Recommended or Not?: Applying Supervised Machine Learning Models to Understand UofT Course Evaluations

Emily Su, Luka Tosic, Karen Wong*

1 Introduction

Course evaluations play a significant role in how students select courses and how departments assess teaching quality, yet these evaluations are often only accessible in anecdotal form. Our paper investigated how supervised machine learning models can be used to analyze and predict student perceptions of university courses based on course evaluation data and course descriptions. We evaluated different baseline models like Gaussian Discriminant Analysis, Logistic Regression, Linear Regression, Decision Tree, and a Feedforward Neural Network with an Adam Optimizer and constructed a Majority Voting Ensemble with a Decision Tree, a Logistic Regression model, and a Neural Network. Inspired by The Varsity's reporting on disparities in course ratings across departments, our goal is to classify whether a course is likely to be recommended by students using classification models with a defined rating threshold [1]. Currently, many machine learning-based course recommender systems in research overlook other students' experiences taking a given course; our proposed system differs by directly taking student course evaluation data scraped from the University of Toronto (UofT)'s database to learn how students themselves rate and consider courses, as well as course descriptions [2], [3]. The tabular format of UofT course evaluations makes it difficult to gauge whether other students recommend a course and to understand the overall evaluation of a course over multiple instances of its existence. By applying machine learning, we can not only offer a more complete picture of student sentiment but also enable practical and interpretable predictions that can inform both student decision-making and departmental planning. Moreover, we identified features with our models that influence positive course experiences and a course being recommended.

2 Literature Review

In one study, Arcinas et al created a course recommendation system using PCA for feature selection to reduce the number of dimensions in their dataset and 3 different classifiers: KNN, AdaBoost, and Naive Bayes [4]. The system was trained and tested on 500 students' academic records with different attributes like commute time, study hours, health status, etc, and they found that AdaBoost had the highest accuracy of 99.5% [4]. However, Arcinas et al's system requires attributes like health status, which may not always be readily available and/or students may not be willing to give out personal information. Also, there is the potential that Arcinas et al's system may not generalize well with a larger sample size. Our system does not require student's identifying information to be used. In another study by San et al, they proposed a hybrid course recommendation system that combines Random Forest, Naive Bayes, and Support Vector Machine (SVM) classifiers into an ensemble model that takes into account the student's needs and the course descriptions [5]. They trained their system using a dataset from a college that included course descriptions and course codes, the type and description of the development needs a subject covers. In the ensemble modelling stage, they used a weighted voting system where a classifier's weight is determined by its F1 score [5]. San et al found that SVM gave them the highest classification accuracy of 51.1%, but their ensemble model had an F1 score of 38.9% [5]. The study by Kord, Aboelfetouh, and Shohieb [6] analyzes SVC, KNN, regression, decision tree, and several other machine learning and deep learning algorithms to create a prediction model to forecast expected student grades and a recommender model that uses student grades, among other features as to recommend courses students may perform well in. The paper finds that SVC stood out, achieving the highest multi-class accuracy (approx. 78.04%) and an F1-score of approximately 75.37%, making it a potent choice for this type of recommender [6].

*Authors are listed alphabetically by last name. Contributions of authors and course evaluation screenshots can be found in the Appendix.

Limitations of Arcinas et al and San et al's studies are that their datasets do not include the experiences of other students who have taken a course [4], [5]. Students' experiences in a course could vary based on factors outside of the material taught in the course, such as the instructor's teaching style, and student feedback can be valuable in recommending a course or not. Our system incorporates the experiences of students, which was not done previously by San et al [5] and Arcinas et al [4] alongside course descriptions like San et al [5] did.

3 Problem Formulation

Figure 11 shows the project architecture of how we went from our data to analysis in our paper.

3.1 Data

Our data is originally from UofT's Course Evaluation Website for the Faculty of Arts & Sciences [2]. However to get our final dataset, we combine already scraped data from The Varsity [1] and Reddit [7] for the course evaluations and UofT's Faculty of Arts & Sciences Academic Calendar [3] for the course descriptions. During the data pre-processing stage, we filtered the datasets to only look at Computer Science courses (n = 6779) and constructed a new column called "recommended", a binary column that is 1 if the column, "I would recommend this course" is ≥ 4, otherwise it is 0, which corresponds to the course being somewhat recommended or not recommended. In Figure 6b, the distribution for the "I would recommend this course" peaks at around 4. In order to prevent class imbalance between recommended and non-recommended courses for our classification models' target variable, we chose the value 4 as the threshold. After applying this threshold, we can see in Figure 6a that the number of observations in each class is similar.

We used the following columns from our dataset for our models: course, term, year, item 1 (i found the course intellectually stimulating), item 2 (the course provided me with a deep understanding of the subject manner), item 3 (the instructor created a course atmosphere that was condusive to my learning), item 4 (course projects, assignments, tests, and/or exams improved my understanding of the course material), item 5 (course projects, assignments, tests, and/or exams provided opportunity for me to demonstrate an understanding of the course material), item 6 (overall, the quality of my learning experience in the course was:), instructor generated enthusiasm, course workload, i would recommend this course, the instructor's last name, and course description. Further discussion on the ethics of including the lecturer's last name can be found in the Appendix under "Ethics Discussion". We used a Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer to encode the categorical columns like the instructor's last name, course description, term, and course. The vectorizer filtered out common English words and identified important and unique words for each column. We perform dimensionality reduction with the vectorizer by setting it so it only obtains the top 100 features of the dataset based on its TF-IDF score for each categorical column for logistic regression and linear regression, and 80,000 features for the other models. We split our datasets into 80% for the training set and 20% for the test set for all models except for the neural network, where the test set was 10% of the original dataset and the other 10% was for the validation set.

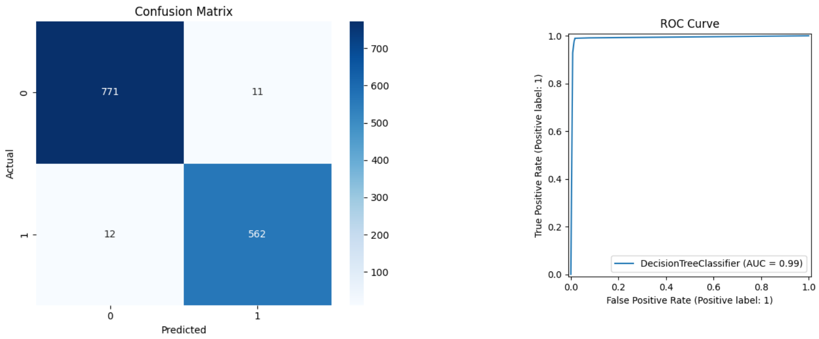
3.2 Models

We chose the following models as our baseline: Decision Tree, Linear Regression, Logistic Regression, Gaussian Discriminant Analysis, and Feedforward Neural Network with Adam Optimizer and L2 Regularization. The purpose of the Gaussian Discriminant Analysis model was to look at how the continuous variables in our dataset impacted whatever or not a course was recommended and how it compared to other classifiers like logistic regression. Linear regression was implemented with a threshold on its output and logistic regression were implemented to look at features that could influence the outcome if a course is recommended or not. The logistic regression model, decision tree, and neural network were used to classify if a course would be recommended or not by a student for our ensemble model, which uses a majority voting system. In the ensemble, all models made predictions on a test set, whereas the prediction of the ensemble for a datapoint would be based on the mode of the predictions of the three models. More specifically, the decision tree used the Gini coefficient as the criterion and had a max depth of 13, chosen based on the following formula $log_2(6000)$, a heuristic value based on the number of rows in our feature matrix. The linear regression model was implemented using the Ordinary Least Squares (OLS) method without explicit regularization (i.e., penalty term = 0). For the logistic regression model, it used the `liblinear` solver, which is suitable for smaller datasets and binary classification tasks, along with L2 regularization to reduce overfitting. The inverse regularization strength was set to $C = 1.0$, meaning no strong penalty was applied. For GDA, the model was implemented using the maximum likelihood estimation approach. It assumes that each class follows a multivariate Gaussian distribution with a shared covariance matrix across classes. For our feedforward neural network, it uses Adam as an optimization objective with a l2 regularization penalty of $\lambda = 0.001$ to prevent overfitting. Adam is an optimization algorithm based on stochastic gradient descent that updates the weights of a neural network iteratively as it is trained and is ideal for large datasets and/or datasets with many parameters [8], [9]. The number of hidden layers, 50, was chosen after evaluations with a validation dataset since it gives the lowest average loss after 100 epochs. For the batch size, we chose 10. The models are compared based on metrics such as accuracy on the dataset (percentage of correct predictions), F1 score, precision, and recall. Out of all our models, linear regression and logistic regression are the most interpretable models alongside decision trees. For our analysis, we looked at the coefficients for the linear regression and logistic regression models to understand the impact of different features on their output.

4 Results

4.1 Decision Tree

Figure 1: Classification with Decision Tree



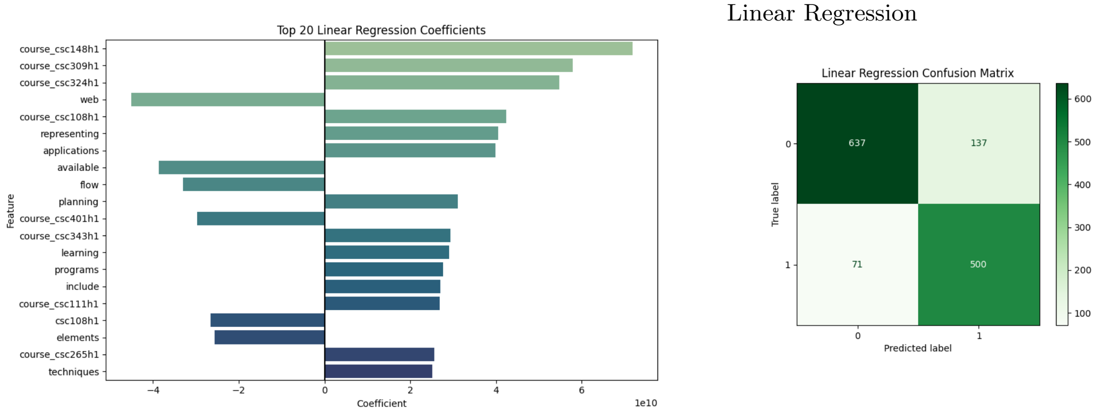
(a) Decision Tree Confusion Matrix

(b) Decision Tree ROC Curve

Figure 10 shows common performance metrics of a decision tree classifier. The high values across all metrics point to a model that rarely misclassifies, as seen by the roughly 98% true positive rate, or recall. While the dataset does boast some class imbalance at around 57% of observations being of one class, both the accuracy and precision are unaffected, and the model classifies well regardless. Figures 1a and 1b both also point to a very highly accurate model. The ROC curve, in particular, hugs the top left of the graph, indicating a near-perfect classifier for this problem.

4.2 Linear Regression

Figure 2: Classification with Linear Regression



(a) Linear Regression Coefficients Plot on Test Set

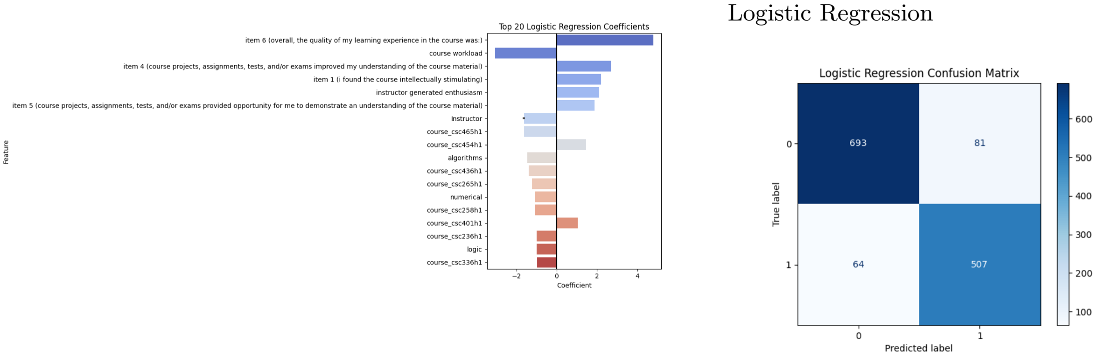
(b) Confusion Matrix on Test Set for Linear Regression

Figure 2a shows the top coefficients ranked by absolute value. Positive coefficients increase the likelihood of a course being recommended, while negative coefficients decrease it. The strongest positive predictors include `course_csc148h1`, course_csc309h1, and course_csc324h1. Strong negative predictors include the keywords web, available, flow.

Figure 2b resulted in 637 true negatives, 500 true positives, 137 false positives, and 71 false negatives, giving an accuracy of 84.55%. Linear regression shows less favorable trade-offs between precision and recall, particularly for the negative class, suggesting it is less effective for binary classification tasks.

4.3 Logistic Regression

Figure 3: Classification with Logistic Regression



(a) Logistic Regression Coefficients Plot on Test Set

(b) Confusion Matrix on Test Set for Logistic Regression

Figure 3a presents the top coefficients ranked by their absolute values. The strongest positive predictors are the features related to the overall quality of the learning experience in the course, course workload, the extent to which course projects and assignments improved understanding of the material, and intellectual stimulation provided by the course. These suggest that students are more likely to recommend courses that they find engaging and well-structured. The strongest negative predictors are course workload, instructors and course_csc465, indicating that they reduce the probability of recommending a course.

Figure 3b resulted in 693 true negatives, 507 true positives, 81 false positives, and 64 false negatives. This corresponds to an overall precision of 89.22%.

4.4 Gaussian Discriminant Analysis (GDA)

Figure 4: Confusion Matrix on Test Set for GDA

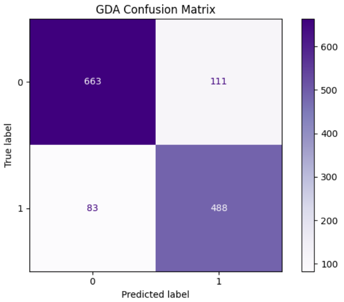
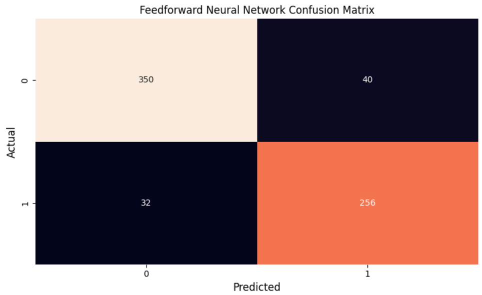


Figure 4 resulted in 663 true negatives, 488 true positives, 111 false positives, and 83 false negatives. This provided an overall accuracy of 85.57%. These results indicate that the GDA model performs reliably, with a slight trade-off favouring recall for the recommended class, beneficial in scenarios where missing a good course is more costly than recommending a less favourable one. However, the 111 false positives suggest that some non-recommended courses are being misclassified as recommended, which could be addressed through feature refinement, threshold tuning, or model comparison with alternative classifiers.

4.5 Feedforward Neural Network with Adam Optimizer and L2 Penalty

Figure 5: Classification with Neural Network and Ensemble

(a) Confusion Matrix on Test Set for Neural Network



(b) Comparison of baseline models' test accuracies with majority voting ensemble

Model	**Accuracy**
Decision Tree	99.11%
Logistic Regression	89.68%
Neural Network	88.57%
Ensemble	94.99%

Figure 12 shows that the accuracy of the neural network was 0.89 or 89% with a F1 score of 0.91 or 91% for the class 0, which means that students do not recommend or somewhat recommend the course. On the other hand, the F1 score for the class 1, which means that students recommend a course, is 0.88 (88%). Figure 5a shows that for class 0, $\frac{350}{390}$ or 89.74% of all courses that fall under class 0 was classified correctly. For class 1, $\frac{256}{288}$ or 88.89% of its courses were classified correctly.

4.6 Ensemble with Majority Voting System

Figure 5b shows that after being trained on 80% of the data and tested on 20% of the data, the decision tree had the highest prediction accuracy of 99.11% on the test data while the ensemble had the second highest prediction accuracy of 94.99%.

5 Conclusions

5.1 Discussion

We found that our decision tree model performed the best in terms of predicting if a CS course is recommended or not (99.11% during training for our ensemble) with our majority voting ensemble performing the second best with 94.99% accuracy. We saw that with our logistic regression model, the log-odds of a course being recommended increased if students were intellectually stimulated and had a good learning experience in a course, if the assignments, project, and tests/exams from the course are fair and improved understanding of the course content, and instructors are enthusiastic. However, we saw that the log-odds of a course being recommended decreased with increased workload. Specifically with CS courses, the log-odds of a course being recommended decreases if it is a logic-based/algorithm-based course.

5.2 Next Steps/Further Development

Some limitations of our analysis were that we only trained the model on CS courses at UofT. Our models can also be further refined to allow users to select features they want, or to classify course recommendations based on a specific year, term, or course level. PCA could have also been used for dimensionality reduction and we could have used boosting ensembles like AdaBoost, which could have given us higher accuracy and they were used in Arcinas et al's study [4]. Moreover, as seen in San et. al and Kord. et al, Special Vector machines for Classification (SVCs) perform very well for such classification tasks, perhaps owing to their effective handling of sparse and correlated data.

Our models could have had high accuracy due to binary classification being an easy problem; as such, switching to a multiclass classification problem could provide a more demanding task for the chosen models. Moreover, being able to classify course recommendations into multiple levels would lend itself to more versatile and practically useful predictions.

We omitted Naive Bayes as part of our analysis since Naive Bayes was often a worse-performing model in previous studies [4], [5], which could be due to its independence assumption. Our GDA model suffered in terms of performance, however, a hybrid model with GDA and Naive Bayes could have better prediction power than the two individually.

6 Appendix

6.1 Contributions

All authors contributed equally to the project. Emily worked on the ideation and planning of the project and its architecture, conducted literature review, built the feedforward neural network and the ensemble, compiled and wrote a script to pre-process the data, reviewed and made edits to other models, managed the codebase, analyzed and interpreted results, and wrote the paper. Luka performed an exploratory data analysis, built the decision tree model, reviewed other models, organized team meetings, wrote the introduction and ethics sections and contributed to the literature review, problem formulation, results, and conclusions sections of the paper, as well as editing the paper. Karen built models (i.e. linear regression, logistic regression and gaussian discriminant analysis), reviewed other models, analyzed and interpreted results, and contributed in editing the paper.

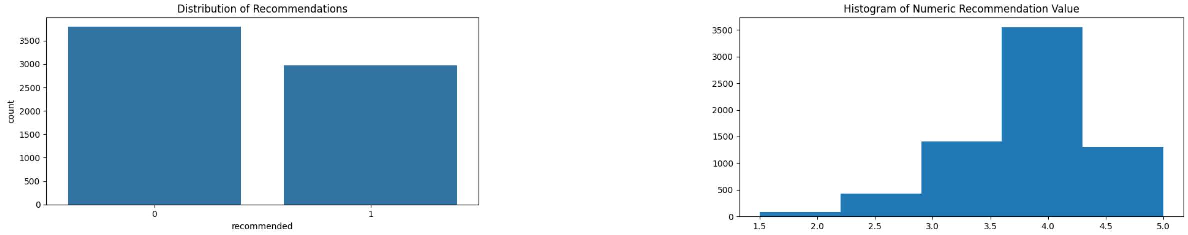
6.2 Ethics Discussion

Each observation in our dataset includes the last name of the lecturer who taught the course. The original dataset from The Varsity has this information removed, but it is still present on the University's course evaluation website. We chose to include the last names from the scraped dataset with deliberate consideration to be outlined as follows. The identity of the lecturer can be a powerful predictive tool for machine learning models, and is one that is particularly of importance to students as a lecturer is one of the most important aspects of a course. We included this feature despite understanding the potential ethical risks and implications associated with the decision, discussed below.

All the numerical features of this dataset as well as the target we are classifying are created from opinion-based metrics that carry with them the biases of all the people who gave responses to them. As such, both conscious and unconscious bias about the identity of the lecturer has the potential to make its way into the data. As such, our models have the potential to find or eke out and even amplify negative associations between course evaluation scores and certain professors; this is especially damaging when these associations are linked to a lecturer's racial or ethnic identity, sex, or otherwise and not to teaching ability. It is important to note that a correlation between a lecturer's identity and low recommendation score or level does not imply a causal relationship between the two. Another ethical consideration is privacy. Despite being on a UofT-maintained database, lecturers did not consent to data embedded with their identities being used. A course recommender system can incentivize the ranking of lecturers or targeting of those with low correlated scores. Both of these scenarios can be potentially damaging to careers and the mental health of the affected. We have removed the instructors' last names from our graphs to maintain anonymity.

6.3 Additional Graphs

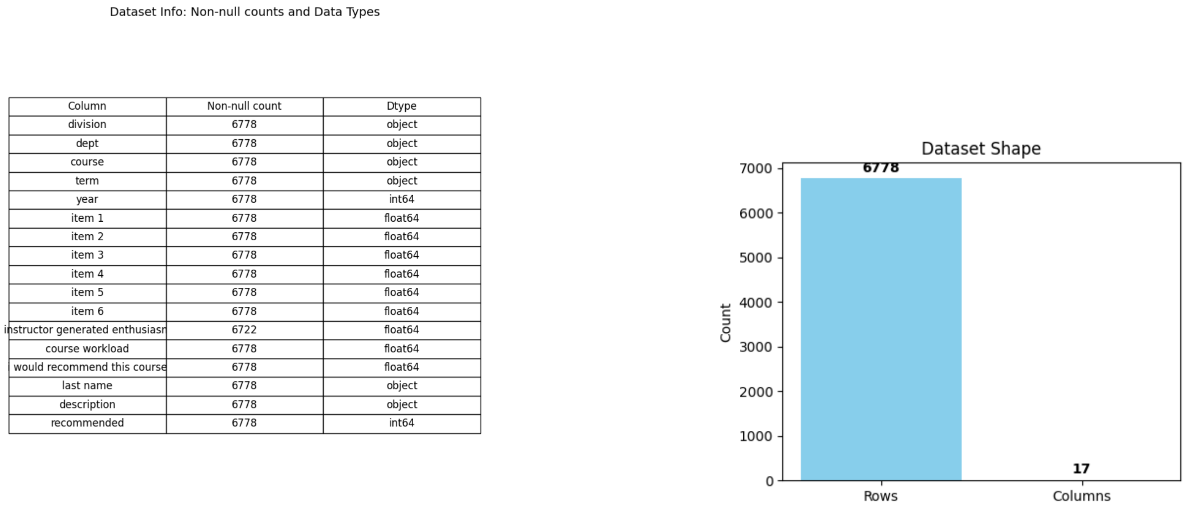
Figure 6: Distribution/count of target variable for recommendation



(a) Count for each class

(b) Distribution of target on the range of 1 to 5

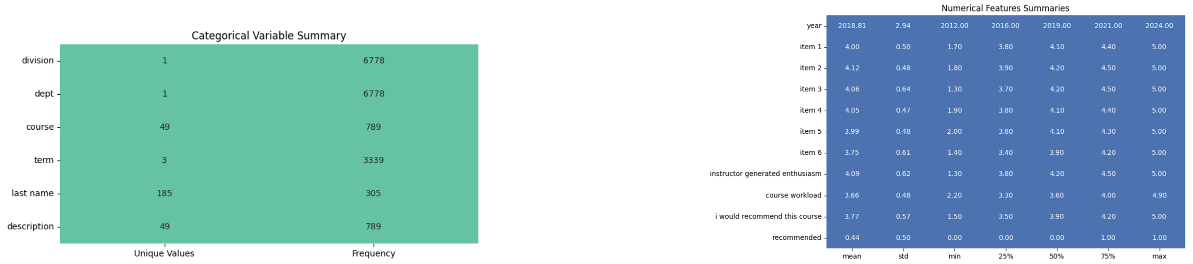
Figure 7: Dataset Info and Shape



Dataset Info: Non-null counts and Data Types

Column	Non-null count	Dtype
division	6778	object
dept	6778	object
course	6778	object
term	6778	object
year	6778	int64
item 1	6778	float64
item 2	6778	float64
item 3	6778	float64
item 4	6778	float64
item 5	6778	float64
item 6	6778	float64
instructor generated enthusiasm	6722	float64
course workload	6778	float64
would recommend this course	6778	float64
last name	6778	object
description	6778	object
recommended	6778	int64

Figure 8: Dataset Summary Statistics



Categorical Variable Summary

	Unique Values	Frequency
division	1	6778
dept	1	6778
course	49	789
term	3	3339
last name	185	305
description	49	789

Numerical Features Summaries

	mean	std	min	25%	50%	75%	max
year	2018.81	2.94	2012.00	2016.00	2019.00	2021.00	2024.00
item 1	4.00	0.50	1.70	3.80	4.10	4.40	5.00
item 2	4.12	0.48	1.80	3.90	4.20	4.50	5.00
item 3	4.06	0.64	1.30	3.70	4.20	4.50	5.00
item 4	4.05	0.47	1.90	3.80	4.10	4.40	5.00
item 5	3.99	0.48	2.00	3.80	4.10	4.30	5.00
item 6	3.75	0.61	1.40	3.40	3.90	4.20	5.00
instructor generated enthusiasm	4.09	0.62	1.30	3.80	4.20	4.50	5.00
course workload	3.66	0.48	2.20	3.30	3.60	4.00	4.90
i would recommend this course	3.77	0.57	1.50	3.50	3.90	4.20	5.00
recommended	0.44	0.50	0.00	0.00	0.00	1.00	1.00

Figure 9: Project architecture

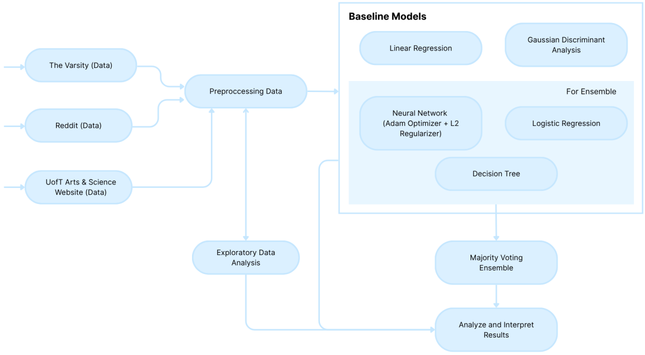
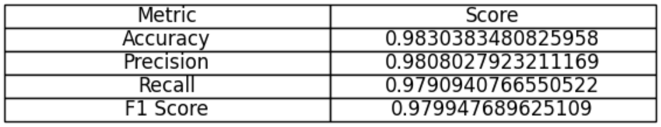

Baseline Models
Linear Regression
Gaussian Discriminant Analysis
The Varsity (Data)
Preproccessing Data
For Ensemble
Reddit (Data)
Neural Network (Adam Optimizer + L2 Regularizer)
Logistic Regression
UofT Arts & Science Website (Data)
Decision Tree
Exploratory Data Analysis
Majority Voting Ensemble
Analyze and Interpret Results

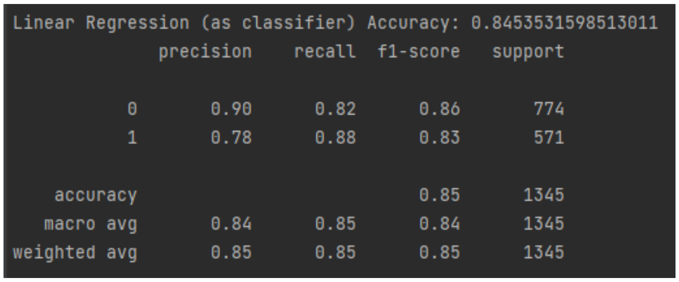
Figure 10: Decision Tree Summary Table



Metric	Score
Accuracy	0.9830383480825958
Precision	0.9808027923211169
Recall	0.9790940766550522
F1 Score	0.979947689625109

Figure 11: Classification Reports

(a) Classification Report on Test Set for Linear Regression

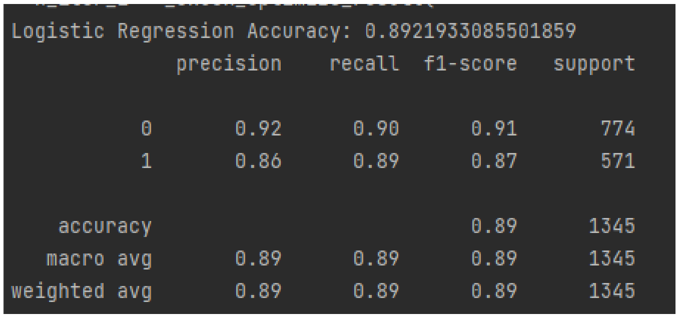


```
Linear Regression (as classifier) Accuracy: 0.8453531598513011
              precision    recall  f1-score   support

           0       0.90      0.82      0.86       774
           1       0.78      0.88      0.83       571

    accuracy                           0.85      1345
   macro avg       0.84      0.85      0.84      1345
weighted avg       0.85      0.85      0.85      1345
```

(b) Classification Report on Test Set for Logistic Regression

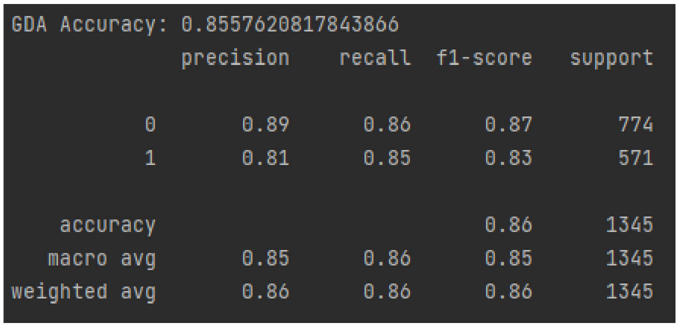


```
Logistic Regression Accuracy: 0.8921933085501859
              precision    recall  f1-score   support

           0       0.92      0.90      0.91       774
           1       0.86      0.89      0.87       571

    accuracy                           0.89      1345
   macro avg       0.89      0.89      0.89      1345
weighted avg       0.89      0.89      0.89      1345
```

(c) Classification Report on Test Set for GDA



```
GDA Accuracy: 0.8557620817843866
              precision    recall  f1-score   support

           0       0.89      0.86      0.87       774
           1       0.81      0.85      0.83       571

    accuracy                           0.86      1345
   macro avg       0.85      0.86      0.85      1345
weighted avg       0.86      0.86      0.86      1345
```

Figure 11a shows in the non-recommended class (0), precision was 0.90, recall 0.82, and F1 score 0.86, indicating strong precision but lower recall due to more false positives. For the recommended class (1), precision was 0.78, recall 0.88, and F1 score 0.83, reflecting better recall and fewer false negatives but at the cost of more false positives. Macro-averaged metrics (precision = 0.84, recall = 0.85, F1 = 0.84) and weighted averages (≈ 0.85) show balanced performance overall.

Figure 11b shows that in the not recommended class (0), the model achieved a precision of 0.92, recall of 0.90, and F1 score of 0.91, reflecting an excellent ability to correctly reject non-recommended courses while keeping false positives low. For the recommended class (1), the precision of the model was 0.86, the recall was 0.89, and the F1 score was 0.87, demonstrating strong performance with a slightly higher tendency to miss fewer positive cases than GDA (lower count of false negatives). The macro-averaged metrics (precision = 0.89, recall = 0.89, F1 = 0.89) and weighted averages (all ≈ 0.89) indicate balanced performance across classes, even with the slight class imbalance present.

Figure 11c shows in the not recommended class (0), the model achieved a precision of 0.89, recall of 0.86, and F1 score of 0.87, showing strong ability to correctly identify non-recommended courses while minimizing false alarms. For the recommended class (1), the model's precision was 0.81, recall was 0.85, and F1 score was 0.83, indicating good identification of recommended courses with slightly more false positives than false negatives. The macro-averaged metrics (precision = 0.85, recall = 0.86, F1 = 0.85) suggests balanced performance across both classes, and the weighted averages (all ≈ 0.86) confirm stability given the class distribution.

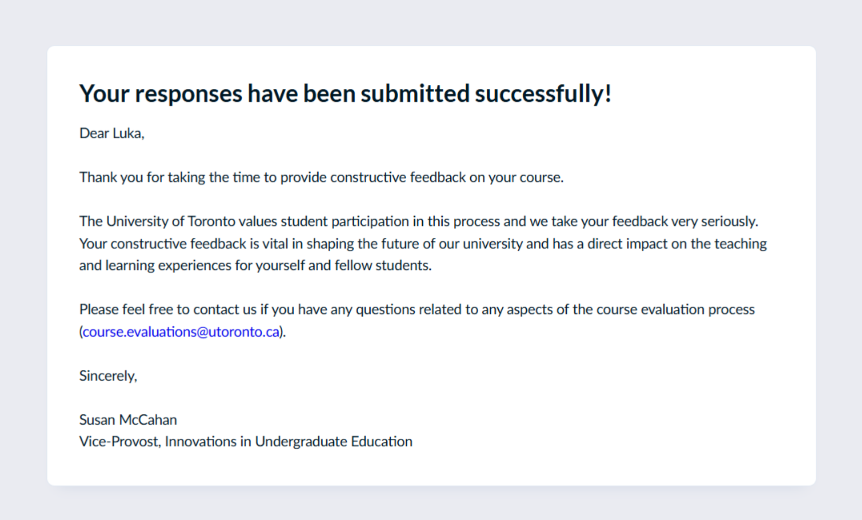
Figure 12: Classification Report on Test Set for Neural Network

Metric	**Label**	**Score**
Precision	0	0.92
	1	0.86
Recall	0	0.90
	1	0.89
F1-score	0	0.91
	1	0.88
Accuracy		0.89

6.4 GitHub Repository

All data, the code, and any additional graphs for the project can be found here: https://github.com/uoft-course-eval-team/course-recommender

6.5 Course Evaluation Screenshots



Your responses have been submitted successfully!

Dear Luka,

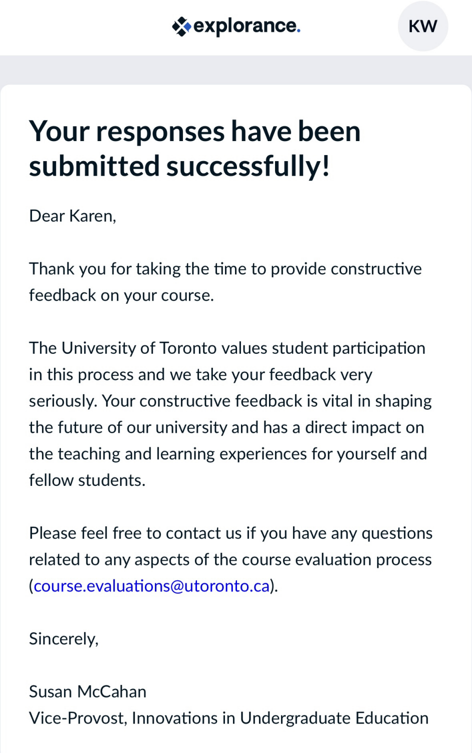
Thank you for taking the time to provide constructive feedback on your course.

The University of Toronto values student participation in this process and we take your feedback very seriously. Your constructive feedback is vital in shaping the future of our university and has a direct impact on the teaching and learning experiences for yourself and fellow students.

Please feel free to contact us if you have any questions related to any aspects of the course evaluation process (course.evaluations@utoronto.ca).

Sincerely,

Susan McCahan
Vice-Provost, Innovations in Undergraduate Education



Your responses have been submitted successfully!

Dear Karen,

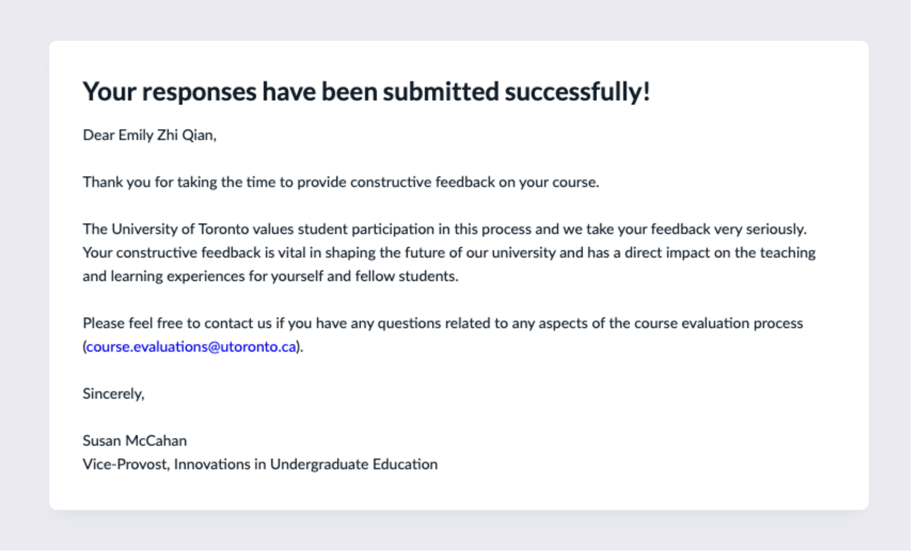
Thank you for taking the time to provide constructive feedback on your course.

The University of Toronto values student participation in this process and we take your feedback very seriously. Your constructive feedback is vital in shaping the future of our university and has a direct impact on the teaching and learning experiences for yourself and fellow students.

Please feel free to contact us if you have any questions related to any aspects of the course evaluation process (course.evaluations@utoronto.ca).

Sincerely,

Susan McCahan
Vice-Provost, Innovations in Undergraduate Education



Your responses have been submitted successfully!

Dear Emily Zhi Qian,

Thank you for taking the time to provide constructive feedback on your course.

The University of Toronto values student participation in this process and we take your feedback very seriously. Your constructive feedback is vital in shaping the future of our university and has a direct impact on the teaching and learning experiences for yourself and fellow students.

Please feel free to contact us if you have any questions related to any aspects of the course evaluation process (course.evaluations@utoronto.ca).

Sincerely,

Susan McCahan
Vice-Provost, Innovations in Undergraduate Education

6.6 Material Referenced for Development

Neural Network Script

- https://www.datacamp.com/tutorial/pytorch-tutorial-building-a-simple-neural-network-from-scratch
- https://scikit-learn.org/stable/modules/generated/sklearn.compose.ColumnTransformer.html
- https://numpy.org/doc/2.1/reference/generated/numpy.nan_to_num.html
- https://datascience.stackexchange.com/questions/45916/loading-own-train-data-and-labels-in-dataloader-using-pytorch
- https://docs.pytorch.org/tutorials/beginner/basics/quickstart_tutorial.html
- https://machinelearningmastery.com/develop-your-first-neural-network-with-pytorch-step-by-step/
- https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ [8]

Data Cleaning Script

- https://stackoverflow.com/questions/56358888/how-to-remove-https-links-from-a-string-column-in-pandas
- https://stackoverflow.com/questions/39782418/remove-punctuations-in-pandas
- https://stackoverflow.com/questions/68759305/which-pattern-was-matched-among-those-that-i-passed-through-a-regular-expression
- https://www.geeksforgeeks.org/python/pattern-matching-python-regex/
- https://stackoverflow.com/questions/21546739/load-data-from-txt-with-pandas
- https://github.com/learnbyexample/py_regular_expressions/blob/master/py_regex.md
- https://flexiple.com/python/python-regex-replace
- https://stackoverflow.com/questions/22588316/pandas-applying-regex-to-replace-values
- https://stackoverflow.com/questions/68759305/which-pattern-was-matched-among-those-that-i-passed-through-a-regular-expression
- https://www.geeksforgeeks.org/python/pattern-matching-python-regex/
- https://pandas.pydata.org/docs/reference/api/pandas.Series.str.replace.html
- https://www.geeksforgeeks.org/python/ways-to-apply-an-if-condition-in-pandas-dataframe/

Decision Trees Script

- https://www.w3schools.com/python/python_ml_decision_tree.asp
- https://scikit-learn.org/stable/modules/tree.html
- https://scikit-learn.org/stable/modules/ensemble.html#random-forests

Linear Regression Script

- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- https://scikit-learn.org/stable/modules/preprocessing.html
- https://www.geeksforgeeks.org/machine-learning/linear-regression-python-implementation/
- https://www.w3schools.com/python/python_ml_linear_regression.asp

Logistic Regression Script

- https://www.w3schools.com/python/python_ml_logistic_regression.asp
- https://www.geeksforgeeks.org/understanding-logistic-regression/
- https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://www.geeksforgeeks.org/understanding-logistic-regression/

Gaussian Discriminant Analysis Script

- https://scikit-learn.org/stable/modules/lda_qda.html
- https://www.geeksforgeeks.org/machine-learning/sklearn-feature-extraction-with-tf-idf/

References

[1] M. Ellis and M. Quarshie, *Student course evaluations rank statistics department lowest, small humanities centres highest*, Feb. 2025. [Online]. Available: `https://thevarsity.ca/2025/02/03/student-course-evaluations-rank-%20statistics-department-lowest-small-humanities-centres-highest/`.

[2] U. of Toronto, *University of toronto course evaluations - faculty of arts science (undergraduate)*, 2025. [Online]. Available: `hhttps://course-evals.utoronto.ca/BPI/fbview.aspx?blockid=OjzZ9-LrM-peMm6q2u&userid=cYQTzF-3fo2ufLLGH26rS-YRliCjyxiGeI8T&lng=en`.

[3] U. of Toronto, *University of toronto arts science academic calendar*, 2025. [Online]. Available: `https://artsci.calendar.utoronto.ca/`.

[4] M. M. Arcinas, M. Meenakshi, P. S. Bahalkar, *et al.*, "An efficient course recommendation system for higher education students using machine learning techniques," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 2, pp. 1468–1475, 2025.

[5] K. K. San, H. H. Win, and K. E. E. Chaw, "Enhancing hybrid course recommendation with weighted voting ensemble learning," *Journal of Future Artificial Intelligence and Technologies*, vol. 1, no. 4, pp. 337–347, 2025.

[6] A. Kord, A. Aboelfetouh, and S. Shohieb, "Academic course planning recommendation and students' performance prediction multi-modal based on educational data mining techniques," *Journal of Computing in Higher Education*, vol. n/a, no. n/a, 2025.

[7] ravines$_{t}rees_{r}ocks$, *We webscrapped over 40,000 rows of u of t course evaluation results*, 2025. [Online]. Available: `https://www.reddit.com/r/UofT/comments/1ihe5tu/we_webscrapped_over_40000_rows_of_u_of_t_cou%20rse/`.

[8] J. Brownlee, *Gentle introduction to the adam optimization algorithm for deep learning*, Jul. 2017. [Online]. Available: `https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/`.

[9] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.