Project Report

Kushal Vipul Adhvaryu (202106851)
x2021guh@stfx.ca

Data Preparation -

For the data preparation step, I first loaded the data into Google Colab, and explored the data. The data contained two columns - ‘Id’ and ‘Expected’

I split the ‘Id’ column into ‘chemical_id’ and ‘assay_id’ columns and renamed the ‘label’ column to ‘Expected’ column.

After this, I dropped the ‘Id’ and ‘Expected’ columns since they were not needed.

I did this same procedure for both train and test data.

Next, I loaded the Chem.Descriptors, Chem.rdMolDescriptors, Chem.GraphDescriptors from the rdkit library. Using the MolfromSmiles functionality I converted the chemical_id column into molecules and used these molecules to get some features.

Feature Selection -

I researched chemical informatics datasets and which features are optimal when dealing with these datasets. After trying a lot of Descriptors, rdMolDescriptor combinations and MorganFingerprints, I found some descriptors which provided the most accuracy -

- Descriptors.MolWt(mol),
- Descriptors.MolLogP(mol),
- Descriptors.TPSA(mol),
- Descriptors.BalabanJ(mol),
- Descriptors.MolMR(mol),

These descriptors provided the best accuracy. After getting the features, I assigned them their column names respectively.

Imputing -

Since some features had null values in them, it was important to deal with those null values by assigning a mean value to them. I assigned mode or the most occurring value to the features that were imbalanced.

After imputation and feature extraction, I found the correlation between these features and if there was less correlation, I eliminated those features.

The same feature extraction and imputation process was performed on the test data.

After the feature selection and imputation process, I selected features for X and y. For the target variable, I selected the ‘label’ column. I dropped ‘chemical_id’ and ‘label’ and selected the other features for X.

Label Encoding -

In the last step for data preparation, I used label encoding to encode the target values in order to have consistent numerical values for the target variable. I did not scale the features since XGBoost is sensitive to scaling and the model produced better accuracy without scaling.

Modeling -

The next process was to apply models to the dataset.

In order to select the best performing model, I tried out several models and checked their accuracy.

The models that I tried out -

- Random Forest Classifier
- K Neighbors Classifier
- Support Vector Machine
- Decision Tree
- XG Boost

Out of these models, the XGBoost Classifier had the best performance in terms of accuracy. Hence I chose the XGBoost model.

In order to get the best hyperparameters, I applied a grid search method with different hyperparameters. This was used with cross validation = 5 in order to improve results.

The parameters providing the best score were noted and used to predict test data.

Internal Evaluation -

I used train_test_split functionality to divide the dataset into training and validation sets. I used the validation set to check the F1 score, precision and recall. I modified the hyperparameters accordingly.

The last step was to predict the results on test data. In order to do this, the same features and preprocessing techniques were used as in the training data. I dropped the ‘chemical_id’ column from the test data. Using the pre-trained XGBoost model, I predicted the labels on the test data. I used an inverse transform on the predicted value to get real predicted values.

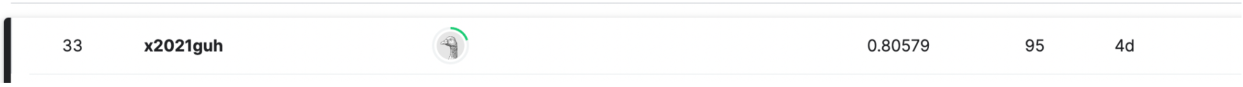
I created a dataframe using the predicted values, ‘chemical_id’ and ‘id’ columns. After converting the data frame to a csv file, it was ready for submission.

Leaderboard Score -

Best score achieved -

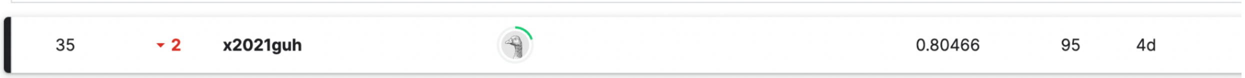
By using the following features, models and techniques, the best accuracy that I achieved was 80.5% on the public leaderboard and 80.46% on the private leaderboard.

Public Leaderboard ranking -



33	x2021guh	0.80579	95	4d

Private leaderboard ranking -



35	▾ 2	x2021guh	0.80466	95	4d

Project link -

https://github.com/Kushal-55/chemical_toxicity_prediction_challenge