University of Verona

Master’s of Artificial Intelligence 2024/2025

Machine Learning & Deep Learning

Machine Learning project: Evaluating Regression Models for Student Grade Prediction

Name: Omar Khalil

Matriculation number: VR528cS5

Supervisor: Prof. Cigdem beyan

Date: 17/02/2025

Contents

1. Introduction

Regression is one of the main applications of supervised learning in machine learning. Alongside Classification they make up the most widely known tasks in machine learning. Regression is used to model a relationship between input variables and output. This relationship is a function that maps continuous features to target variable.
The project discusses the analysis and implementation of various regression models to predict values in a continuous label. The dataset is about academic performance and the model uses study hours, sleep hours and other factors as features [1]. Multiple regression models are tested to analyse their performance, analyse the factors that decide the performance of each model and learn the best regression model possible.

1.1 Problem and motivation

The forecasting of student grades can be used to adjust the education process form the perspective of the student or the educational institutions. Using the predictive model the students can evaluate their studying plan and everyday routine based on the predicted values. It could also be used in educational research to better the educational process.
To learn the best model possible the regression models are to be tested and based on the available dataset the parameters are to be chosen so that the model produces the best results. Th testing and comparison of various models is also helpful to identify the factors that decide the performance of each model to further understand regression techniques and define the best possible model with respect to the available data.

1.2 State of the art

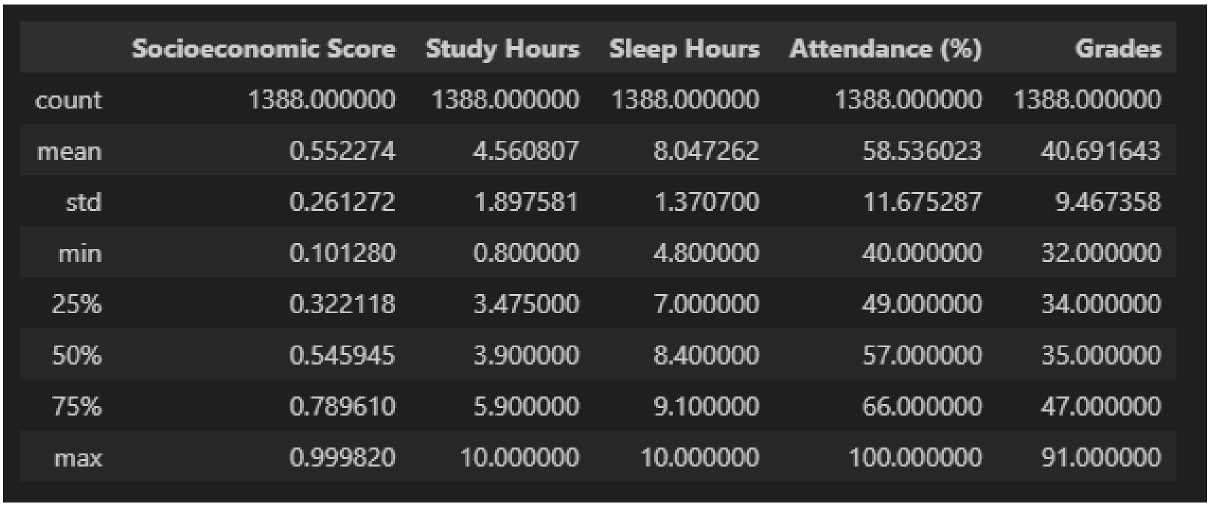
Regression models vary according to the used technique, but they all share the same concept of regression. The techniques include from simple linear regression using statistical models to model the relationship between input and output as well as decision and Random Forest.
linear Regression assumes linearity and captures linear relationships between the data. It is also sensitive to collinearity.
Ridge regression and lasso regression are extensions of linear regression that introduce regularization to the cost function. Regularization deal with the correlation in the dataset. While regularization allows to control the magnitudes of coefficient and emphasize more important features, lasso regression can effectively eliminate some features effectively performing feature selection.
Elastic nets are the combination of lasso and ridge regression.
Polynomial regression models the relationship between features and target as an nth-degree polynomial. This technique is used when the relationship is nonlinear.
Random forest use ensemble methods and bagging to train multiple decision trees each trained on random split of data.
Among others are techniques like logistic regression and regression using support vector machines.

1.3 Objective

The project’s objective is to analyse different regression models for predicting student grades. And evaluate how different regression models as well as model training and preprocessing steps perform. In addition, the project aims to analyse the impact of the dataset characteristics and processing and modelling techniques. This helps to decide the best modelling approach and to understand the reasons behind it.

2 Dataset

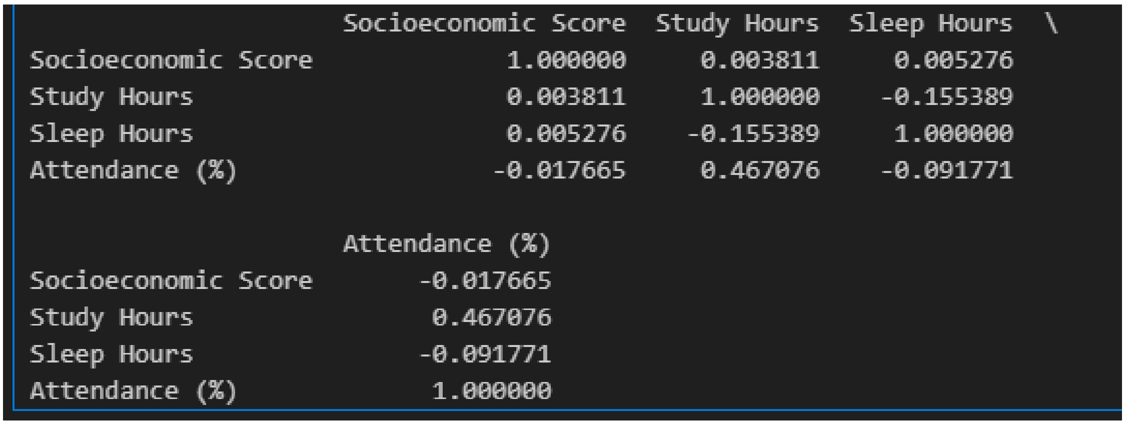
The chosen dataset is a synthetic dataset representing student performance. It is designed to mimic real world scenarios. The dataset includes factors such as study habits, sleep patterns, socioeconomic background, and class attendance. The target is the grades of the students with each row representing a student. The dataset has 1388 samples. The study hours col represents the average daily hours spent studying. The sleep hours col is the average daily hours of sleep. The socioeconomic score is a normalized score representing the socioeconomic score of the student. The attendance col is the percentage of classes attended by students. The grades col is the target of the dataset.
The characteristics of the dataset are displayed in figure 1.



	Socioeconomic Score	Study Hours	Sleep Hours	Attendance (%)	Grades
count	1388.000000	1388.000000	1388.000000	1388.000000	1388.000000
mean	0.552274	4.560807	8.047262	58.536023	40.691643
std	0.261272	1.897581	1.370700	11.675287	9.467358
min	0.101280	0.800000	4.800000	40.000000	32.000000
25%	0.322118	3.475000	7.000000	49.000000	34.000000
50%	0.545945	3.900000	8.400000	57.000000	35.000000
75%	0.789610	5.900000	9.100000	66.000000	47.000000
max	0.999820	10.000000	10.000000	100.000000	91.000000

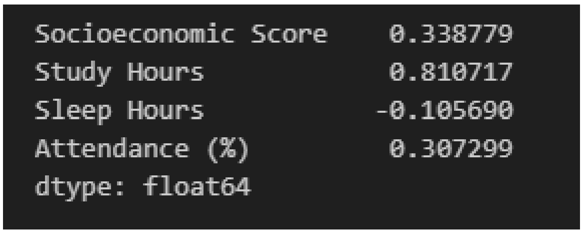
Figure 1 Description of the dataset

The correlation of features is another aspect relevant to regression. Here we will monitor the correlation to help analyse the models and derive useful conclusions. The correlation is displayed in figure 2 where the correlation between each feature is calculated. And in figure 3 the correlation between each feature and the target is displayed.
Figure 2 shows little to no correlation between features but figure 3 shows high correlation between the study hours and the target. In section 5 the impact of correlation will be discussed in detail.



```
                        Socioeconomic Score  Study Hours  Sleep Hours  \
Socioeconomic Score                1.000000     0.003811     0.005276
Study Hours                        0.003811     1.000000    -0.155389
Sleep Hours                        0.005276    -0.155389     1.000000
Attendance (%)                    -0.017665     0.467076    -0.091771

                        Attendance (%)
Socioeconomic Score          -0.017665
Study Hours                   0.467076
Sleep Hours                  -0.091771
Attendance (%)                1.000000
```

Figure 2 the correlation between features

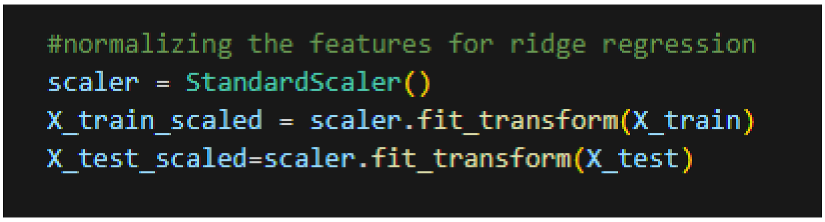


```
Socioeconomic Score     0.338779
Study Hours             0.810717
Sleep Hours            -0.105690
Attendance (%)          0.307299
dtype: float64
```

Figure 3 the correlation between features and target

3 Data processing

The data processing depends on the characteristic of the data as well as the models to be used. In this instance the dataset shows no need for preprocessing except for standardization needed for ridge regression as it is sensitive to scale. The cross-validation method is also used to determine how well the model's generalization is by training different models on different instances of the dataset consider that the dataset is relatively small.



```
#normalizing the features for ridge regression
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled=scaler.fit_transform(X_test)
```

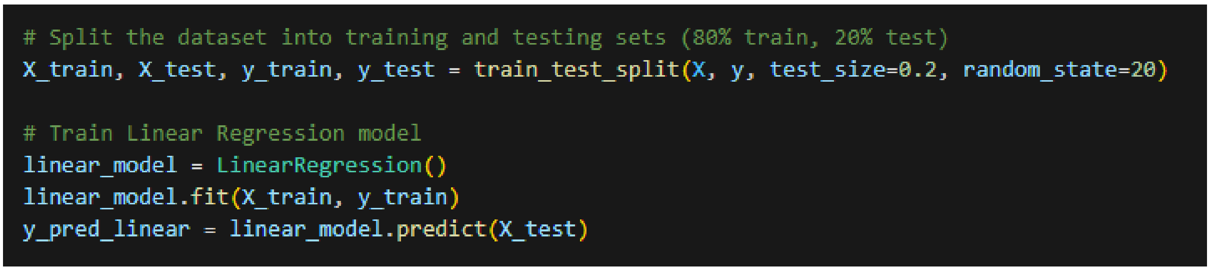
Figure 4 normalization of the features for ridge regression

4 Methodology

The dataset is split into 80% training and 20% testing. The split is only between training and validation sets to compensate for the relatively small dataset. The models will be evaluated based on regular split as well as cross validation using evaluation metrics like mean absolute error and mean square error. Furthermore, normalization will be used when its deemed necessary and its results will also be compared.

4.1 Linear regression model

The linear regression model is trained and evaluated using the linear regression model from sklearn. In linear regression correlation could have an impact on the results. According to the correlation in figure 2 there is no correlation between features therefore the model doesn't need any data processing.

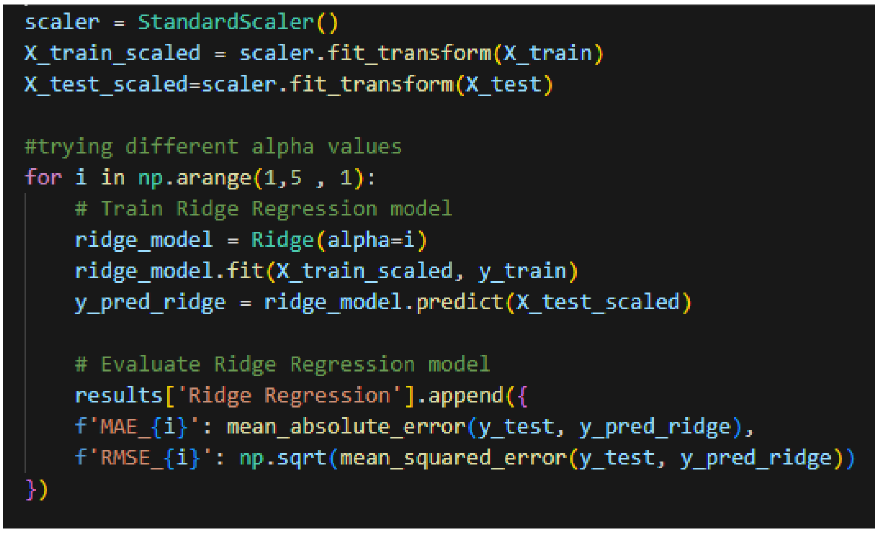


```
# Split the dataset into training and testing sets (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=20)

# Train Linear Regression model
linear_model = LinearRegression()
linear_model.fit(X_train, y_train)
y_pred_linear = linear_model.predict(X_test)
```

Figure 5 Linear regression model

4.2 Ridge regression model

The ridge regression with regularization is tested with and without standardization of features. As it is sensitive to scale. Using different alpha values, the ridge model will be evaluated with alpha values iterating from 1 through 5 using the same split used in linear regression. Each iteration is calculated using mean absolute error and mean squared error as evaluation metrics.

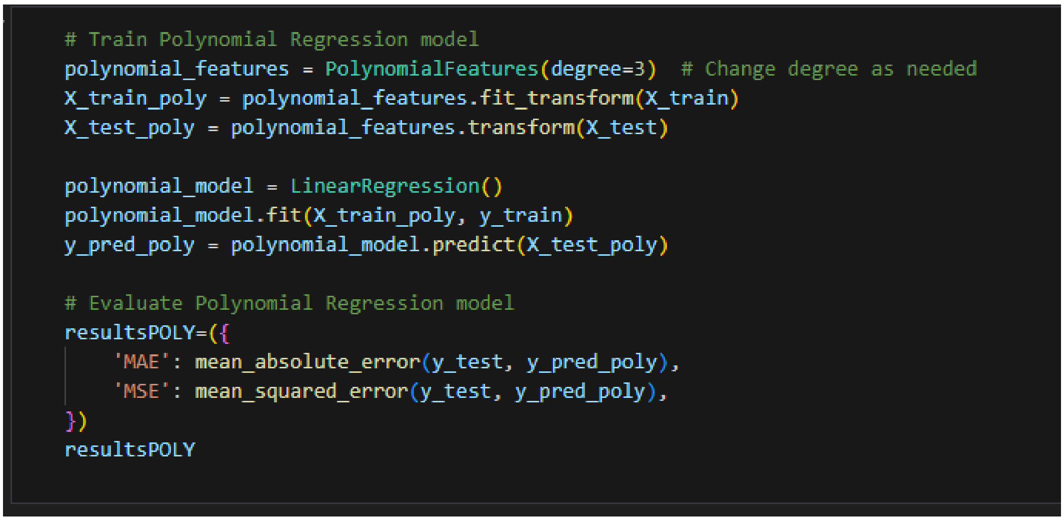


```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled=scaler.fit_transform(X_test)

#trying different alpha values
for i in np.arange(1,5 , 1):
    # Train Ridge Regression model
    ridge_model = Ridge(alpha=i)
    ridge_model.fit(X_train_scaled, y_train)
    y_pred_ridge = ridge_model.predict(X_test_scaled)

    # Evaluate Ridge Regression model
    results['Ridge Regression'].append({
    f'MAE_{i}': mean_absolute_error(y_test, y_pred_ridge),
    f'RMSE_{i}': np.sqrt(mean_squared_error(y_test, y_pred_ridge))
})
```

Figure 6 Ridge regression model

4.3 Polynomial regression model

For the polynomial regression model the choice of the parameter is crucial to the model's performance. The optimal degree of the polynomial is decided by trying multiple values. The value that was found to be optimal is the third-degree polynomial. After the third-degree further values improve the model slightly but that improvement is not worth it as further values increase the complexity of the model and increases the chance of overfitting.



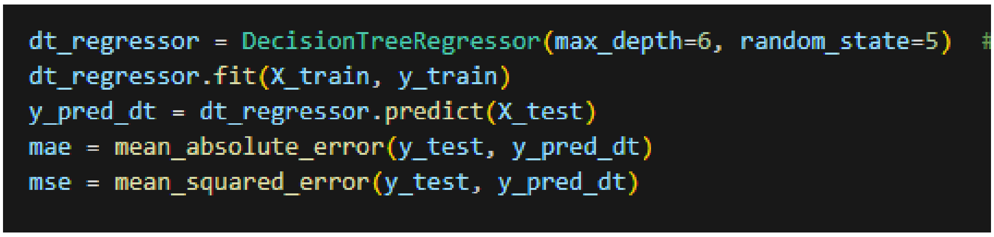
```python
# Train Polynomial Regression model
polynomial_features = PolynomialFeatures(degree=3)  # Change degree as needed
X_train_poly = polynomial_features.fit_transform(X_train)
X_test_poly = polynomial_features.transform(X_test)

polynomial_model = LinearRegression()
polynomial_model.fit(X_train_poly, y_train)
y_pred_poly = polynomial_model.predict(X_test_poly)

# Evaluate Polynomial Regression model
resultsPOLY=({
    'MAE': mean_absolute_error(y_test, y_pred_poly),
    'MSE': mean_squared_error(y_test, y_pred_poly),
})
resultsPOLY
```

Figure 7 Polynomial regression model

To test the generalization of the model the cross-validation technique is implemented on the polynomial regression model.

4.4 Decision trees and Random Forest

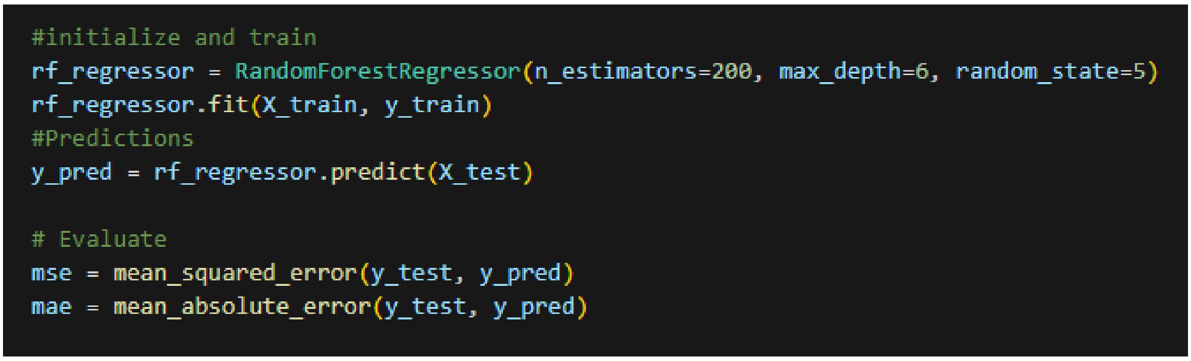
The model used is the decisionTreeRegressor from sklearn. The decision tree model was trained on split data without any processing. The depth parameter was chosen by increasing the depth until the model performed worse. The results are evaluated using the mae and mse scores. Also, here the cross-validation technique will be implemented.



```python
dt_regressor = DecisionTreeRegressor(max_depth=6, random_state=5)  #
dt_regressor.fit(X_train, y_train)
y_pred_dt = dt_regressor.predict(X_test)
mae = mean_absolute_error(y_test, y_pred_dt)
mse = mean_squared_error(y_test, y_pred_dt)
```

Figure 8 Decision Tree model

Considering the small size of the dataset it is a good idea to test the Random Forest technique. The Random Forest is tested with the same depth as the decision tree and the n-estimators are increased in steps of 50 from 100 until we reach the optimal value of 200. Cross validation will be applied to the Random Forest model for analysis.



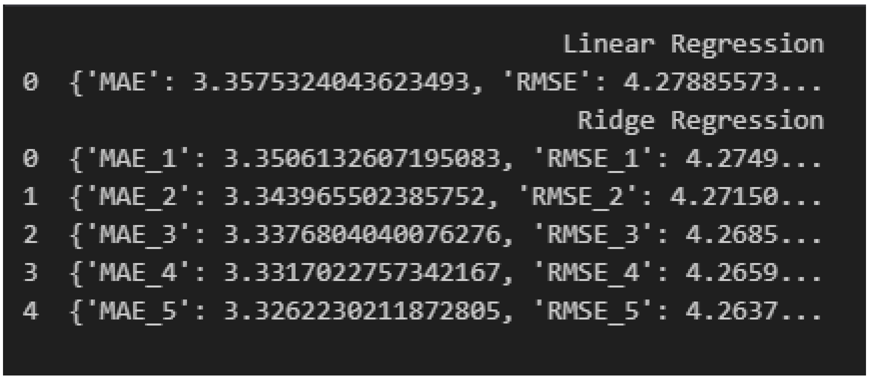
```
#initialize and train
rf_regressor = RandomForestRegressor(n_estimators=200, max_depth=6, random_state=5)
rf_regressor.fit(X_train, y_train)
#Predictions
y_pred = rf_regressor.predict(X_test)

# Evaluate
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
```

Figure 9 Random Forest model

5 Results

5.1 Linear regression and Ridge regression

In the case of non-normalized values, the results of linear and ridge regression are very similar. The MAE values across various alpha values are similar with a slight difference as shown in Figure 10. Furthermore, the choice of the alpha parameters has little effect on the results.

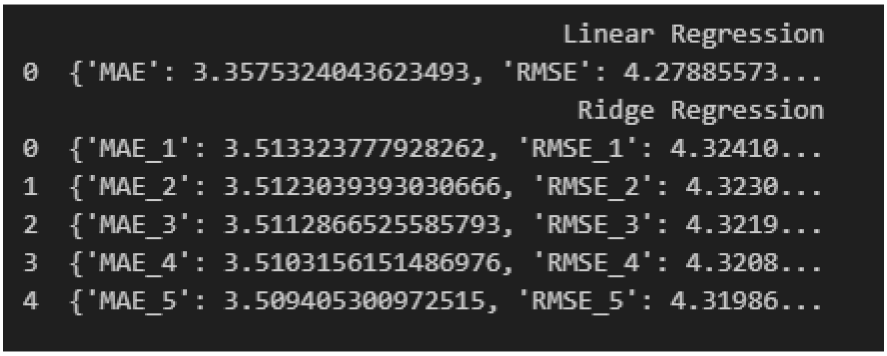


```
                                    Linear Regression
0  {'MAE': 3.3575324043623493, 'RMSE': 4.27885573...
                                     Ridge Regression
0  {'MAE_1': 3.3506132607195083, 'RMSE_1': 4.2749...
1  {'MAE_2': 3.343965502385752, 'RMSE_2': 4.27150...
2  {'MAE_3': 3.3376804040076276, 'RMSE_3': 4.2685...
3  {'MAE_4': 3.3317022757342167, 'RMSE_4': 4.2659...
4  {'MAE_5': 3.3262230211872805, 'RMSE_5': 4.2637...
```

Figure 10 the evaluation of linear and ridge regression models with no normalization

Given the fact that there is no correlation between the features the similarity between the two models is expected.

In the case of normalized features, the results vary slightly. The ridge model performs slightly worse than the regression model as seen in Figure 11.

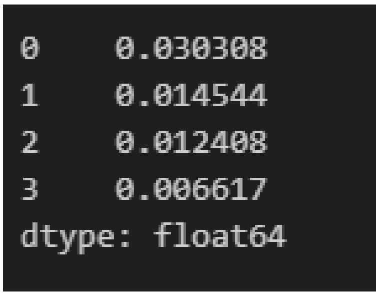


```
                                   Linear Regression
0  {'MAE': 3.3575324043623493, 'RMSE': 4.27885573...
                                    Ridge Regression
0  {'MAE_1': 3.513323777928262, 'RMSE_1': 4.32410...
1  {'MAE_2': 3.5123039393030666, 'RMSE_2': 4.3230...
2  {'MAE_3': 3.5112866525585793, 'RMSE_3': 4.3219...
3  {'MAE_4': 3.5103156151486976, 'RMSE_4': 4.3208...
4  {'MAE_5': 3.509405300972515, 'RMSE_5': 4.31986...
```

Figure 11 the evaluation of linear and ridge regression models with normalization

This result is surprising as normalization is meant to ensure the variables are on the same scale and avoid features with larger scales that can dominate the regularization penalty. So, the values should have been better if there was correlation between values or they should have been the same.

But the values are slightly worse this means that normalization contributed to the model being slightly worse. In Figure 12 is the correlation between the normalized training data and the target displayed.



```
0    0.030308
1    0.014544
2    0.012408
3    0.006617
dtype: float64
```

Figure 12 Correlation between X_train_scaled and y

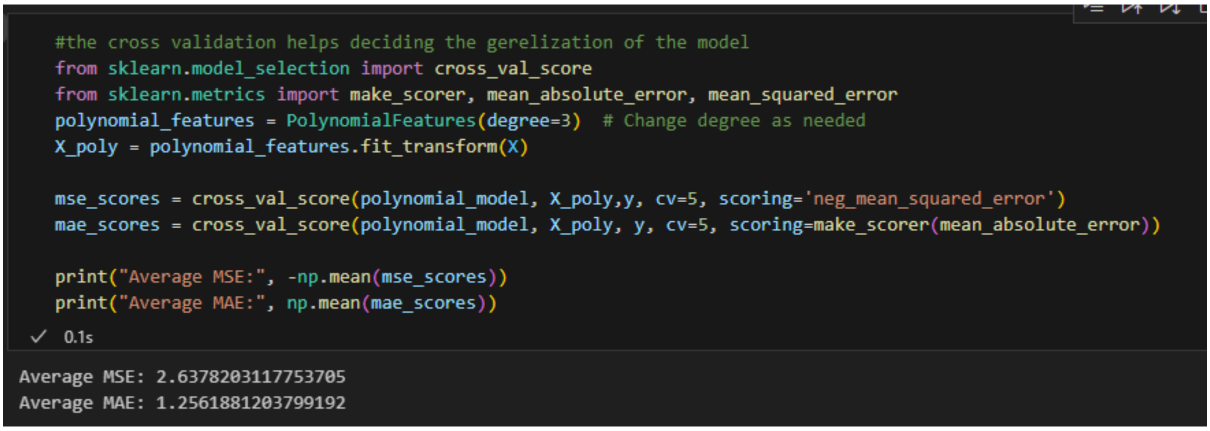
The difference between figure 3 that represents the correlation before normalization and figure 12 is the study hours. That correlation between study hours and grade is 0.8 before normalization which is a high correlation. after normalization that correlation is no longer available.

This could be a possible explanation for the difference in performance after normalization in ridge regression. Also testing the ridge regression model wasn't necessary as there was no correlation between the features.

5.2 Polynomial regression

The mean absolute error of the third-degree polynomial regression is 1,26 which makes it the best performing model so far. this means that the data has non-linear relationships that require a polynomial model to capture.

To ensure the generalization of the model we will use cross validation to test the model on different instances of the data. 5 models will be generated each evaluated based on a different section of the data.



```
#the cross validation helps deciding the gerelization of the model
from sklearn.model_selection import cross_val_score
from sklearn.metrics import make_scorer, mean_absolute_error, mean_squared_error
polynomial_features = PolynomialFeatures(degree=3)  # Change degree as needed
X_poly = polynomial_features.fit_transform(X)

mse_scores = cross_val_score(polynomial_model, X_poly,y, cv=5, scoring='neg_mean_squared_error')
mae_scores = cross_val_score(polynomial_model, X_poly, y, cv=5, scoring=make_scorer(mean_absolute_error))

print("Average MSE:", -np.mean(mse_scores))
print("Average MAE:", np.mean(mae_scores))
```

```
Average MSE: 2.6378203117753705
Average MAE: 1.2561881203799192
```

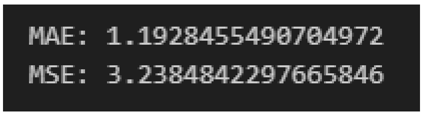
Figure 13 Cross validation

The average MAE of the Cross-validation process shows similar results to the initial MAE which means the model can generalize well on unseen data.

5.3 Decision trees and random forest

5.3.1 Decision trees

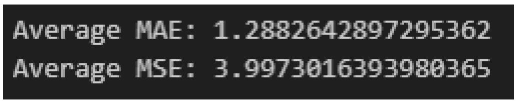
The results of the decision tree are like the polynomial regression model. The model was able to capture the nonlinear relation correctly.



```
MAE: 1.1928455490704972
MSE: 3.2384842297665846
```

Figure 14 Evaluation of Decision tree model

To test the generalization of the model and make sure its nor overfitted cross validation used. in figure 15 the cross-validation results of the decision tree model are displayed.



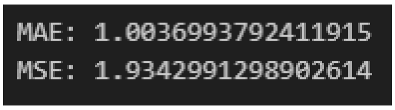
```
Average MAE: 1.2882642897295362
Average MSE: 3.9973016393980365
```

Figure 15 Cross validation of Decision tree model

The results show that the model performs slightly worse on average which indicate overfitting. This suggests that techniques like decision trees are more prone to overfitting than polynomial regression.

5.3.2 Random Forest

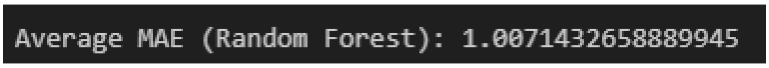
To further enhance the decision tree model, we will use random forest which is an ensemble method using decision trees. The Random Forest model results are displayed in Figure 16.



```
MAE: 1.0036993792411915
MSE: 1.9342991298902614
```

Figure 16 Random Forest model evaluation

The model shows the best result out of all models with mean absolute error of 1. This result is logical following the usage of random forest as it is an average of multiple trees that are trained on different instances of the dataset which automatically prevents overfitting. this is confirmed by applying cross validation to the random forest and getting similar results.



```
Average MAE (Random Forest): 1.0071432658889945
```

Figure 17 Cross validation with Random Forest model

5.4 Table of results

The mean absolute error of the models is displayed in the following table.

Table 1 Table of mae values

Model	Normal Split	Cross validation
Linear Regression	3,36	-
Ridge Regression with Normalization	3,51	-
Ridge Regression without Normalization	3,34	-
Polynomial Regression	1,32	1,26
Decision Tree	1,19	1.29
Random Forest	1,00	1,01

6 Conclusion

The goal of the project was to evaluate multiple regression models, analyse them, and determine the best model for accurately predicting student grades. Several regression models were used, including Linear Regression, Ridge Regression, Polynomial Regression, Decision Trees, and Random Forest. The models were evaluated using metrics like MAE and MSE.

Random Forest performed better than all models, providing the most accurate forecasting of grades. Polynomial Regression also showed improvement over the Linear Regression models. Ridge Regression was of less relevance due to the lack of correlation between the features, but it demonstrated how normalization could impact the generalization ability of the model depending on the technique used. Using cross-validation, we concluded that for better generalization on small datasets with non-linear relationships, Random Forest performed best.

The dataset used is synthetic and simple, which may limit the scope of analysis and results.

Future work could involve using larger, more diverse datasets to better analyse the models. Additionally, exploring other models, such as SVMs and Neural Networks, could potentially offer improved performance and accuracy.

7 List of figures

8 List of tables

References

[1] Umair Zia. 2024. *Predict Student Performance*. *DOI*=10.34740/kaggle/dsv/10303802.