Name: Sai Tejaswi Kondapally

M Number: M20336664

Machine Learning - CMPS 5253- Final Project

Report

Introduction:

With huge improvement of online platforms for education, Massive Open Online Courses (MOOCs) has become one of the major modes of online education, allowing many learners worldwide to access high quality learning content. However, one of the biggest challenges faced by MOOC platforms is the low course completion and certification rate. Many students enroll into one or more courses, but only a very small percentage of them successfully complete the course and earn certification. Predicting whether a learner will complete a course is therefore an important problem for improving student engagement, course design, and overall learning outcomes.

For this project I've chosen to explore the MOOC dataset from Kaggle, which originally contained 416,921 records with 22 features. For this project, a subset of 80,000 random samples was selected for modeling and evaluation to ensure computational feasibility. The dataset includes demographic, academic, and behavioral attributes such as institute, course ID, semester, country, education level, gender, age, and multiple activity-based features like number of events, active days, video plays, chapters accessed, and forum posts. The target variable is "certified", which indicates whether a student has successfully completed the course (1) or not (0). So, this is a binary classification problem.

The dataset consists of the categorical, numerical, and time-based features, and it was found to be a highly imbalanced dataset, with very fewer certified students compared to non-certified ones. In order to handle these challenges, categorical features are performed with one hot encoding or label encoding. Also, some features required scaling, feature engineering, and SMOTE for class imbalance problem were applied before model training. Several machine learning and a deep learning model like Logistic Regression, SVM, KNN, Decision Tree, Random Forest, and a Fully Connected Neural Network (FCNN) were used to train and test this dataset. Each of these models I tried to experiment with base model and hyperparameter tuning and best-found parameters were used as a final model. Since accuracy alone can be misleading for imbalanced datasets, this project focused on precision, recall, and F1-score to compare model performance and identify the best model for prediction.

Background:

MOOCs dataset has been extensively studied in the fields of data mining and learning analytics. Prior research has focused on understanding predicting dropout behavior and identifying factors that contribute to course completion. Both supervised and unsupervised machine learning techniques have been applied in this domain, including clustering based analysis and predictive modeling using behavioral and demographic features. These studies consistently highlight that activity based features such as video interaction, active learning days, and forum posts play a critical role in predicting learner success.

The dataset used in this project, the MOOC Dataset from Kaggle, has therefore received only limited prior analytical attention. However, all these remain exploratory and do not address supervised prediction. And they do not implement a full modeling pipeline involving feature preprocessing, class imbalance handling, hyperparameter tuning, and multi-model comparison. To the best of my knowledge, this work might be the first representing the different experiments on this dataset, providing a comparison of classical machine learning and deep learning models for MOOC certification prediction with Random Forest as the best model with highest F1 score: 0.8148.

Methodology:

As already mentioned, the original MOOC dataset contains 416,921 records with 22 features, from which a subset of 80,000 samples was randomly selected for this project to ensure computational feasibility. As an initial step, all columns were individually checked for duplicate values, missing values (nulls and Nans), and incorrect data types. Three columns were dropped due to the risk of data leakage. The columns dropped are Id (unique identifier), userid_DI (user identifier), and grade (which is numerically correlated with the target variable).

Next, the categorical features like institute, course_id, semester, LoE_DI (level of education), and gender were converted into numerical format using one hot encoding. To reduce the high cardinality of the country column (final_cc_cname_DI), it was grouped into three categories: USA, India, and Other, and then one-hot encoded. The time-based columns start_time_DI and last_event_DI were cleaned, converted into datetime format, and new feature named days_active_total was created from start_time_DI and last_event_DI (feature engineering). This represents the total number of days a student remained active in the course. After feature creation, both timestamp columns were dropped. Additionally, the behavioral features nevents, ndays_act, nplay_video, nchapters, and nforum_posts, which showed skewness, were handled using logarithmic transformation. The age column was also corrected using range based clipping to remove unrealistic values. After data preprocessing and feature engineering, the dataset has expanded from 22 original columns to 44 processed features.

The processed dataset was then split into training (70%), validation (15%), and test (15%) sets. Since all models used for this project are sensitive to feature magnitude, Standard Scaler was applied to all numerical features. Scaling was performed only on the training set and then applied to validation and test sets to prevent data leakage. The target variable certified was found to be highly imbalanced, with significantly less positive samples. To handle this, SMOTE (Synthetic Minority Oversampling Technique) was applied.

All machine learning and deep learning models I used in this project were trained using the same preprocessed, scaled, and SMOTE-balanced training data to ensure fair comparison. For each model, a consistent experimental pipeline was followed: as first step baseline model training next hyperparameter tuning and then final model training using the best parameters, and evaluation on the unseen test set.

Results:

The table below shows the final test-set performance of all final models used in this project. Since the dataset is highly imbalanced, F1-score is treated as the primary evaluation metric, while accuracy is reported as just a supporting metric.

Model	Test Accuracy	Test Precision	Test Recall	Test F1 Score
Logistic Regression	0.9682	0.5307	0.9651	0.6848
Support Vector Machine (RBF)	0.9735	0.5881	0.8698	0.7016
K-Nearest Neighbors	0.9791	0.6642	0.8419	0.7426
Decision Tree	0.9792	0.6530	0.8930	0.7544
Random Forest	**0.9847**	**0.7181**	**0.9419**	**0.8149**
FCNN	0.9812	0.6688	0.9448	0.7830

Logistic Regression:

The final Logistic Regression model was trained using the best hyperparameters obtained through tuning. They are C = 10, max_iter = 500, solver = lbfgs and tolerance = 0.0001. The final test F1 score achieved by this model was 0.6848. The model showed very high recall but low precision, meaning that it successfully identified most certified students but also produced many false positives. The training suggested mild underfitting, indicating that the model was too simple to fully capture the complex patterns in the data.

Support Vector Machine (SVM):

The final SVM model used the RBF kernel with tuned parameters C = 1 and gamma = scale. This model achieved a final test F1-score of 0.7016, which was a clear improvement over Logistic Regression. The performance showed reduced underfitting compared to Logistic Regression, with balanced generalization and no signs of overfitting.

K-Nearest Neighbors (KNN):

The final KNN model used the following optimal parameters: n_neighbors = 1, weights = uniform and p = 1 (Manhattan distance) and achieved a final test F1-score of 0.7426. Compared to Logistic Regression and SVM, KNN showed a noticeable improvement in precision while maintaining strong recall. However, KNN showed slight sensitivity to noise, but no major overfitting was observed.

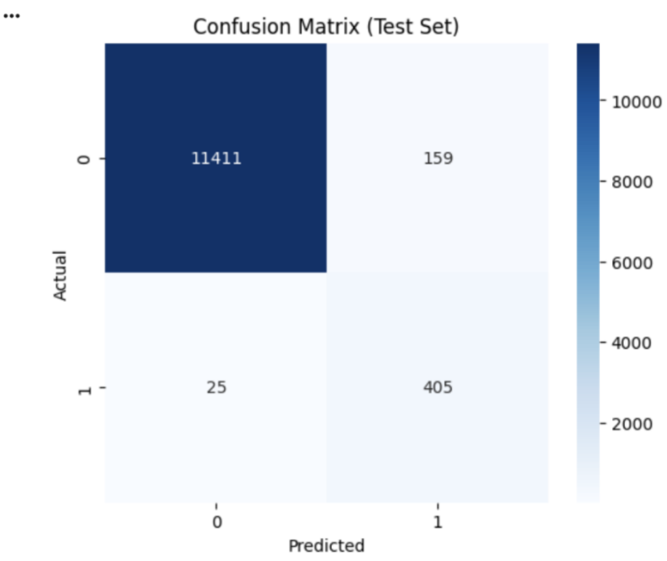
Decision Tree:

The final Decision Tree model was trained with the following tuned parameters: max_depth = 15, min_samples_split = 10, min_samples_leaf = 2 and criterion = entropy achieved a test F1-score of 0.7544, outperforming KNN. The model demonstrated good generalization with minimal overfitting.

Random Forest (Best Performing Model):

The Random Forest model produced the best overall performance among all models. The final tuned hyperparameters were n_estimators = 300, max_depth = None, min_samples_split = 2, min_samples_leaf = 1, max_features = 'sqrt'.This model achieved a test F1-score of 0.8149, which is the highest among all models tested.

It also produced excellent precision and recall balance, with very few false negatives. The ensemble nature of Random Forest significantly reduced overfitting while improving generalization. Based on these results, Random Forest is selected as the best-performing model in this dataset.



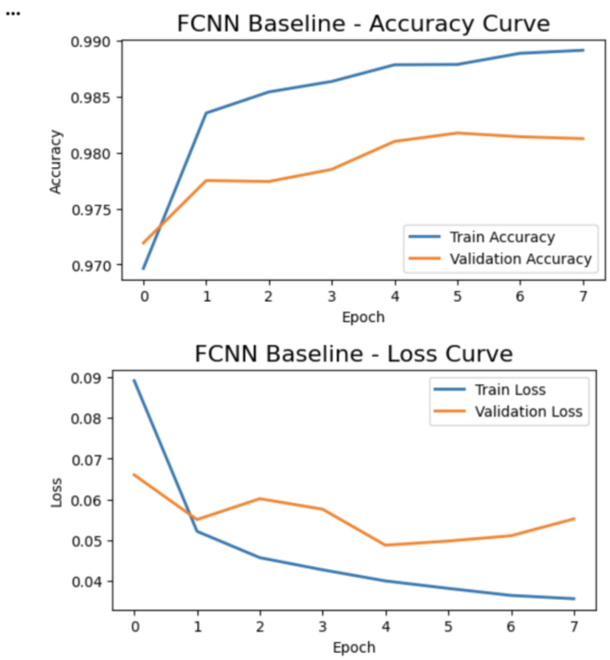
Confusion Matrix of best final model: Random Forest

Fully Connected Neural Network (FCNN):

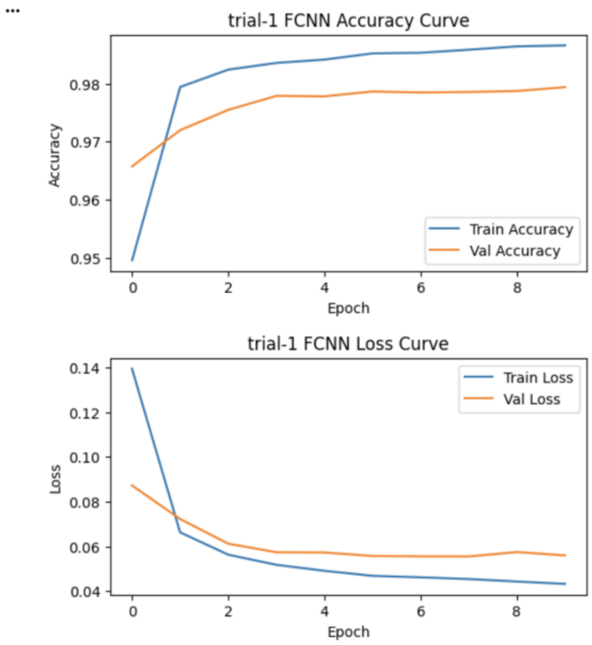
Before training the FCNN, rescaling was applied after SMOTE since synthetic sampling can change the statistical properties of the data. Neural networks perform best when input features are on a controlled scale; otherwise, unstable gradients can cause loss explosion or poor convergence. When I experimented without resampling, the model exploded.

The baseline FCNN consisted of two hidden layers (64 and 32 neurons) with ReLU activation, dropout regularization, binary cross-entropy loss, Adam optimizer (learning rate = 0.001), batch size of 64, and early stopping. This baseline model achieved strong validation performance with good training and validation curves.

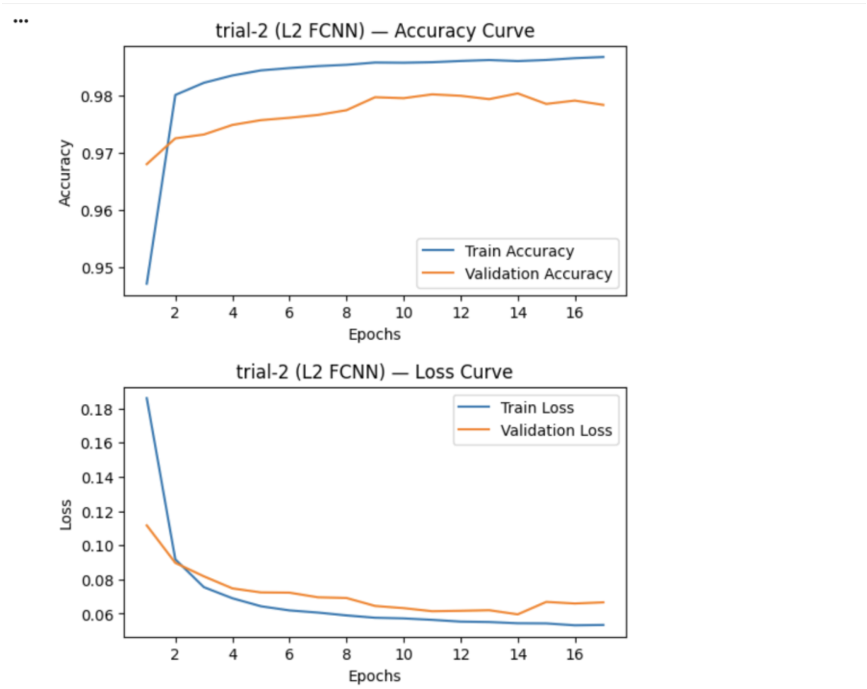
Three experimental trials were conducted to improve generalization: Trial-1 used a smaller network (32 and 16 neurons), Trial-2 applied L2 regularization, and Trial-3 batch normalization. Among these, the Batch Normalization model performed best and was selected as the final FCNN. On the test set, this model achieved a Test F1-score of 0.7830, indicating strong predictive capability with high recall and controlled overfitting. FCNN performed competitively but did not surpass Random Forest.



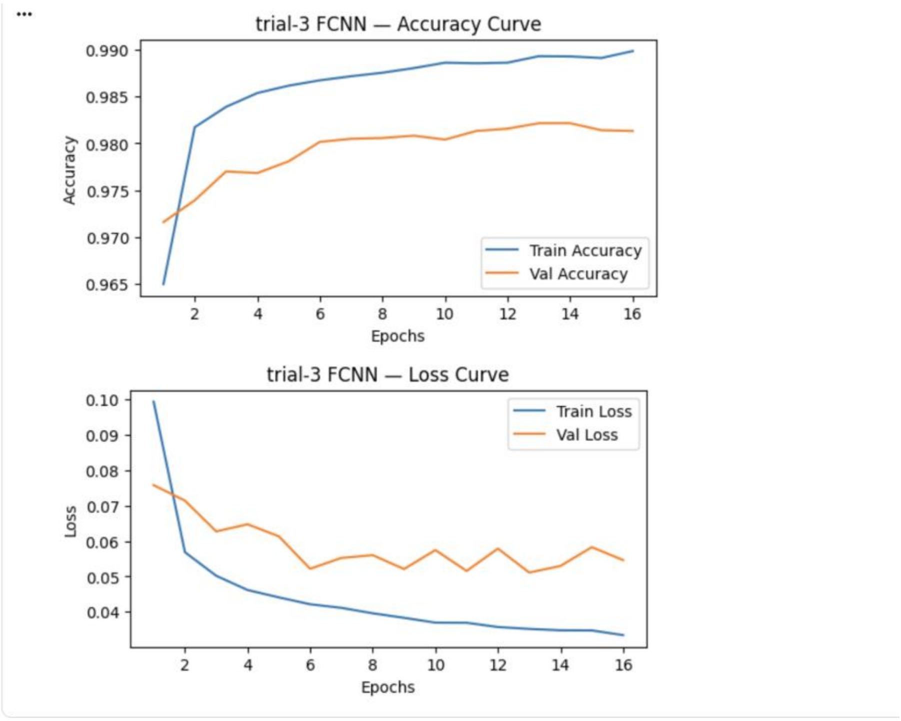
Accuracy and loss curves for the FCNN Base Model



Accuracy and loss curves for the FCNN Trial-1 (smaller network)



Accuracy and loss curves for the FCNN Trial-2 with L2 Regularization



Accuracy and loss curves for the FCNN Trial-3 with Batch Normalization

All models achieved high test accuracy greater than 96%, mainly due to majority-class dominance. The F1 score revealed the true performance differences among the models. Logistic Regression and SVM suffered with low precision. KNN and Decision Tree showed balanced improvements. Random Forest achieved the best overall balance between precision and recall. FCNN performed competitively but did not surpass Random Forest.

Discussion:

The result shows that random forest and FCCN models outperform traditional linear classifiers for the MOOC dataset. One of the major challenges in this project was the severe class imbalance, which significantly affected the model performance. Without proper handling through SMOTE, feature scaling, and feature engineering, it was difficult to achieve meaningful F1-scores. Additionally, several categorical variables required one-hot encoding, and multiple behavioral features (numerical features) required log transformation to correct skewness. Proper preprocessing, scaling, and balanced training played a crucial role in improving the generalization ability of all models. The Random Forest model achieved the best overall performance, while the FCNN also showed strong predictive capability when combined with normalization and smaller network.

Conclusion:

In this project, I had a chance of evaluating multiple machine learning and deep learning models to predict student certification outcomes in MOOCs dataset. The preprocessing pipeline involving encoding, scaling, log transformation, feature engineering, and SMOTE played a critical role in improving model performance on this highly imbalanced binary classification dataset. Among all the models I experimented with, the Random Forest achieved the best overall performance, followed closely by the FCNN, which demonstrated strong learning capability with proper regularization and normalization with a small network. The best model Random Forest has won with a F1 score of 0.8148.

References:

1. https://www.kaggle.com/datasets/kanikanarang94/mooc-dataset
2. https://www.tensorflow.org
3. https://www.kaggle.com/code/alishanmustafa09/kmean-for-clustering
4. https://scikit-learn.org/stable/modules/model_evaluation.html
5. https://keras.io
6. https://keras.io/api/callbacks/early_stopping/

7. https://scikit-learn.org/stable/modules/preprocessing.html