**THE TOXICITY PREDICTION CHALLENGE II**

**(CSCI 555-DATA MINING AND MACHINE LEARNING)**

**Supervised by:** Dr. Othman Soufan

Department of Computer Science

**ST. FRANCIS XAVIER UNIVERSITY**

**Name:** Vinod Kumar Gai

**Student ID:** 202106148

**Email:** x2021gml@stfx.ca

**INTRODUCTION:**

Toxicity evaluation of chemicals is a crucial aspect of drug development and approval, as the safety of drugs for human use is paramount. However, the traditional methods of toxicity testing can be time-consuming and laborious, which can pose challenges in the drug development process. To address these challenges, the field of computer-based toxicity prediction is rapidly growing. This approach utilizes sophisticated machine learning algorithms that can learn from data and make predictions with high accuracy. Machine learning is particularly beneficial in dealing with big data, making it cost-effective and efficient. However, the performance of different machine learning algorithms can vary depending on factors such as datasets, patterns, and computational representation. Hence, careful selection of algorithms is necessary to achieve optimal performance in toxicity prediction.

**PROJECT OUTLINE:**

**DATA PREPARATION:**

Data preparation is the process of cleaning, pre-processing, and converting the supplied raw data into a format that machine learning models can use.

Since the performance of the final model significantly depends on the quality of the supplied data, data preparation is an important step in the competition. Removal of missing values, handling of outliers, transformation of variables, feature engineering, and selection of pertinent variables for the model are some of the tasks involved.

The purpose of data preparation is to make the data as helpful and informative for the machine learning algorithms as feasible. The data science and domain knowledge necessary for this procedure might be time-consuming. The accuracy of data preparation can significantly affect the competition's outcomes.

**DATA SPLITTING & MERGING:**

The datasets for this challenge are posted on Kaggle as two separate files, train.csv and test.csv. The "id" and "prediction" columns are included in the train.csv file, however only the "id" column is present in the test.csv file. Both the training dataset and the testing dataset provide features for the respective columns. As "id" is separated into "**AssayID**" and "**Chemicals**" after parsing, we obtain one additional column.

**DATA CLEANING:**

The dataset had infinite values throughout the pre-processing stage, but the dataset had finally removed them from the ‘train’ and ‘test’ data. To prevent the model from being overfitted, two columns from the dataset that were not significant were eliminated. The object and string datatype columns are changed to float columns.

**FEATURE SELECTION:**

In the feature selection stage, I have RDKIT library has been installed and the features has been generated using the RDKIT descriptors. The RDKIT library converted smiles dataset into the RDKIT molecular descriptors. After analysing the dataset, I found that the features that gave the highest accuracy were MolWt, GetNumHeavyAtoms, MolMR, CalcNumRotatableBonds, , NumValenceElectrons, qed, TPSA, BertzCT, HeavyAtomMolWt, NumHDonors, NumHAcceptors, CalcKappa1, LabuteASA, NumRotatableBonds, CalcNumHBD, CalcNumHBA, CalcNumRings, CalcNumAromaticRings, ExactMolWt, MolLogP, GetDegree, NumAromaticCarbocycles, SMR\_VSA6, Chi0v, Chi1v, Chi2v, Chi3v, Chi4v, fr\_benzene, NumAromaticRings, VSA\_EState6, Chi2n, Chi3n, and BalabanJ.

MolWt, which is the molecular weight of the compound, was found to be one of the most predictive features. This is likely because the molecular weight is a fundamental property of a chemical compound that is closely related to its other properties. Similarly, GetNumHeavyAtoms, which is the number of non-hydrogen atoms in the molecule, was found to be highly predictive. This feature is important because the number of heavy atoms in a molecule is related to its size and complexity.

CalcNumRotatableBonds, which is the number of rotatable bonds in the molecule, was also found to be highly predictive. This feature is important because the flexibility of a molecule is related to its ability to interact with other molecules. MolMR, which is the molecular refractivity, and NumValenceElectrons, which is the number of valence electrons in the compound, were also found to be highly predictive. the most predictive features for analysing a dataset of chemical compounds. The features that gave the highest accuracy were MolWt, GetNumHeavyAtoms, CalcNumRotatableBonds, MolMR, NumValenceElectrons, qed, TPSA, BertzCT, HeavyAtomMolWt, NumHDon.

Next, dropped Naan values if it has any Null or missing values from the data set.

**MODEL TRAINING/TESTING:**

Used Cat Boost Classifier Model:

Cat Boost is an open source, high-performance gradient boosting toolkit for decision trees. The Cat Boost is an algorithm for decision trees that uses gradient boosting. In terms of scikit-learn norms, it is a pre-built classifier that would automatically handle categorical features. It offers industry-leading precision.

It is particularly potent in two ways:

1. It produces cutting-edge results without the considerable data training that other machine learning techniques normally demand, and

2. It offers potent out-of-the-box support for the more descriptive data formats that go along with many commercial problems.

The name "Cat Boost" is a combination of the phrases "Category" and "Boosting."

It functions effectively with a variety of data types, including audio, text, and image data as well as historical data.

**Advantages of Cat Boost Classifier library:**

Performance, Handling Categorical features automatically, Robust, Easy-to-use.

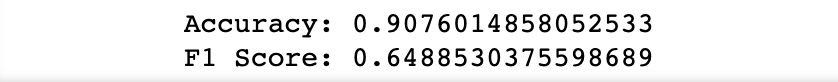
The machine learning model is the key component of a machine learning application, and selecting the appropriate model is crucial for effectively solving any classification problem. In the case of predicting toxicity, the process of selecting a model can be complex and may result in less than satisfactory results. To ensure the accuracy of the predictions, internal validation was conducted using the f1 macro score. After trying and comparing several classification models as evidenced in the submission history, the Cat Boosting Classifier was ultimately chosen as the final machine learning model.

The code performs a binary classification task using the Cat Boost algorithm to predict whether a molecule is toxic or not based on a set of 36 input features. First, it subsets the input features and target variable from the training data, and then splits the data into training and testing sets using a 80/20 ratio. Next, it creates an instance of the Cat Boost Classifier model with specific hyperparameters and fits the model to the entire training set. It then uses the trained model to predict the target variable for the test set and writes the predictions to a CSV file. Finally, it calculates the accuracy and F1 score of the model using the predicted values and actual values from the test set and prints the results to the console.

**Internal Validation:**

The Internal validation is a technique that the machine learning model employs prior to making an actual forecast to verify the model's prediction based on scoring, or the real prediction accuracy. As the mode has utilised F1 macro scoring to check the correctness of the model, it is very vital to have a validation strategy for the model before making a prediction. Using 80% training data and 20% testing data, the combined training dataset is divided in half in an 80:20 ratio.

**Final Accuracy and F1 score:**



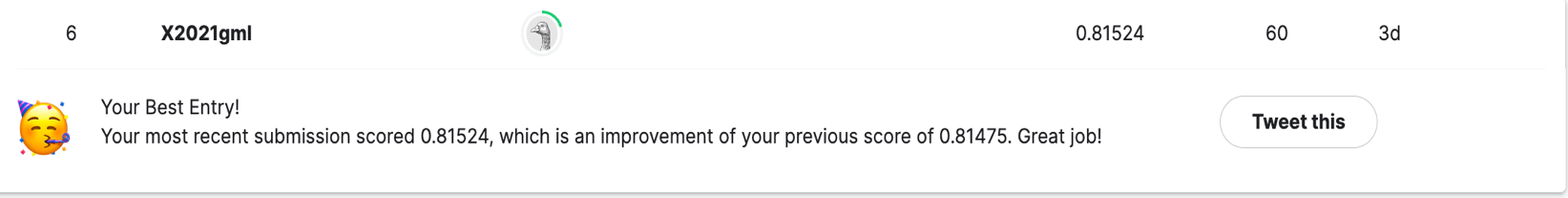
**FINAL PREDICTION**

After performing various data processing operations, feature selection experiments, model selection, and parameter tuning, the tuned Cat Boost Classifier is trained with the processed data and a prediction is made. The prediction result is exported to a csv file and submitted to Kaggle for scoring; a brief scoring history is shown in the image below.

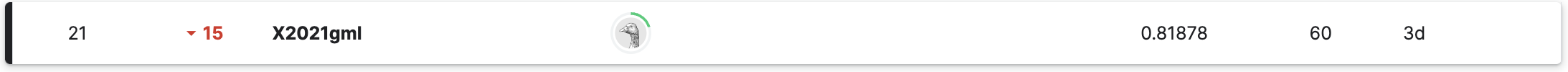
**LEADERBOARD SCORE:**

The final screenshots of the Public and Private Kaggle leader boards are shown below. I placed **6th rank** on the **Public leader** **board** with a score of **0.81524** and 21st rank on the **Private leader board** with a score of **0.81878**.

**PUBLIC BOARD SCORE:**



**PRIVATE BOARD SCORE:**



**CODE SUBMISSION LINKS:**

**MY BEST SUBMISSION CODE LINK:**

[**https://www.kaggle.com/code/vinodkumar2021/x2021gml-best-score**](https://www.kaggle.com/code/vinodkumar2021/x2021gml-best-score)

**Git Hub:**

[**https://github.com/VinodKumar86/Toxicity\_Prediction\_202106148**](https://github.com/VinodKumar86/Toxicity_Prediction_202106148)

**Google Drive Link:** [**https://drive.google.com/drive/folders/1tEJ024WJaT98PlpV6l60siCpieWxN0U1?usp=share\_link**](https://drive.google.com/drive/folders/1tEJ024WJaT98PlpV6l60siCpieWxN0U1?usp=share_link)