Final Project Report for CS 184A/284A, Fall 2024

Project Title: Prediction of Human Gut Biotransformation Pathways
Project Number: 17

Student Name(s)
Yashas Raman, 49323976, yraman@uci.edu
Raghav Sriram, 28137297, sriramr2@uci.edu
Rishit Sharma, 45329724, rishits3@uci.edu

1. Introduction and Problem Statement

This project will use machine learning techniques on a dataset consisting of human gut biotransformation reactions to predict how the human gut microbiome transforms chemical compounds. By analyzing chemical and biological data, we aim to forecast the end products of these transformations and the pathways they take. Success will be measured by comparing our predictions with real data to see how closely they match.

The goal of this project is to predict what happens to chemical compounds when they interact with the human gut microbiome. In the gut, microbes break down and modify compounds through a series of chemical reactions, resulting in new molecules called metabolites. Understanding these changes is essential for drug development, assessing chemical safety, and personalized medicine. Our system will take in the chemical structure of a compound and predict the sequence of transformations it undergoes, identifying the most likely pathways and end products.

2. Related Work

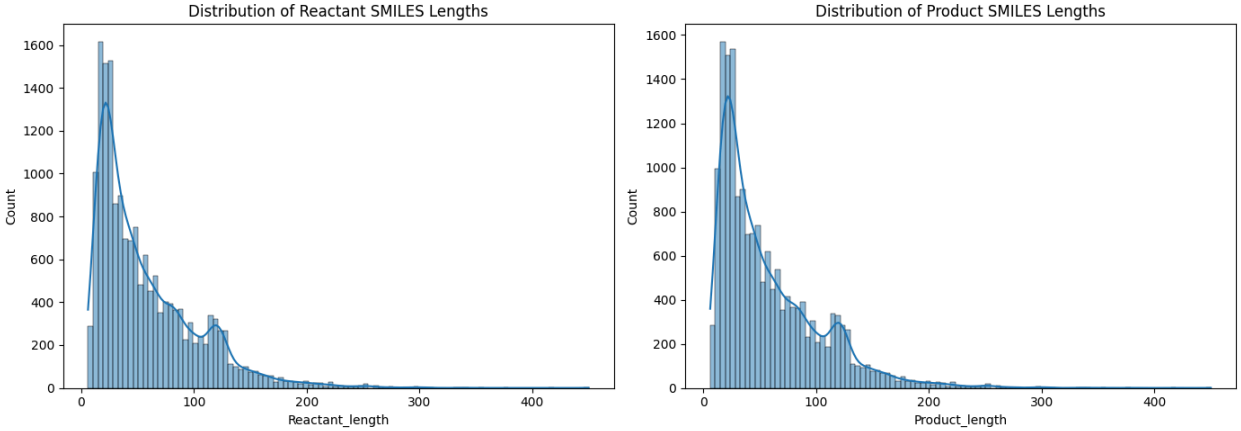
The prediction of chemical transformations in the human gut microbiome has been explored through innovative approaches. Rule-based systems, like cheminformatics libraries such as RDKit, have been widely used for their clear and interpretable reaction templates. Schwaller et al. (2018) made a significant breakthrough by introducing a sequence-to-sequence neural translation model which treated chemical reactions as language-like sequences. A popular approach building off this research, such as by Kaggle user Poma Mikhail, was to apply reaction templates to generate a random walk, which essentially generates a pathway by taking a sequence of random steps. However, this approach faced challenges in generalizing to rare or unseen reactions, which are crucial in biochemical contexts, as it is not unable to properly identify trends in the data.

Our approach establishes a thorough multi-step pathway prediction methodology for chemical transformations occurring within the human gut. Our system mimics transformation sequences using a hybrid strategy by combining template-based prediction and supervised machine learning. Drawing from Rogers and Hahn's (2010) molecular fingerprint techniques and inspired by MoleculeNet's evaluation strategies (Wu et al., 2018), we utilize classification methods such as Random Forest and LightGBM to predict the most likely pathways for a chemical compound in the gut. By combining sophisticated machine learning techniques with interpretable models, we attempt to create a more nuanced way of mapping out complex chemical transformations.

3. Data Sets

Dataset URL:
https://www.kaggle.com/competitions/prediction-human-gut-biotransformation-pathways/data

There are three files contained in our dataset, a training csv, a test csv, and sample submission csv. The training dataset contains 17301 gut biotransformation reactions. Each reaction is uniquely identified with an ID and the SMILES of reactants and products of each reaction. The test csv contains an ID of the pathway, the number of steps each pathway has (ie number of metabolites), and the first compound in the pathway. The csv file our model produced from running on the test dataset contains the ID of the pathway and then the first, second, third, fourth and fifth ranked predictions made by the model. The figures below depict the length and quantity of our SMILES, describing the general shape of our dataset:



The figures below show what our raw training data looks like (CSV file):

	ID	SMILE	Reactant
0	ID0000001	CC(=O)NC(CC(=O)O)C(=O)O>>CC(=O)NC(CC(=O)O)C(=O...	CC(=O)NC(CC(=O)O)C(=O)O
1	ID0000002	NCCc1ccc(O)cc1>>O=CCc1ccc(O)cc1	NCCc1ccc(O)cc1
2	ID0000003	*C(NC(=O)CC1CCC(=O)C1C/C=C\CC(=O)O)C(=O)O>>*C(...	*C(NC(=O)CC1CCC(=O)C1C/C=C\CC(=O)O)C(=O)O
3	ID0000004	CC(=O)N[C@H]1C(O)O[C@H](CO)[C@@H](O)[C@@H]1O[C...	CC(=O)N[C@H]1C(O)O[C@H](CO)[C@@H](O)[C@@H]1O[C...
4	ID0000005	OC[C@H]1O[C@@H]2OC[C@H]3O[C@H](O[C@@H]4[C@@H](...	OC[C@H]1O[C@@H]2OC[C@H]3O[C@H](O[C@@H]4[C@@H](...
5	ID0000006	O=C(O)c1cccc(C(=O)CC[C@H]2O[C@@H](n3cnc4c(=O)[...	O=C(O)c1cccc(C(=O)CC[C@H]2O[C@@H](n3cnc4c(=O)[...
6	ID0000007	CC(CO)(c1ccc(O)cc1)c1ccc(O)cc1>>CC(C)(c1ccc(O)...	CC(CO)(c1ccc(O)cc1)c1ccc(O)cc1
7	ID0000008	*C(=O)N[C@@H](CO[C@@H]1O[C@H](CO)[C@@H](O[C@@H...	*C(=O)N[C@@H](CO[C@@H]1O[C@H](CO)[C@@H](O[C@@H...
8	ID0000009	*OP(=O)(O)OC[C@H]1O[C@@H](*)C[C@@H]1OP(=O)(O)O...	*OP(=O)(O)OC[C@H]1O[C@@H](*)C[C@@H]1OP(=O)(O)O...
9	ID0000010	NP(=O)(OCCC=O)N(CCCl)CCCl>>O=P1(N(CCCl)CCCl)NC...	NP(=O)(OCCC=O)N(CCCl)CCCl

	Reactant	Product	Reactant_length	Product_length
	CC(=O)NC(CC(=O)O)C(=O)O	CC(=O)NC(CC(=O)O)C(=O)NC(CCC(=O)O)C(=O)NC(CCC(...	23	57
	NCCc1ccc(O)cc1	O=CCc1ccc(O)cc1	14	15
	*C(NC(=O)CC1CCC(=O)C1C/C=C\CC(=O)O)C(=O)O	*C(NC(=O)CC1CCC(=O)C1C/C=C\CC=O)C(=O)O	41	38
	CC(=O)N[C@H]1C(O)O[C@H](CO)[C@@H](O)[C@@H]1O[C...	CC(=O)N[C@H]1C(O)O[C@H](CO)[C@@H](O)[C@@H]1O[C...	84	119
	OC[C@H]1O[C@@H]2OC[C@H]3O[C@H](O[C@@H]4[C@@H](...	OC[C@H]1O[C@H](OC[C@H]2OC(O)[C@H](O)[C@@H](O)[...	163	79
	O=C(O)c1cccc(C(=O)CC[C@H]2O[C@@H](n3cnc4c(=O)[...	Nc1ncnc2c1ncn2[C@@H]1O[C@H](CCC(=O)c2cccc(C(=O...	74	68
	CC(CO)(c1ccc(O)cc1)c1ccc(O)cc1	CC(C)(c1ccc(O)cc1)c1ccc(O)cc1	30	29
	*C(=O)N[C@@H](CO[C@@H]1O[C@H](CO)[C@@H](O[C@@H...	*C(=O)N[C@@H](CO[C@@H]1O[C@H](CO)[C@@H](O[C@@H...	230	188
	*OP(=O)(O)OC[C@H]1O[C@@H](*)C[C@@H]1OP(=O)(O)O...	*OP(=O)(O)O[C@H]1C[C@H](*)O[C@@H]1COP(=O)(O)O	117	45
	NP(=O)(OCCC=O)N(CCCl)CCCl	O=P1(N(CCCl)CCCl)NC(O)CCO1	25	26

4. Technical Approach

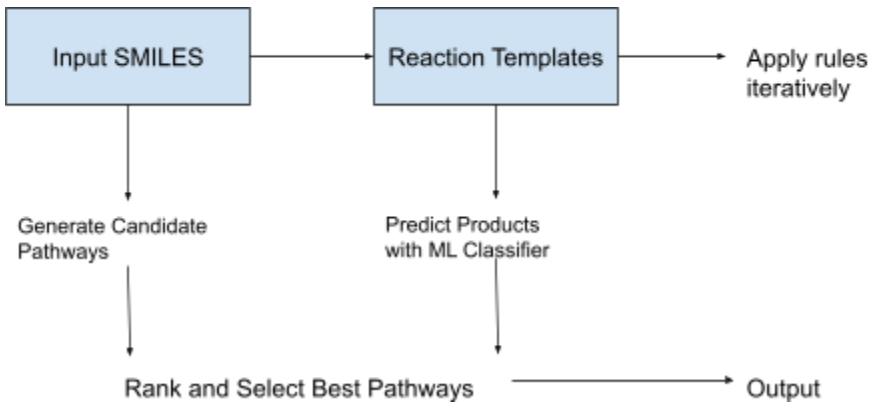
Our project aims to predict human gut biotransformation pathways for given chemical compounds. The overarching goal is to start with a known reactant and determine the sequence of transformations it might undergo in the human gut environment, producing a set of likely metabolites.

Data Representation:
We use SMILES (Simplified Molecular Input Line Entry System) strings to represent chemical compounds. These textual representations capture the structure of molecules in a linear form. By processing the training data, we attempted to establish a mapping between a reactant and product SMILES. We removed invalid SMILES and ensured only valid chemical structures remained. We also removed duplicate reactant/product pairs to avoid bias. For modeling, we used feature selection, cutting down the number of features from 2048 to just 500, in order to reduce dimensionality and improve training speeds.

Core Algorithms and Techniques:
We employed the RDKit library, a powerful toolkit for cheminformatics, to parse and handle chemical structures. Each reactant/product pair in the training set was used to extract a reaction template. A reaction template provides a generalized rule representing how a certain reactant turns into a certain product. Given a starting molecule, we iteratively apply our reaction templates. By chaining these transformations step-by-step, we simulate a pathway. We generate multiple candidate pathways and then use scoring methods to rank the most likely ones.

To improve predictions and possibly rank outcomes, we used molecular fingerprints as features. Specifically, we used Morgan fingerprints (circular fingerprints), which were generated from the SMILES using RDKit. These fingerprints transform the molecular structure into a high-dimensional binary vector that represents chemical structures. We then combined reactant and product fingerprints to create training examples for supervised learning models.

We experimented with two machine learning algorithms, LightGBM and Random Forest. LightGBM is a machine learning model that improves its predictions by focusing on the most difficult examples during training, a process known as gradient boosting. In our project, it is used to predict and prioritize the most likely chemical reactions, especially when the data is uneven or imbalanced. Random Forest is a machine learning method that combines the results of many decision trees to make better predictions. For our project, it helps rank and classify the possible chemical transformations by analyzing patterns in the molecular features of reactants and products. We expanded on Random Forest by creating two models, one using 100 estimators and another using 200 estimators, in order to see the effects of increasing the amount of estimators on model performance. The figure below outlines our general system pipeline used in this project:



5. Software

The table below outlines all major pieces of software and code involved in the execution of this project:

Code Component	Author	Functionality	Input	Output
Data Cleaning Script	50% team, 50% external code derived from Kaggle user Poma Mikhail	Loads and parses data and validates SMILES strings.	Training and testing data	Cleaned dataset
Reaction Template Code	External (Kaggle user Poma Mikhail)	Extracts reaction templates from reactant-product pairs (via RDKit)	Training data	List of reaction templates
Fingerprint Generation	Team (written)	Functions using RDKit to generate Morgan fingerprints from SMILES.	SMILES string	Numpy array of features
ML Training Scripts	Team (written)	Code that fits LightGBM and Random Forest model to our training data, and evaluates them when applied to our testing data	Feature arrays and labels	Fitted model, metrics
Visualization Scripts	Team (written)	Scripts to plot histograms, bar charts, and confusion matrices for interpretability	Dataframes, predictions	Charts/Plots needed to visualize results
RDKit Library	External	SMILES parsing, fingerprint generation, reaction templates	SMILES strings	Molecules, fingerprints
Sklearn/NumPy/ Pandas	External	Data manipulation, ML algorithms, metrics	CSV data, arrays	Processed data, ML models

6. Experiments and Evaluation

We chose to evaluate each of our models by using the Final Score (FS) metric as outlined by the Kaggle competition organizers for this specific project. The formula for the metric can be seen below:

$$Final\ Score\ (FS) = \frac{\sum_{i=1}^{n} PS_i}{n}$$

This metric is an accuracy score that represents the average of the pathways' similarity in the test dataset. The value of FS is between 0 and 1, with higher values indicating better performance. The PS in the formula represents pathway similarity, which compares how similar the pathways predicted by our model are with the true reaction pathway. PS is calculated using the following formula:

$$Pathway\ Similarity\ (PS) = \frac{\sum_{j=1}^{m} PMS_j}{m}$$

The PMS in the formula represents the similarity between a predicted metabolite and the true metabolite in a reaction pathway. PMS is calculated using the following formula:

$$Similarity(PMS) = 1 - \frac{LD(gt, pm)}{max(len(gt), len(pm)))}$$

In this formula LD represents the Levenshtein Distance (LD), which is the difference between two string sequences, between the ground truth metabolite and the predicted metabolite. Essentially, the overall FS metric essentially compares our predicted pathways to the true pathways character-by-character using LD, allowing us to determine the overall effectiveness of our machine learning models.
We used an 80/20 train/validation split of the training data, and evaluated our model's performance using accuracy and F1 precision score (weighted).

Results of the Models:
LightGBM Model:

- Accuracy: 0.7107241788682231
- F1 Score: 0.6498877034018843

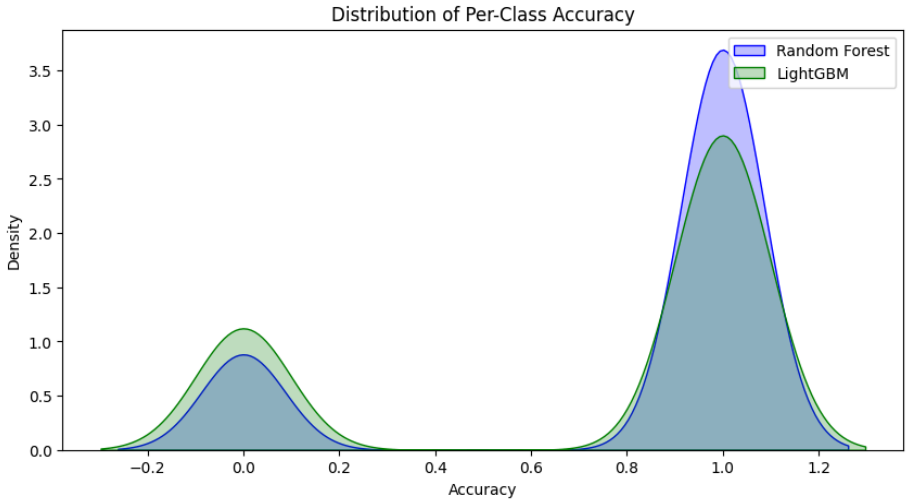
Random Forest (100 Estimators):

- Accuracy: 0.7855164226355362
- F1 Score: 0.7365284432960275

Random Forest (200 Estimators):

- Accuracy: 0.7859121487930352
- F1 Score: 0.7371349093358137

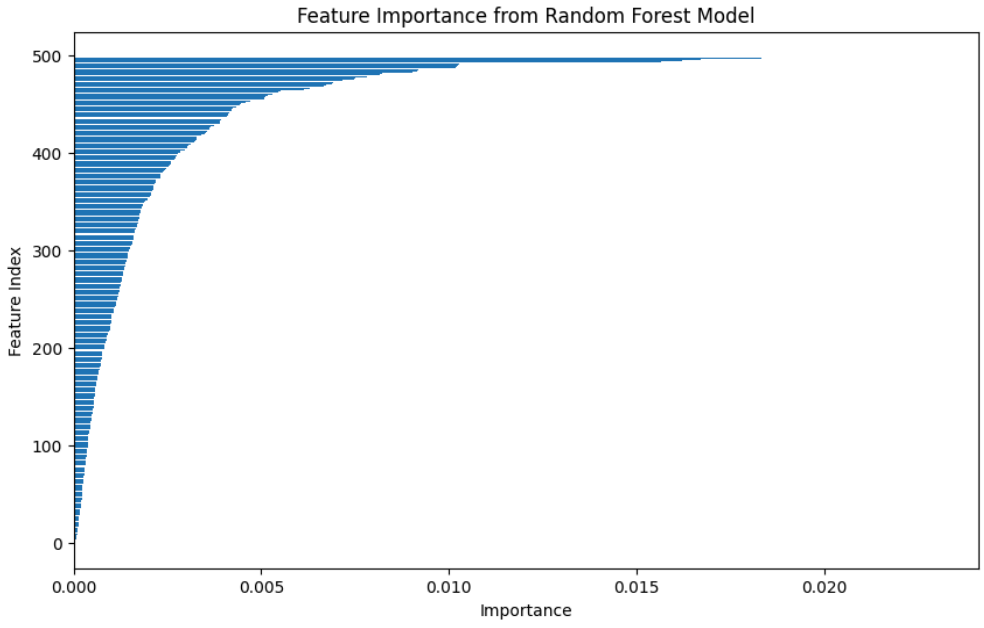
The figure below shows how Random Forest had a higher density of correct predictions (1.0) while LightGBM had a higher density of incorrect predictions (0.0), which is consistent with our results:



We can see that both Random Forest models significantly outperformed the LightGBM model in both accuracy and F1 score. This is likely due to the nature of our dataset, as Random Forest models tend to perform better than LightGBM models on complex datasets. As random forest models train each decision tree independently, as opposed to LGBM which uses gradient boosting, it produces more robust models that tend to fit complex data better. However, we found that increasing the amount of estimators from 100 to 200 did not seem to cause any significant improvement. This suggests that the model had already reached a point of diminishing returns in terms of additional estimators, with further increases providing minimal benefits. Overall, these results highlight the robustness of Random Forest for handling complex, high-dimensional datasets, making it a strong choice for our project.

Sensitivity to algorithmic choices played a critical role in shaping the performance of our models. We initially intended for our Random Forest models to use all 2048 features from the dataset. However, we found that the models simply would not run with all the features present, as they took far too much time to even produce results.Therefore, we found it necessary to use feature selection, and cut down the number of features to 500. This likely affected our model performance, as the remaining 1548 features may have contained crucial information that could have improved our models' performance. Normalization and feature selection were critical for model stability. Techniques like SKLearn's SelectKBest improved training, and we found that proper data cleaning significantly enhanced model performance. Our template selection strategy required careful calibration: more restrictive templates increased precision but limited pathway generation, while broader templates produced more diverse results at the cost of accuracy. Ultimately, finding the right balance between these approaches was key to optimizing our model's effectiveness.

The figure below depicts the trend seen in feature importance for our 500 features. The exponential curve shows that the most important features had by far the most significant impact on the models:



More figures pertaining to our results can be found in the appendix section of this report.

7. Discussion and Conclusion

We learned that the complexity of chemical transformations can be partially captured by reaction templates and molecule fingerprints. The ML models managed to capture some structural transformation patterns but struggled with rare transformations. The Random Forest's stable performance suggests that ensemble methods, which combine the predictions of multiple trees, handle complex, high-dimensional cheminformatics data well especially compared to gradient boosting methods such as LightGBM.

We initially expected our LightGBM model to outperform our Random Forest models, due to the large size of our dataset. However, we failed to account for the highly complex nature of our data, which is likely the reason why our Random Forest models actually performed better. We also expected the model to perform slightly better on our test data. This suggests that our pipeline heavily depended on known transformations from training data. Without similar patterns in the training set, predictions were less accurate.

Despite our large dataset, our models may have still been limited by our level of data coverage, as space of possible chemical transformations is vast. Our model only learned from 17301 known reaction pairs, which is still an extremely small subset of all possible reactions. Because of this, the model likely struggled when a test compound's transformations were not sufficiently represented by similar examples in the training data. Our use of feature selection may have also inhibited the performance of our models. In order to have our model run in a timely manner, we had to reduce the number of features in our dataset from 2048 to 500. This may have caused important information in some of our data to be lost, causing our model to perform worse.

Future steps we could take to improve our approach involve exploring advanced computational techniques and expanding the scope of our data. One promising direction is investing in Graph Neural Networks (GNNs) or other deep learning models specifically designed for chemical reaction prediction. These models excel at capturing the relationships between molecular components, making them ideal for complex reaction pathways. Additionally, gathering a more comprehensive reaction database would ensure broader coverage of the total possible reactions, ideally leading to improved model performance. Another avenue worth investigating is the use of reinforcement learning agents, which could iteratively select transformations, potentially generating more accurate results.

In conclusion, our exploration dove into the fascinating complexity of how chemical compounds transform within the human gut, revealing both exciting possibilities and important limitations in our current understanding.

8. Individual Contributions

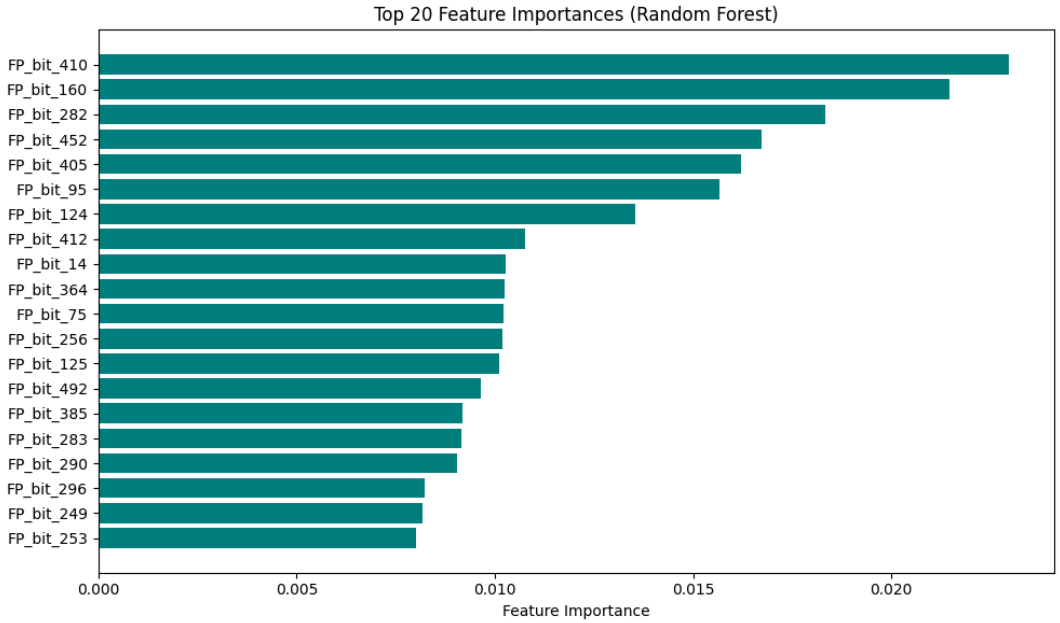
Raghav Sriram: I handled notebook hosting on Kaggle and wrote code to train, develop, and test the models used in this project. I also ensured the notebook was well organized, easy to follow, and that our pipeline was well detailed and clear. I prepared the project.zip file by creating the project.html, README.txt file, and organizing the datasets. Furthermore, I contributed to writing sections 4 and 5 of the project proposal and assisted Yashas in editing the slideshow for the final presentation. For the final report, I helped write sections 3 and 5a and copy edited the final report so that it was ready for submission.

Rishit Sharma: I contributed extensively to the visualization aspects of this project, creating the code for all figures, diagrams, and tables showcased in the final report, and made several suggestions and improvements to the notebook code. I took the lead in interpreting the output of our experiments, ensuring that the results were clearly understood and integrated into our conclusions. I authored the majority of the final report's written content, including sections 1, 3, 4a, 4b, 5, and 6.

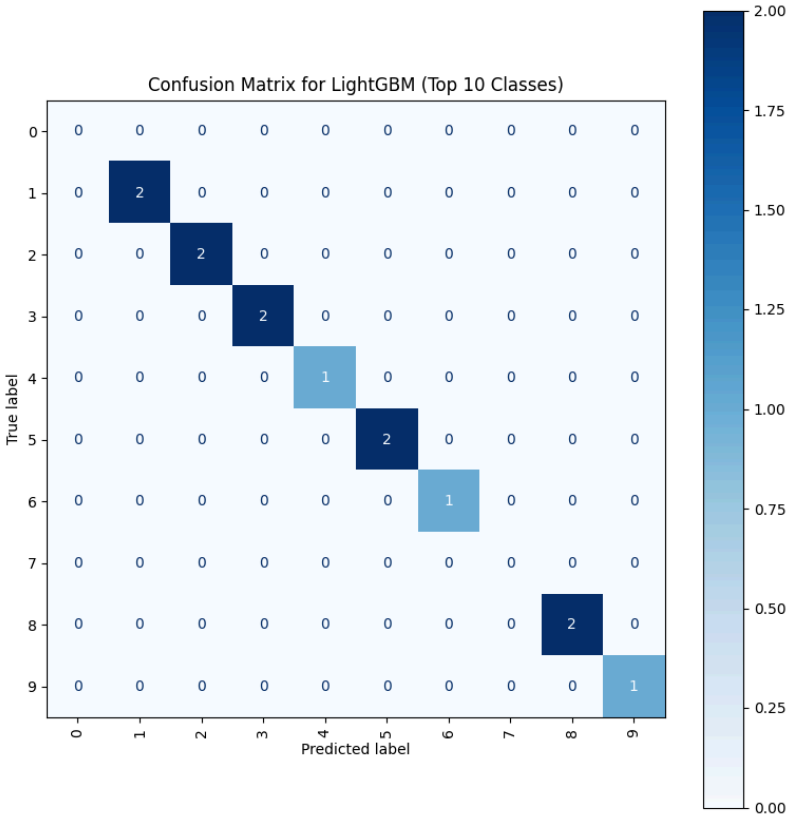
Yashas Raman: I assisted Raghav in writing the code for data preprocessing as well as all three of our models used in this project. I also oversaw and edited the final project report to ensure it was well-organized and ready for submission, along with adding useful contributions to essentially every section in the report. Additionally, I helped make the final presentation slideshow, ensuring it effectively showcased our project's goals, methods, and results in a clear and engaging manner.

Appendix

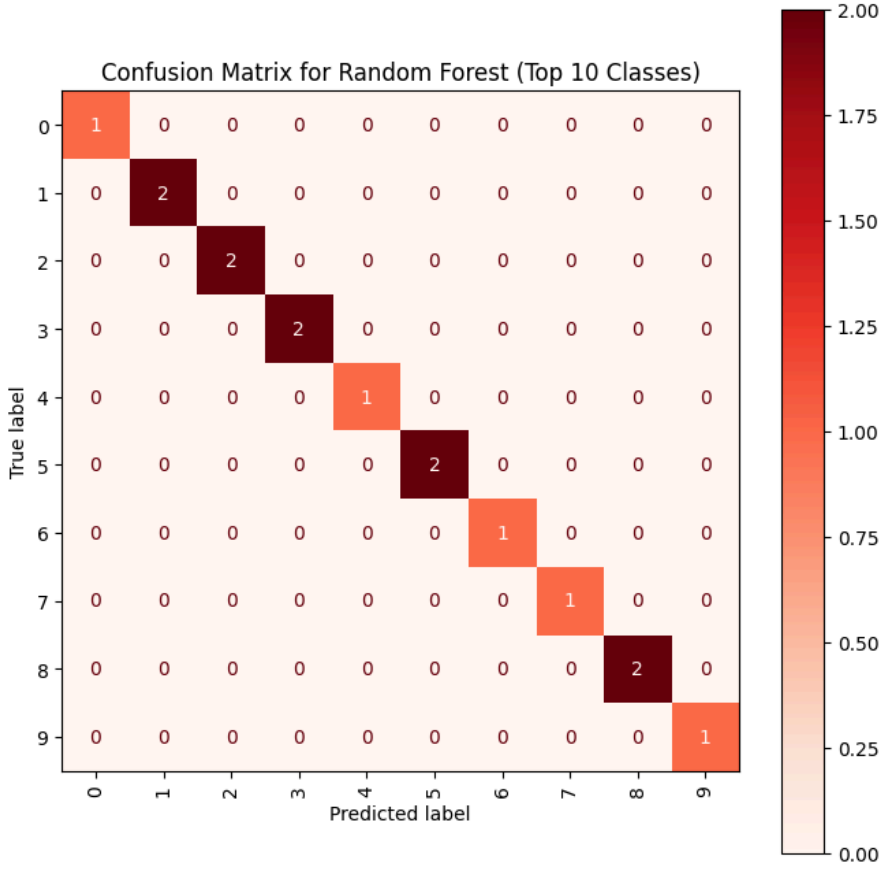
This figure shows the top 20 features integrated into our Random Forest models, giving us a closer look at feature importance:



This figure is a confusion matrix showing the number of correct predictions for each of the first 10 classes in our LightGBM mode. You can see that the LGBM model does not predict accurately as there are not correct predictions for classes 0 through 7. This trend appeared for the rest of the classes:



This figure is a confusion matrix showing the number of correct predictions for each of the first 10 classes in our Random Forest (200 estimators) model. We can see that our Random Forest model either equalled or outperformed our LightGBM model in every class:



Works Cited

Schwaller, P., Gaudin, T., Lanyi, D., Bekas, C., & Laino, T. (2018). Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *arXiv preprint*. https://arxiv.org/abs/1811.02633

Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling, 50*(5), 742–754. https://doi.org/10.1021/ci100050t

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., ... & Pande, V. (2018). MoleculeNet: A benchmark for molecular machine learning. *Chemical Science, 9*(2), 513–530. https://doi.org/10.1039/C7SC02664A