Abstract

Predicting student grades using machine learning is a well-established problem, driven by the need to understand the factors influencing academic performance. Traditionally, most approaches have focused on limited feature sets, such as predicting GPA based solely on SAT scores. However, these models often overlook the complex and complex nature of academic achievement, which can be influenced by a student's past performance, socio-economic background, and study habits.

In this project, a more systematic and nuanced approach is proposed. Rather than relying on a narrow set of predictors, we use a diverse range of features and evaluate performance on regression and classification-based tasks. This methodology allows to exploit the dataset to full.

With the goal of improving academic performance prediction, several machine learning models were evaluated, including decision trees, linear models, and multilayer perceptrons (MLPs). The data used is publicly available set posted on Kaggle, and it consists of 3,607 student records with 19 features and one target variable - exam score. The features include 13 categorical and 6 numerical variables, all of which required preprocessing, including encoding of categorical variables and standardization of numerical ones.

For deeper analysis, the dataset was manually divided into two groups: secondary and primary performance indicators. Models were then trained on the full dataset, combining all features. To evaluate the effectiveness of different approaches, models were compared across three prediction tasks: regression, multiclass classification, and binary classification.

The results show that secondary features alone predict poorly, but still correlated with performance. Directly related features performed much better. Combining all features improved results by 10–15% across tasks. In regression, only the full dataset reached 72% accuracy, highlighting the value of including qualitative metrics.

Introduction

With the growing emphasis on data-driven decision-making in education, the ability to accurately predict student performance has become increasingly important for academic institutions and policymakers alike. This project seeks to improve academic performance prediction by developing system capable of identifying patterns in student data. Such predictive model is essential not only for enabling early interventions and support planning but also for informing teaching strategies and curriculum design tailored to students' needs.

For instance, admission procedure is basically a prediction problem with a goal of selection ones who would excel at studies. In many universities in Kazakhstan, including Nazarbayev University, admission decisions are based solely on standardized MCQ tests. While this approach works, selection procedure does not account for qualitative or contextual information. A more nuanced, but data-driven method could complement existing practices by incorporating broader predictors without compromising objectivity.

It is evident that academic performance is influenced by variety of factors such as parental education level, occupation, income, household resources, and prior academic achievements (Rodríguez-Hernández et al., 2020), but traditionally, when Machine Learning is used for grade prediction, it is often relied on a narrow set of predictors - such as standardized test scores and past performance (Hsu & Schombert, 2010). This project, however, takes a more comprehensive approach by integrating a diverse range of features, including socio-economic background, study habits, physical activity level, and previous academic records. This enables a deeper and more nuanced understanding of the factors shaping student outcomes, ultimately leading to improved prediction accuracy.

To assess how a diversified data set affects accuracy, the project conducts a comparative analysis. Initially, models are built using only secondary variables, followed by models that uses features with higher correlation such as past test scores and attendance. These models are tested across three predictive tasks: regression, multiclass classification (where the target variable, test scores, is split into four quartiles), and binary classification (where the score is divided at the average). Results are then compared against a model trained on the full feature set.

A practical application of this project lies in the potential integration of such predictive systems within university settings. They could help identify students with special academic needs early on or serve as a foundation for reforming admission procedures to better recognize student potential.

The project's innovation lies in its dual contribution: the inclusion of varied data types and its structured comparison of narrow versus broad predictive models based on performance across different prediction tasks. By highlighting the limitations of conventional methods and showing how machine learning can accurately predict academic performance, this project contributes meaningfully to more equitable and informed educational decision-making.

Methodology

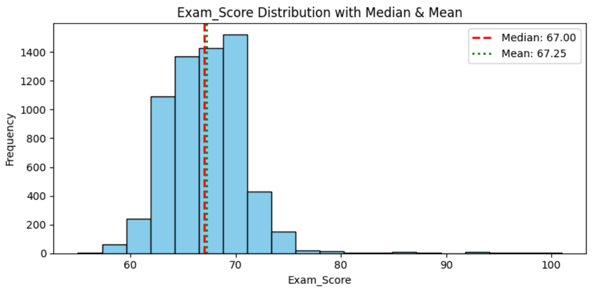
1. Introduction to Data

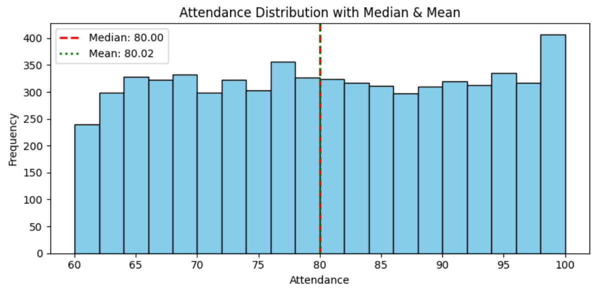
The data is publicly accessible on internet; it is posted on Kaggle in 2024. The dataset has 6607 records grouped into 20 features (Table 1), with the variable “Test Scores” chosen as the target. Data distribution is partially presented in Table 2. The dataset was split into 80% train and 20% test sets.

The dataset was checked for null and empty values. No empty strings were found, but Teacher_Quality, Parental_Education_Level, and Distance_from_Home had around 1% null values each. All other columns were complete. These missing values were addressed through imputation during preprocessing to ensure smooth model training.

Numerical	**Categorical**
- Hours Studied **- Attendance** - Sleep Hours **- Previous Scores** **- Tutoring Sessions** - Physical Activity **- Exam Score**	- Parental Involvement - Access to Resources - Extracurricular Activities - Motivation Level - Internet Access - Family Income **- Teacher Quality** - School Type - Peer Influence - Learning Disabilities - Parental Education Level - Distance from Home - Gender

Table 1. All features





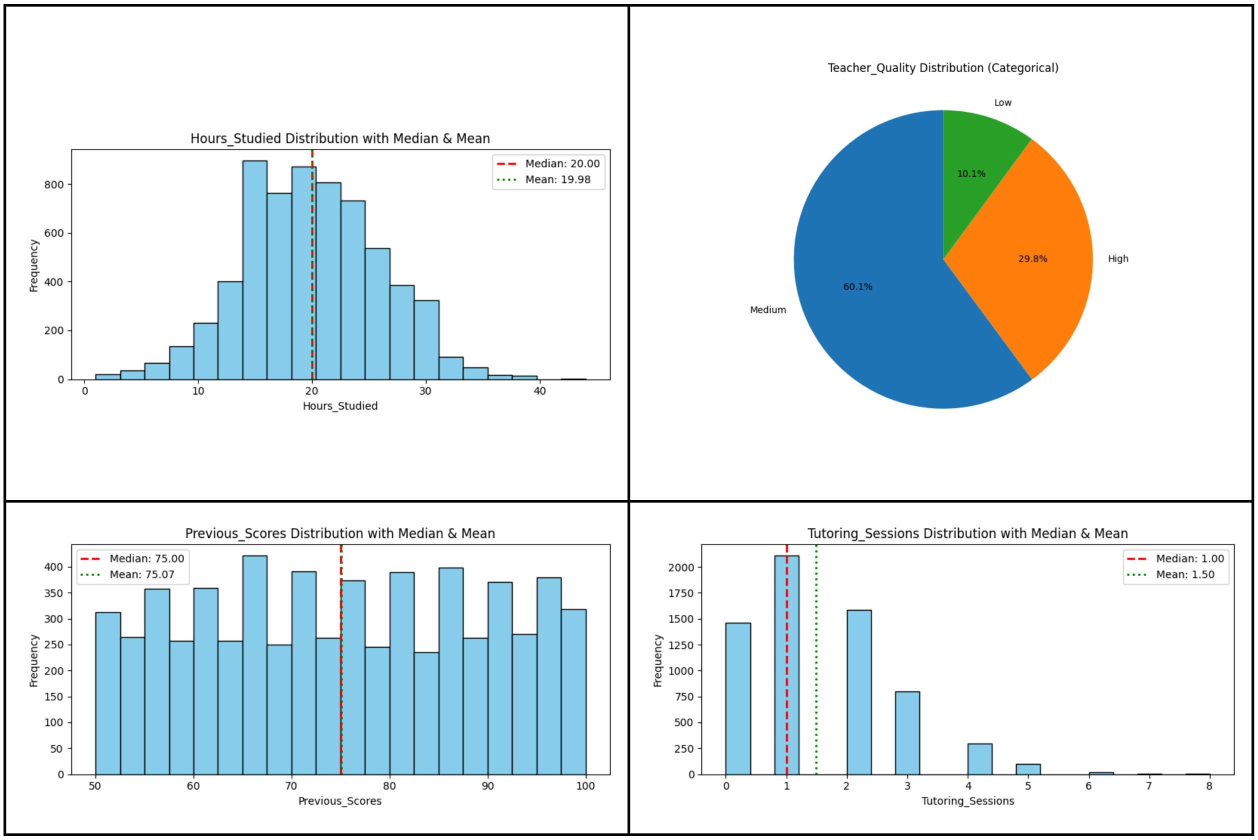


Table 2. Some feature distribution

2. Dataset Preprocessing and Pipeline

Since the dataset includes both categorical and numerical features, and contains missing values, preprocessing was essential before model training. For categorical features, missing values were imputed using the most frequent value and then encoded with an OrdinalEncoder, assigning unseen categories a specific code (-1). For numerical features, missing values were imputed with the mean and standardized using StandardScaler to ensure all features are on the same scale. A ColumnTransformer was used to apply these transformations separately to categorical and numerical columns within a unified framework.

However, applying preprocessing to the entire dataset before splitting would lead to data leakage. To avoid this, Scikit-learn's Pipeline was used, which combines preprocessing and model training into a single, safe procedure. The pipeline takes in the training and test sets, transforms the training data and fits the test, and trains the model properly, preserving the integrity of the test set.

The pipeline was also crucial for preventing another form of data leakage during cross-validation. Since cross-validation involves creating multiple train-validation splits, preprocessing must be applied within each fold, not beforehand. The pipeline ensures that each fold is properly preprocessed and trained independently.

The overall training procedure was as follows:

1. A model (e.g., Linear Regression, Logistic Regression, Random Forest, or MLP) was selected, along with a set of potential hyperparameters.
2. RandomizedSearchCV was used to tune these hyperparameters.
3. The best configuration was then used to train the final model.
4. Performance was evaluated on the test set and validated via cross-validation.

3. Predicting student performance

With the pipeline in place, a comparative analysis was conducted. First, models were trained using only socio-economic features, such as parental education, motivation, and peer influence, totaling 14 variables. Then, models were trained using only academic-performance-related features, such as test scores and attendance, totaling 5 features (bolded ones in Table 1).

Following that, these models were evaluated across three predictive tasks:

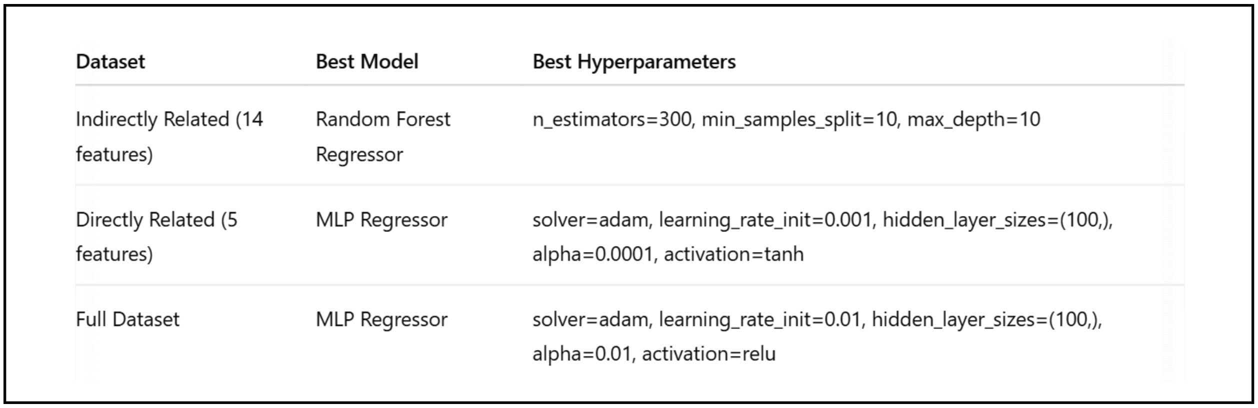
Regression Task: Linear Regression, Random Forest Regressor, and MLP Regressor were used to predict students' test scores. Model performance was evaluated using the R^2 metric.

Multiclass Classification: The continuous test score was split into four quartiles, creating a four-class classification problem. Logistic Regression, Random Forest, and MLP classifiers were applied, and performance was assessed using F1 score, accuracy, weighted recall, and weighted precision.

Binary Classification: Test scores were split based on the mean into two classes (above/below average). The same classifiers and evaluation metrics were used as in the multiclass setup

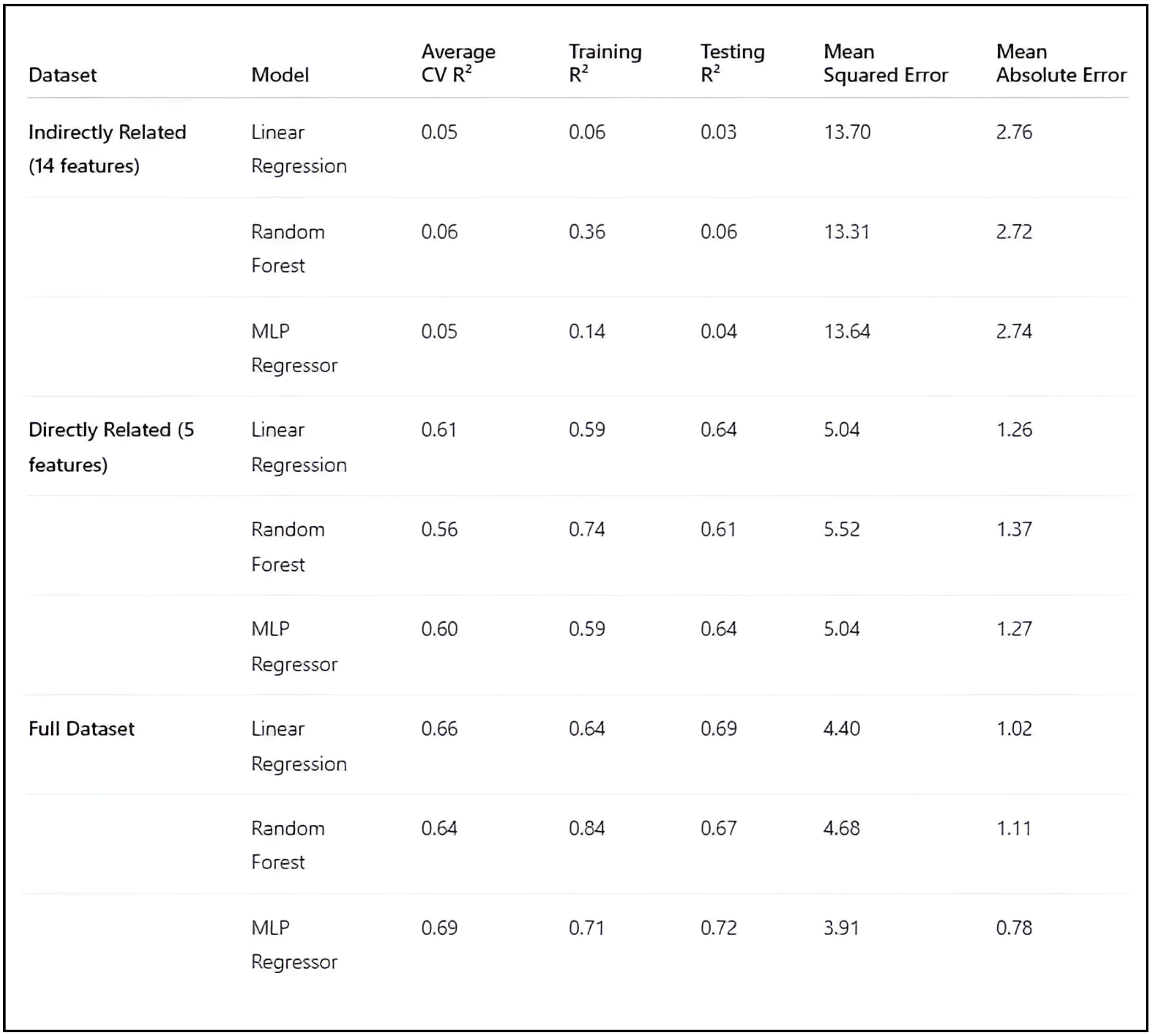
Finally, models were trained on the full feature set (combining both socio-economic and academic features), and their performance was compared against models trained on selective subsets. This allowed for an in-depth understanding of how different groups of features contribute to predictive accuracy.

Results



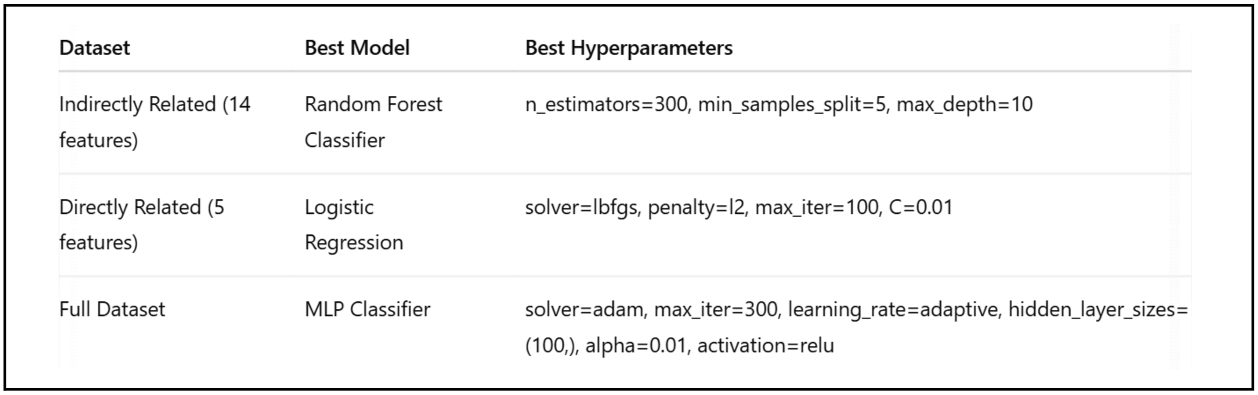
Dataset	Best Model	Best Hyperparameters
Indirectly Related (14 features)	Random Forest Regressor	n_estimators=300, min_samples_split=10, max_depth=10
Directly Related (5 features)	MLP Regressor	solver=adam, learning_rate_init=0.001, hidden_layer_sizes=(100,), alpha=0.0001, activation=tanh
Full Dataset	MLP Regressor	solver=adam, learning_rate_init=0.01, hidden_layer_sizes=(100,), alpha=0.01, activation=relu

Table 3. Best Performer models and Tuned Hyperparameters during Regression



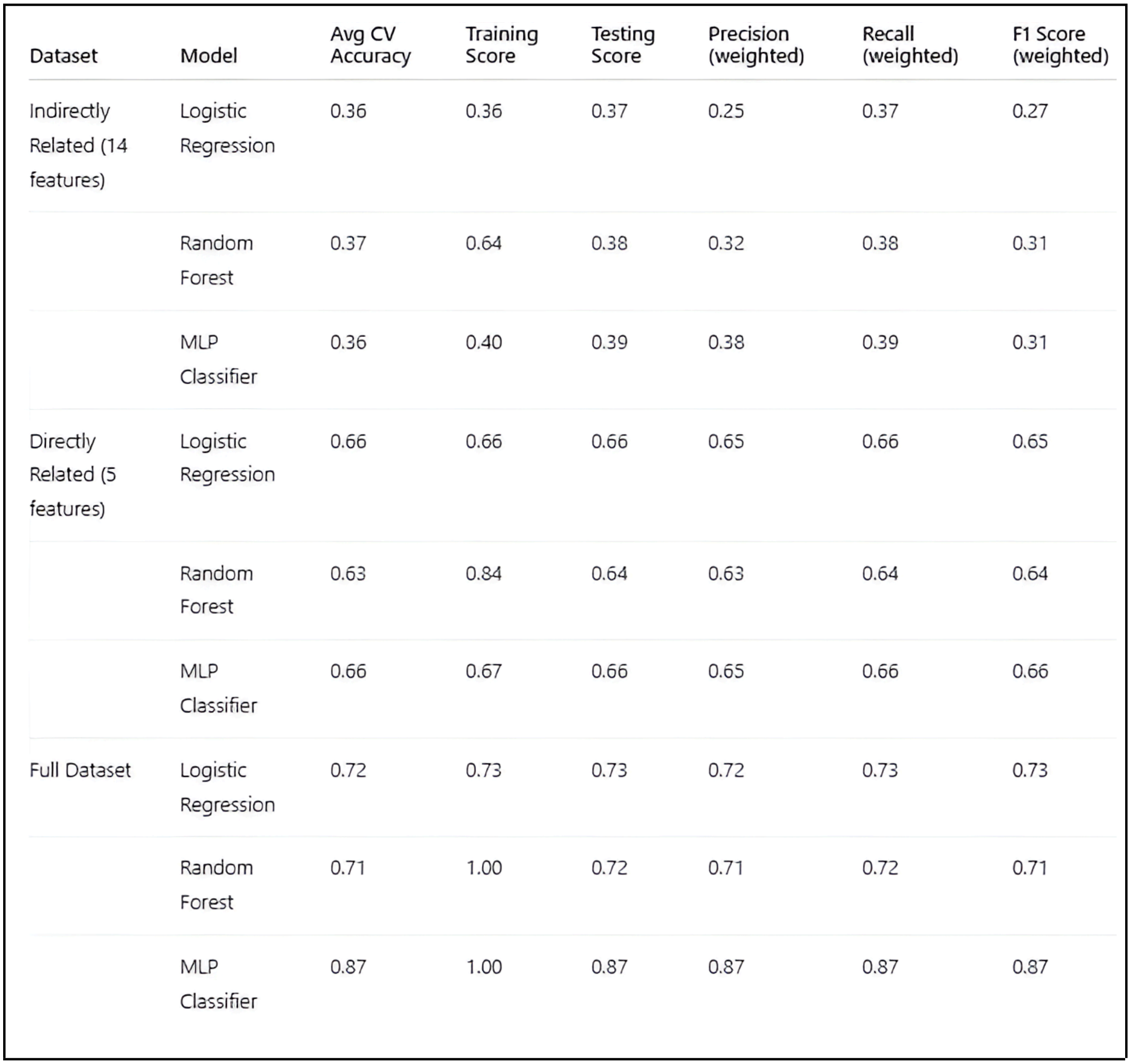
Dataset	Model	Average CV R^2	Training R^2	Testing R^2	Mean Squared Error	Mean Absolute Error
Indirectly Related (14 features)	Linear Regression	0.05	0.06	0.03	13.70	2.76
	Random Forest	0.06	0.36	0.06	13.31	2.72
	MLP Regressor	0.05	0.14	0.04	13.64	2.74
Directly Related (5 features)	Linear Regression	0.61	0.59	0.64	5.04	1.26
	Random Forest	0.56	0.74	0.61	5.52	1.37
	MLP Regressor	0.60	0.59	0.64	5.04	1.27
Full Dataset	Linear Regression	0.66	0.64	0.69	4.40	1.02
	Random Forest	0.64	0.84	0.67	4.68	1.11
	MLP Regressor	0.69	0.71	0.72	3.91	0.78

Table 4. Regression Task Performance



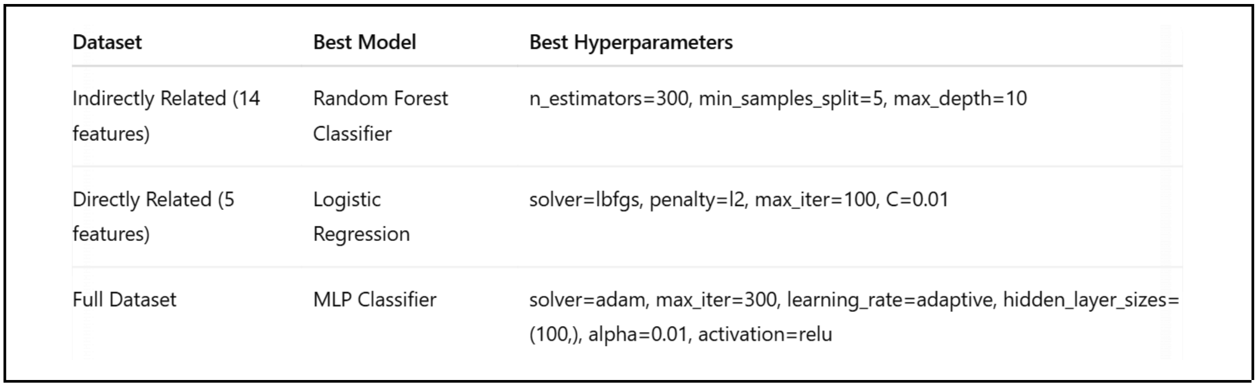
Dataset	Best Model	Best Hyperparameters
Indirectly Related (14 features)	Random Forest Classifier	n_estimators=300, min_samples_split=5, max_depth=10
Directly Related (5 features)	Logistic Regression	solver=lbfgs, penalty=l2, max_iter=100, C=0.01
Full Dataset	MLP Classifier	solver=adam, max_iter=300, learning_rate=adaptive, hidden_layer_sizes=(100,), alpha=0.01, activation=relu

Table 5. Best Performer models and Tuned Hyperparameters during Multiclass Classification



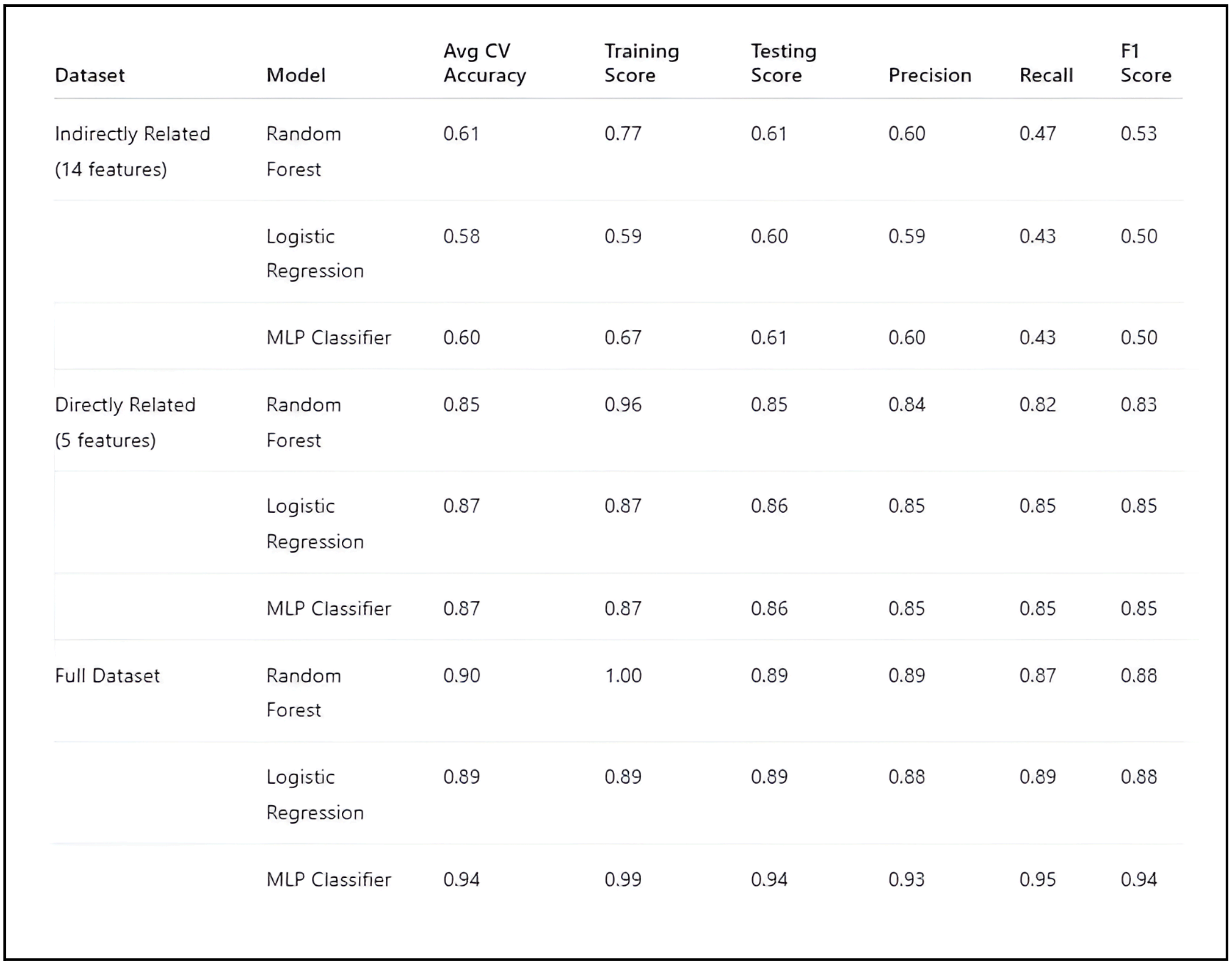
Dataset	Model	Avg CV Accuracy	Training Score	Testing Score	Precision (weighted)	Recall (weighted)	F1 Score (weighted)
Indirectly Related (14 features)	Logistic Regression	0.36	0.36	0.37	0.25	0.37	0.27
	Random Forest	0.37	0.64	0.38	0.32	0.38	0.31
	MLP Classifier	0.36	0.40	0.39	0.38	0.39	0.31
Directly Related (5 features)	Logistic Regression	0.66	0.66	0.66	0.65	0.66	0.65
	Random Forest	0.63	0.84	0.64	0.63	0.64	0.64
	MLP Classifier	0.66	0.67	0.66	0.65	0.66	0.66
Full Dataset	Logistic Regression	0.72	0.73	0.73	0.72	0.73	0.73
	Random Forest	0.71	1.00	0.72	0.71	0.72	0.71
	MLP Classifier	0.87	1.00	0.87	0.87	0.87	0.87

Table 6. Multiclass Classification Task Performance



Dataset	Best Model	Best Hyperparameters
Indirectly Related (14 features)	Random Forest Classifier	n_estimators=300, min_samples_split=5, max_depth=10
Directly Related (5 features)	Logistic Regression	solver=lbfgs, penalty=l2, max_iter=100, C=0.01
Full Dataset	MLP Classifier	solver=adam, max_iter=300, learning_rate=adaptive, hidden_layer_sizes=(100,), alpha=0.01, activation=relu

Table 7. Best Performer models and Tuned Hyperparameters during Binary Classification



Dataset	Model	Avg CV Accuracy	Training Score	Testing Score	Precision	Recall	F1 Score
Indirectly Related (14 features)	Random Forest	0.61	0.77	0.61	0.60	0.47	0.53
	Logistic Regression	0.58	0.59	0.60	0.59	0.43	0.50
	MLP Classifier	0.60	0.67	0.61	0.60	0.43	0.50
Directly Related (5 features)	Random Forest	0.85	0.96	0.85	0.84	0.82	0.83
	Logistic Regression	0.87	0.87	0.86	0.85	0.85	0.85
	MLP Classifier	0.87	0.87	0.86	0.85	0.85	0.85
Full Dataset	Random Forest	0.90	1.00	0.89	0.89	0.87	0.88
	Logistic Regression	0.89	0.89	0.89	0.88	0.89	0.88
	MLP Classifier	0.94	0.99	0.94	0.93	0.95	0.94

Table 8. Binary Classification Task Performance

‘

Discussion and Conclusion

The results across regression, multiclass classification, and binary classification tasks highlight the critical importance of feature complexity in academic performance prediction.

In the regression task, models trained on indirectly related features performed poorly, with very low R^2 scores (~0.03–0.06) and high errors. Using directly related features significantly improved performance (~0.61–0.64), and the full dataset further boosted results, with the MLP Regressor achieving the best outcomes (R^2 = 0.72, MSE = 3.91).

In the multiclass classification task, poor results were observed when using indirectly related features (F1 < 0.32). Switching to directly related features raised the testing scores to ~0.64–0.66, and the full dataset led to even stronger performance, with the MLP Classifier achieving a super high testing score of 0.87 when other models had 0.72-0.73.

For the binary classification task, the pattern was consistent. Indirect features yielded low testing scores (~0.60), while directly related features boosted performance (~0.85–0.86). And when datasets were combined, the MLP Classifier again achieved the highest testing score (0.94) and F1 score (0.94).

Generally, the MLP model’s performance is not distinguishable from Tree-based and Linear models when the feature set is narrowed and more uniform in nature (i.e., either fully categorical or fully numerical). However, on the full dataset, across all three problems, the MLP model demonstrated outstanding performance. This can be attributed to the MLP's more sophisticated learning method, allowing it to capture complex patterns in diverse feature sets effectively

Overall, this project demonstrates that while using only secondary features results in poor predictions of academic performance, binary classification results show that some correlation still exists. As expected, directly related features performed much better, achieving very high performance in binary classification tasks. Interestingly, including secondary student data proved valuable: combining all features boosted performance by 10–15% across all tasks. In the most challenging task - regression - both narrowed datasets performed poorly on their own, but the full dataset achieved 72% accuracy. This highlights that when predicting student performance, considering qualitative and indirect factors can be both possible and beneficial.

However, there are important limitations. This project should be viewed more as an illustrative case study rather than serious research on academic performance, mainly because the dataset is limited and somewhat ambiguous: for example, it is unclear what exactly the exam score represents. If GPA or subject-specific scores were used as targets, the importance of features would vary significantly, which makes sense, as excelling in math versus linguistics would naturally depend on very different factors.

‘

Reference list

Hsu, S. D., & Schombert, J. (2010). Data mining the university: College GPA predictions from SAT scores. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1589792

Rodríguez-Hernández, C. F., Cascallar, E., & Kyndt, E. (2020). Socio-economic status and academic performance in higher education: A systematic review. *Educational Research Review, 29*, 100305. https://doi.org/10.1016/j.edurev.2019.100305

Student performance factors. Kaggle. (2024, November 26). https://www.kaggle.com/datasets/lainguyn123/student-performance-factors