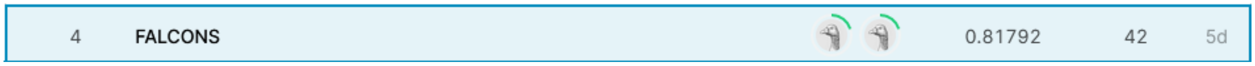
- Internal validation - Used the classification_report, confusion_matrix, accuracy_score and f1_score from sklearn.metrics. F1 score was used as reference for model selection and decision making.
- Feature Selection – After feature reduction through correlation, feature selection activity was performed. RFE and SelectFromModel were used with several models such as Decision Tree, Random Forest, XGBClassifier and LGM Classifier. The features which were suggested by at least three models were chosen and count of such features was 80. So, the model was trained using those 80 features.
- Used RandomizedSearchCV and GridSearchCV to find out the good configuration for the classifiers. GridSearchCV was more accurate but took a long time to execute as all possible combinations were checked.
- Hyper-parameter tuning for Smote (random_state) and XGBoost classifier(learning rate, gamma, max_Depth, n_estimators, booster, random_state etc.) to achieve maximum accuracy.

Private Leaderboard Rank: 6



6 ▾2 FALCONS 0.80153 42 5d

Public Leaderboard Rank : 4



4 FALCONS 0.81792 42 5d

Attempt#	Model	Accuracy	F1 Score	Public Score	Features	Parameters
2	**Decision Tree**	**0.77**	**0.6754**	**0.68735**	Manually selected features with most unique values. 'assay id', V2 – V15	**Default**
5	**Decision Tree**	**0.8745**	**0.7419**	**0.73466**	**859(removed single value columns)**	**Default**
11	**Decision Tree**	**0.876**	**0.7449**	**0.74004**	**859(removed single value columns)**	**Random_state=1 Min_samples_split =20**

16	**RFE+Decision Tree**	**0.87**	**0.743**	**0.74037**	**RFE(100 features)**	**Random_state=1 Min_samples_split=20**
19	**Decision Tree**	**0.87**	**0.74**	**0.74464**	**Correlation 0.8 threshold (450 features)**	**Random_state=1, min_Samples_split=20**
23(R)	**RFE + Random Forest**	**0.88858**	**0.74353**	**0.74690**	**Correlation 0.7 threshold (366 features)**	**Random_state=1, min_Samples_split=20**
24(R)	**XGBoost**	**0.90957**	**0.78954**	**0.79906**	**Correlation 0.7 threshold (366 features)**	**Learning rate=0.25, max_Depth=12, booster='gbtree', n_estimators=100, random_state=0**
26	**XGBoost**	**0.91584**	**0.80790**	**0.80068**	**Correlation 0.7 threshold (366 features)**	**Learning rate=0.25, max_Depth=12, booster='gbtree', n_estimators=250, random_state=1**
32(R)	**XGBoost (after Smote oversampling the whole Train DataSet)**	**0.94**	**0.93926 (internal score is high because we tested on oversampled data. It was rectified in further attempts**	**0.80790**	**Correlation 0.7 threshold (366 features)**	**Learning_rate=0.15, booster='gbtree', gamma=0.4, max_depth=15, random_state=0, n_estimators=250**

			)			
33	XGBoost (after Smote oversampling the whole Train DataSet)	0.95	0.94837 (internal score is high because we tested on oversampled data. It was rectified in further attempts)	0.81248	Correlation 0.7 threshold (366 features)	Learning_rate=0.25 , booster=‘gbtree’, gamma=0.3, max_depth=12, random_state=1, n_estimators=250
42(R)	XGBoost (after Smote oversampling X_train and y_train)	0.91158	0.80673	0.81792	Correlation 0.7 threshold (366 features) Created a Feature Importance Dataset using RFE, SelectFromModel and SelectKBest and selected top 80 features having count of 3	colsample_bytree=0.8, gamma=1.4, learning_rate=0.1, max_delta_step=0, max_depth=16, min_child_weight=5, missing=None, n_estimators=700, objective='multi:softmax', random_state=0,subsample=0.8, num_class = 2

Attempts Detail: Detail of all the attempts that lead to an increase in the public score.

Attempt 2 : We selected some features manually such as assay id, V2 – V15 and applied the DecisionTreeClassifier. It was an initial attempt during which we were trying to understand the problem and test some models such as DecisionTree and RandomForest. The DecisionTree classifier gave us better accuracy/f1 score as compared to the RandomForest classifier. It was the base for future attempts.

Attempt 5 : We removed all the columns that contained the same value for all the rows. For example columns having only the value ‘0’. About 859 columns were left after performing this cleanup. The data was normalized using the MinMaxScaler. The model was trained with all the features (about 859 in number). We saw a good increase in the public score which shot up to 0.73

Attempt 11: We used most of the features (about 859) after cleaning some columns. (as done during the 5th attempt). Tweaked some classifier parameters - Random_state=1 Min_samples_split=20. As a result of parameter tweaking our internal validation and public score saw a slight increase.

Attempt 16: We tried Recursive Feature Elimination(RFE) with 50 and 100 features. Also, tried the SelectKBest(50 features). We noticed that out of all these, RFE with 100 features is giving us a slightly better internal score. So, we used RFE with the Decision Tree classifier in a Pipeline.

Attempt 19: We reduced the number of features using the Correlation (with threshold = 0.8). We were left with 450 features. There was a slight increase in the internal validation score and public score.

Attempt 23: We further reduced the number of features using the Correlation (with threshold = 0.7). We were left with 366 features. After Correlation, we used SelectFromModel using RandomForest Classifier which gave us 32 features out of 366 features. We created a loop for RFE with RandomForest Classifier to check from 1-32 features and give the accuracy score for all counts till 32. Hence we got 4 features giving us the maximum accuracy as compared with other counts. There was a slight increase in the internal validation score and public score.

Attempt 24: After the same data preprocessing with correlation giving us 366 features, we used XGBClassifier with hyperparameter tuning done by RandomSearchCV and GridSearchCV. This significantly improved our score in the public leaderboard from 0.74690 to 0.79906

Attempt 26: We tested our model with another ensemble model (Light GBM), but persisted with XGBoost as it gave better results. We then did some hyperparameter tuning for XGBoost such as updating the n_estimators and random_state parameters and there was again some increase in the internal and public score.

Attempt 32 : After the same data preprocessing with correlation giving us 366 features, we oversampled the entire Train dataset using SMOTE before train test split. We used XGBClassifier with RandomSearchCV and GridSearchCV for hyperparameter Tuning. We saw a slight improvement in the internal evaluation as well as public leaderboard scores

Attempt 33: We made some changes in the hyperparameters such as Learning_rate=0.25, gamma=0.3, max_depth=12, random_state=1 to achieve better accuracy results.

Attempt 42: After the same data preprocessing with correlation giving us 366 features, we created a new Feature importance dataset. We used overall 9 feature selections, i.e, RFE and SelectFromModel were used with four models Decision Tree, Random Forest, XGBClassifier and LGM Classifier as well as SelectKBest with Chi2. The features which were suggested by at least three models were chosen and count of such features were 80. So, the model was trained using these 80 features. We used SMOTE for oversampling only X_train and Y_train. Then used XGBClassifier with manual hyperparameter tuning to train the model. We had achieved the highest score in both our F1 scores and public leaderboard.