IE0005 PROJECT

Contributed by:
Chindepalli Sanhith Krishna
Lakhotia Shreyas
Singh Siddhant
Subramanian Suraj

Agenda

1. Objective
2. Data Analysis
3. Data Cleaning and Feature Engineering
4. Machine Learning Models
5. Conclusion

PROBLEM STATEMENT

Can socioeconomic factors such as parental education level, lunch status, and test preparation courses predict a student's academic performance category?

OBJECTIVE: FIND WHICH ATTRIBUTE(S) PLAYS A MAJOR ROLE IN INFLUENCING STUDENT'S PERFORMANCE

INFLUENCING ATTRIBUTES:

- Gender
- Race & ethnicity
- Parental level of education
- Test preparation course

MEASUREMENT ATTRIBUTES:

- Math score
- Reading score
- Writing score

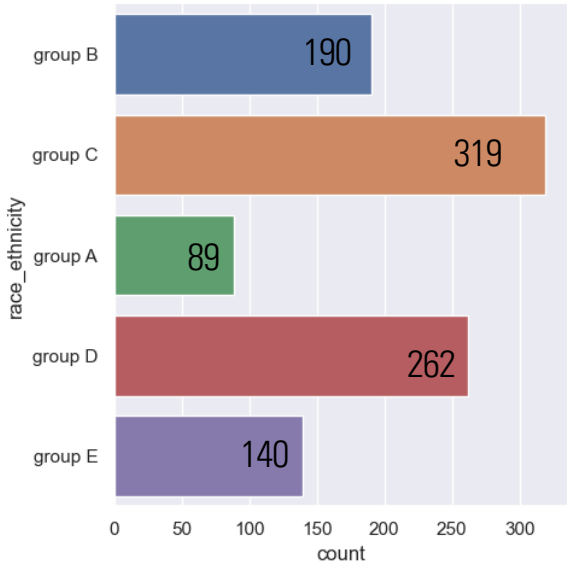
DATA PRE-PROCESSING:

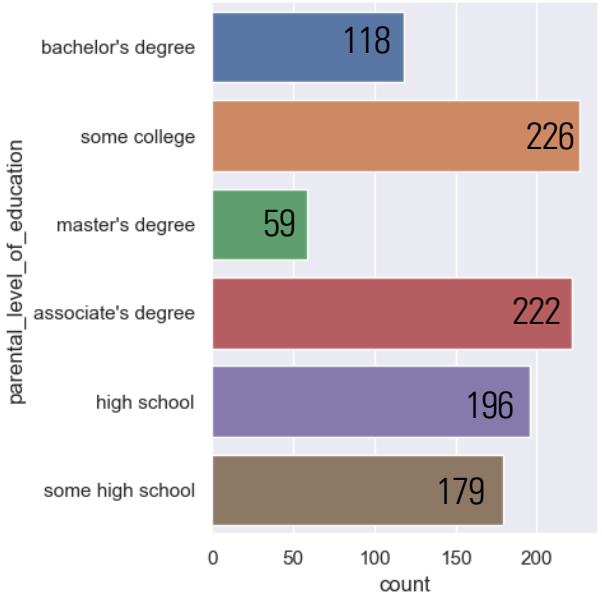
- Dataset of 1000 records
- Standardized dataset with pre-defined values
- Fully filled in data with no blanks

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

	math_score	reading_score	writing_score
count	1000.00000	1000.000000	1000.000000
mean	66.08900	69.169000	68.054000
std	15.16308	14.600192	15.195657
min	0.00000	17.000000	10.000000
25%	57.00000	59.000000	57.750000
50%	66.00000	70.000000	69.000000
75%	77.00000	79.000000	79.000000
max	100.00000	100.000000	100.000000

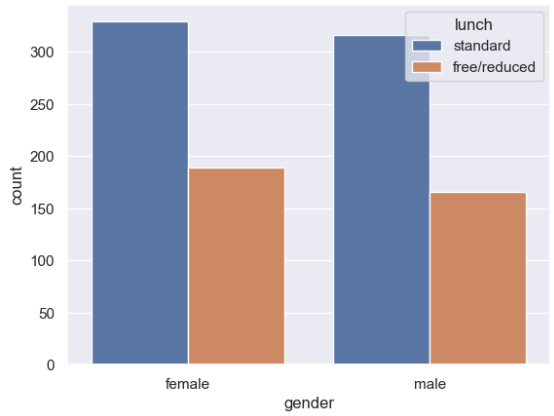
EXPLORATORY DATA ANALYSIS

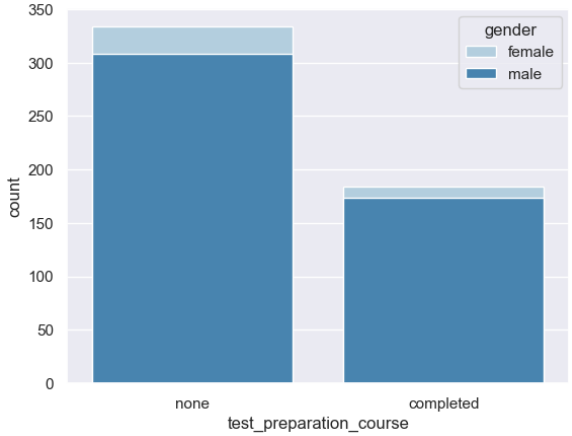


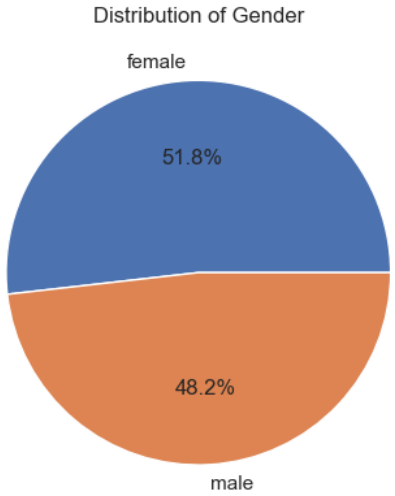


Data Observations: Balanced Data

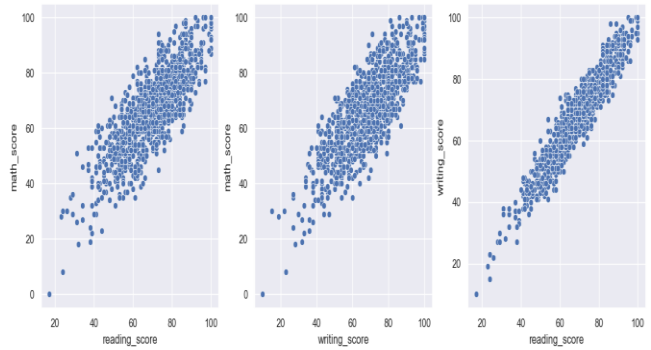
- Data is not skewed to any particular influencing attribute (well distributed)
- Almost equal distribution of gender gives better understanding of the other influencing attributes on the performance outcomes

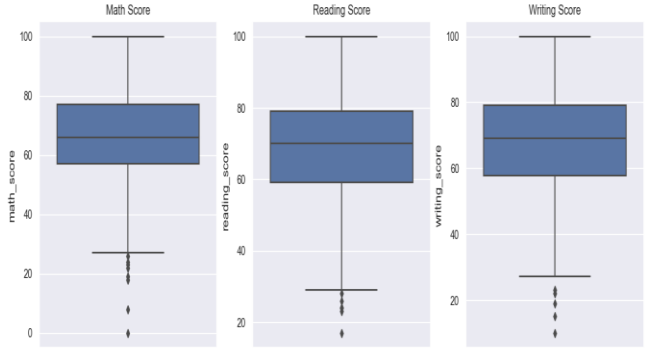


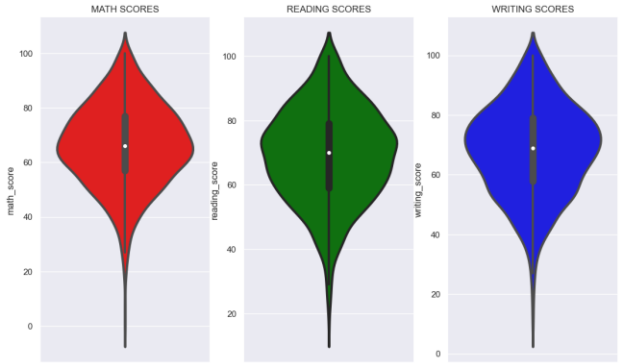




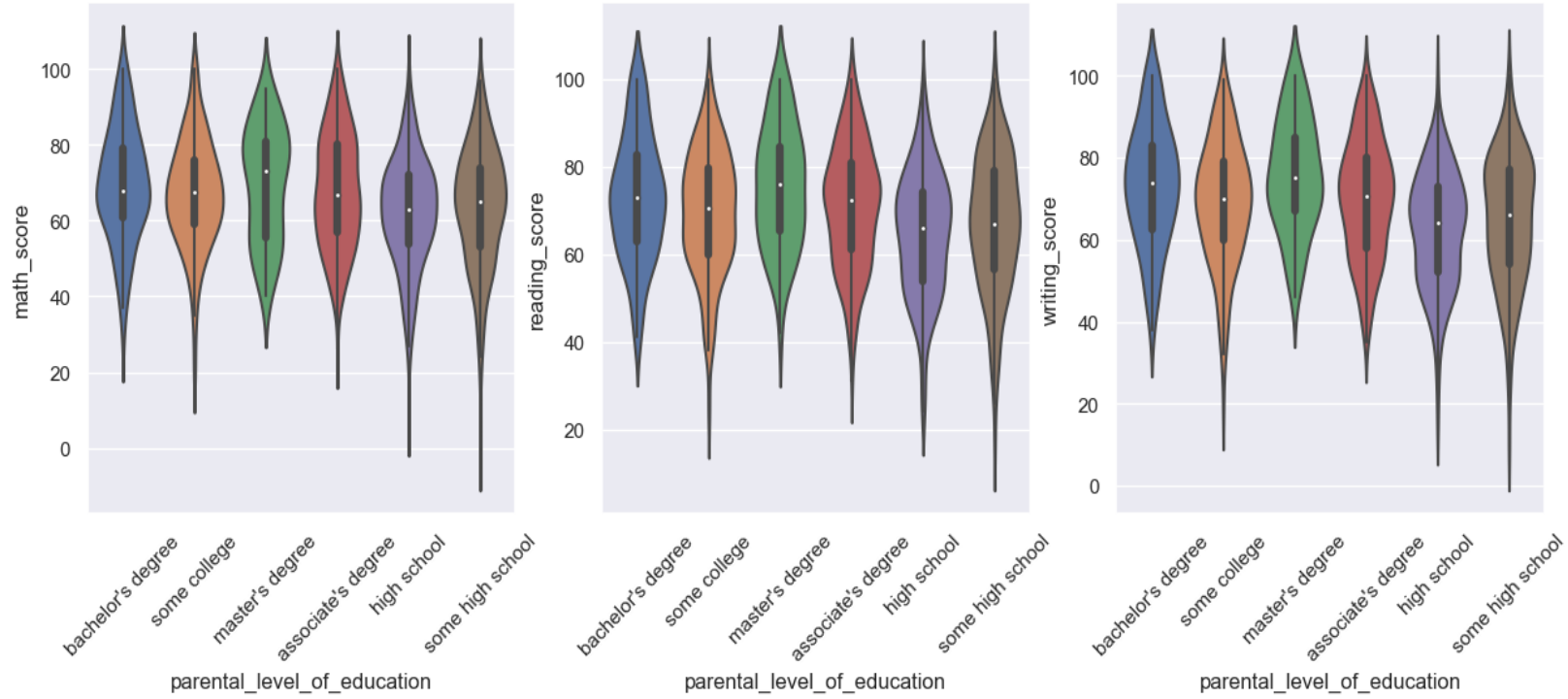
CO-RELATION AMONG PERFORMANCE MEASUREMENT ATTRIBUTES







- Very few outlier cases indicates that though the influencing attributes are many, performance outcome is quite consistent and within the well-defined range of values and median value around 70.
- Box & Violine Charts above shows the data distribution across performance attributes is quite similar and between Math score and Writing score being closer resemblance
- Linear, positive and strong co-relation between across score (measurement) attributes
- Writing and Reading scores have much stronger (denser) co-relation compared to that of math score



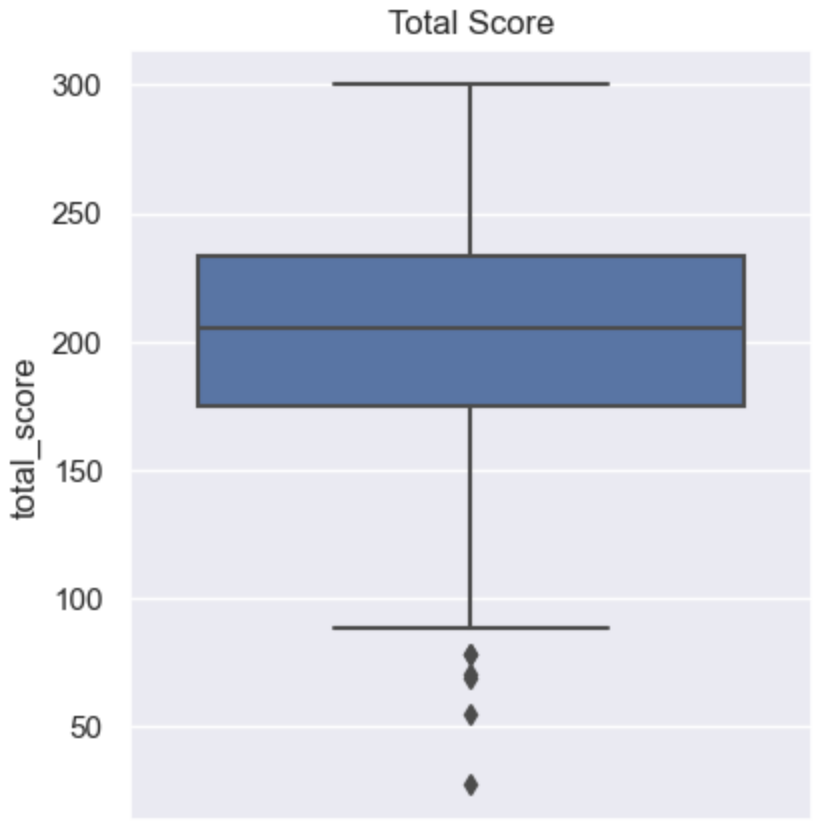
ALL *GIRLS* *BOYS*

Data Observations:

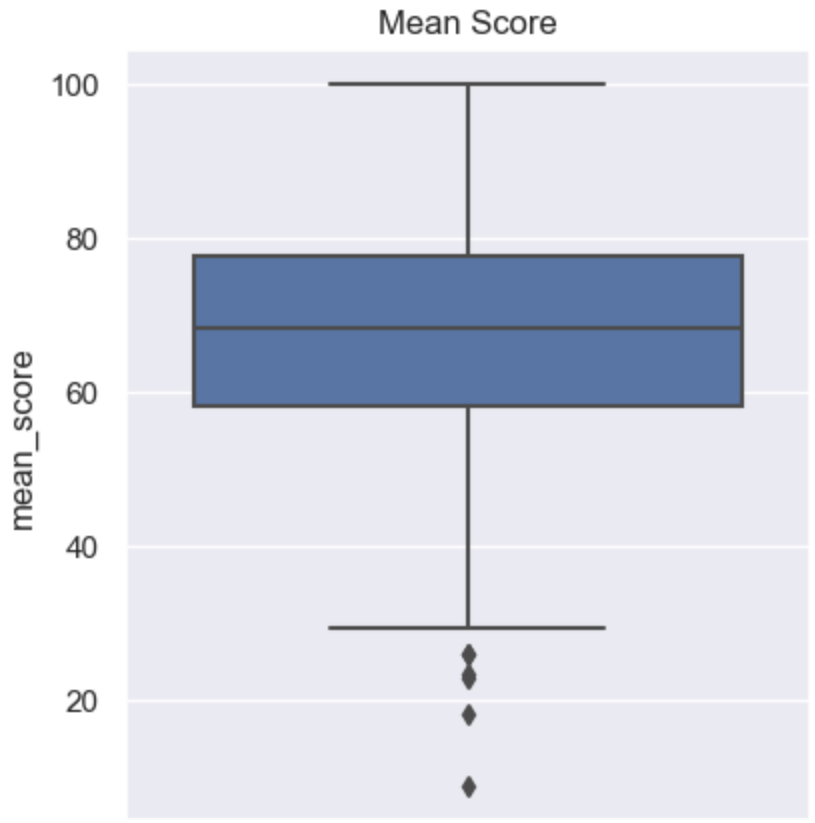
- Median scores are higher and consistent for students with parents having master's degree
- "Some high school" & "High School" categories have more lower scores across subjects
- Performance of students with "Bachelor's degree" are consistent among all subjects

```
studydata['total_score'] = studydata['math_score'] + studydata['reading_score'] + studydata['writing_score']
studydata['mean_score'] = round(studydata['total_score'] / 3, 2)
studydata['median_score'] = studydata[['math_score', 'reading_score', 'writing_score']].median(axis=1)

studydata.head()
```

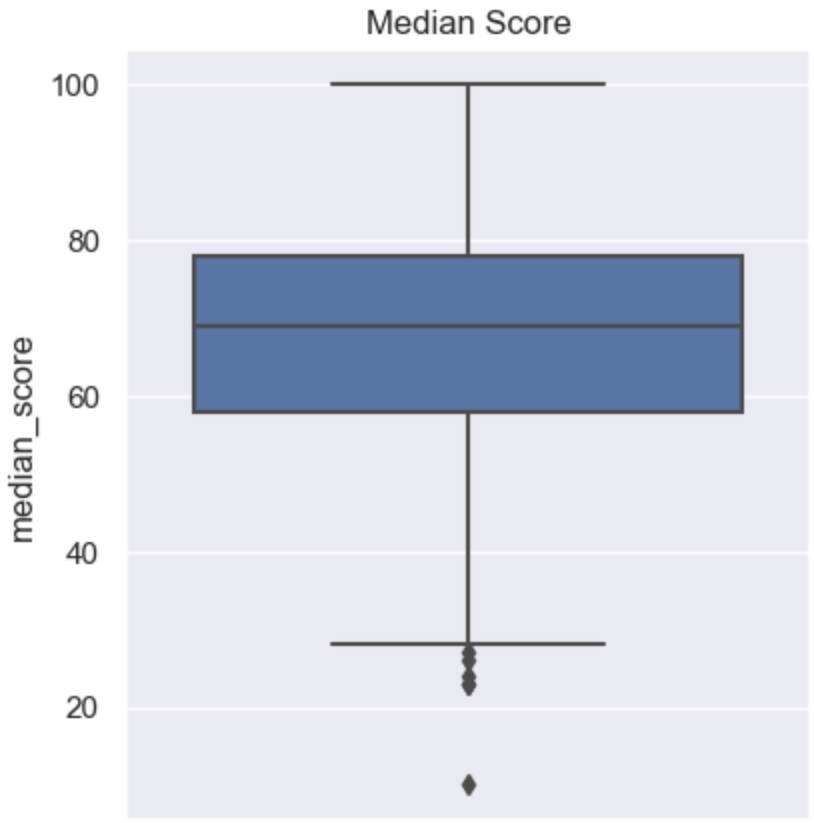
	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score	total_score	mean_score	median_score
0	female	group B	bachelor's degree	standard	none	72	72	74	218	72.67	72.0
1	female	group C	some college	standard	completed	69	90	88	247	82.33	88.0
2	female	group B	master's degree	standard	none	90	95	93	278	92.67	93.0
3	male	group A	associate's degree	free/reduced	none	47	57	44	148	49.33	47.0
4	male	group C	some college	standard	none	76	78	75	229	76.33	76.0



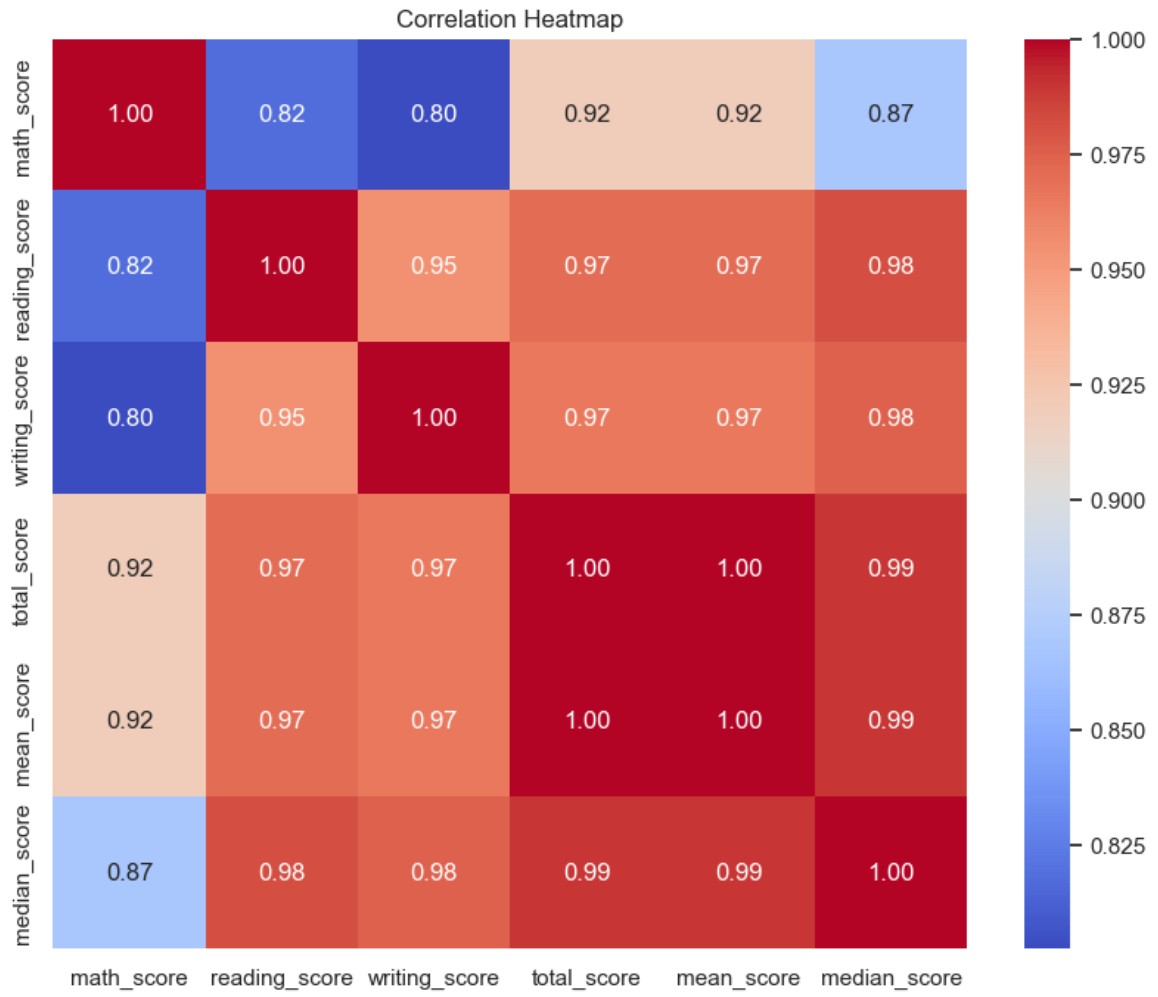
ALL



GIRLS

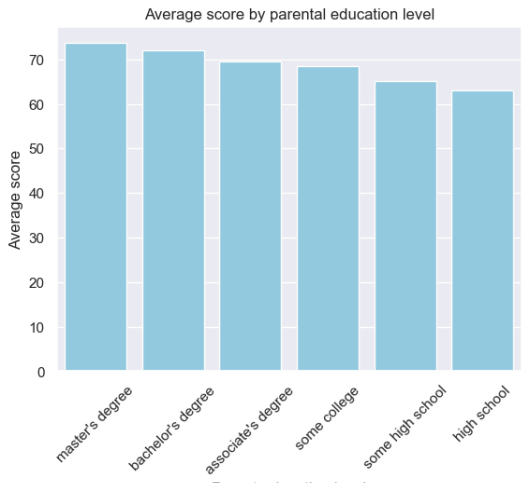


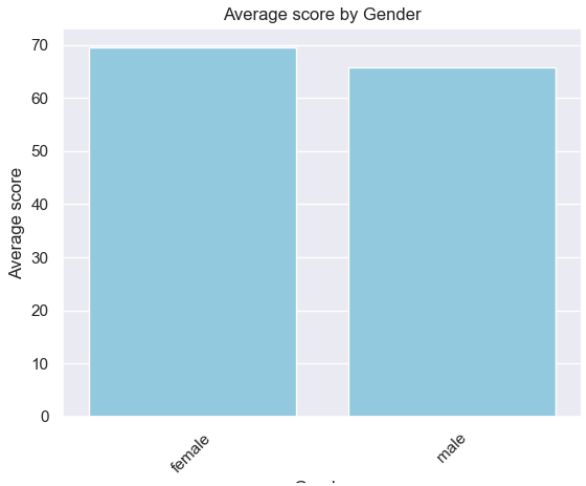
BOYS

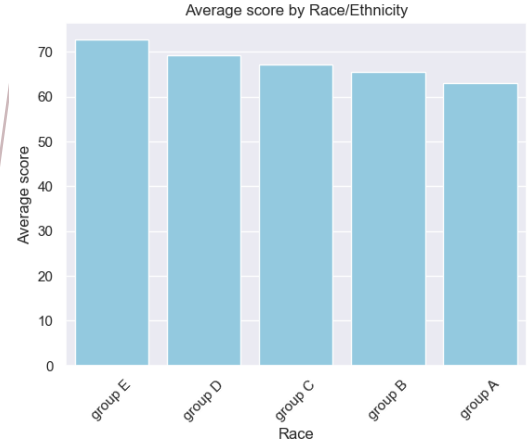


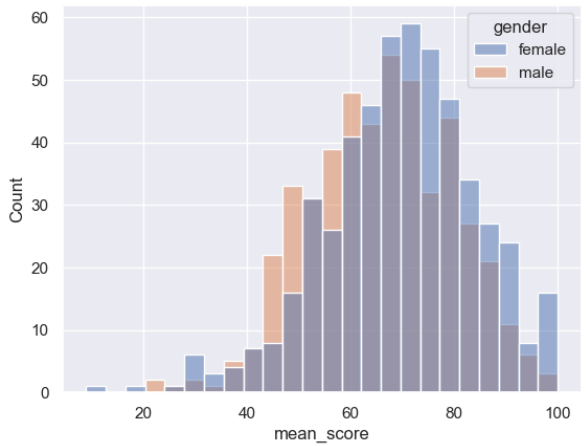
Data Observations:

- Writing and Reading scores have weaker correlation to Math score.
- Writing has a very storing co-relation with reading.



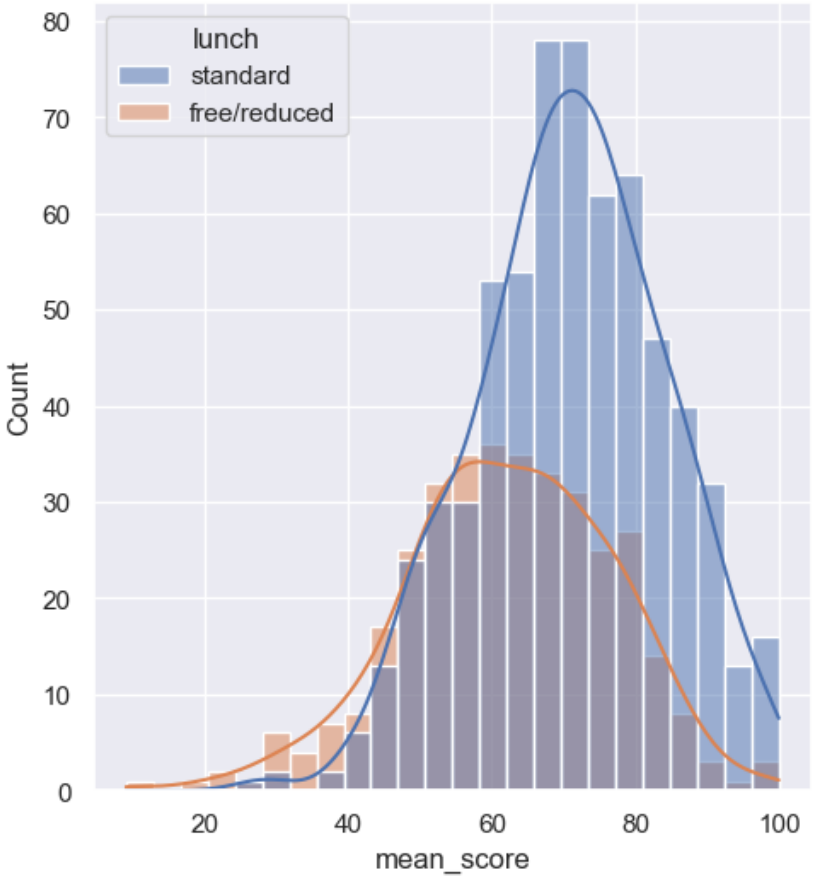




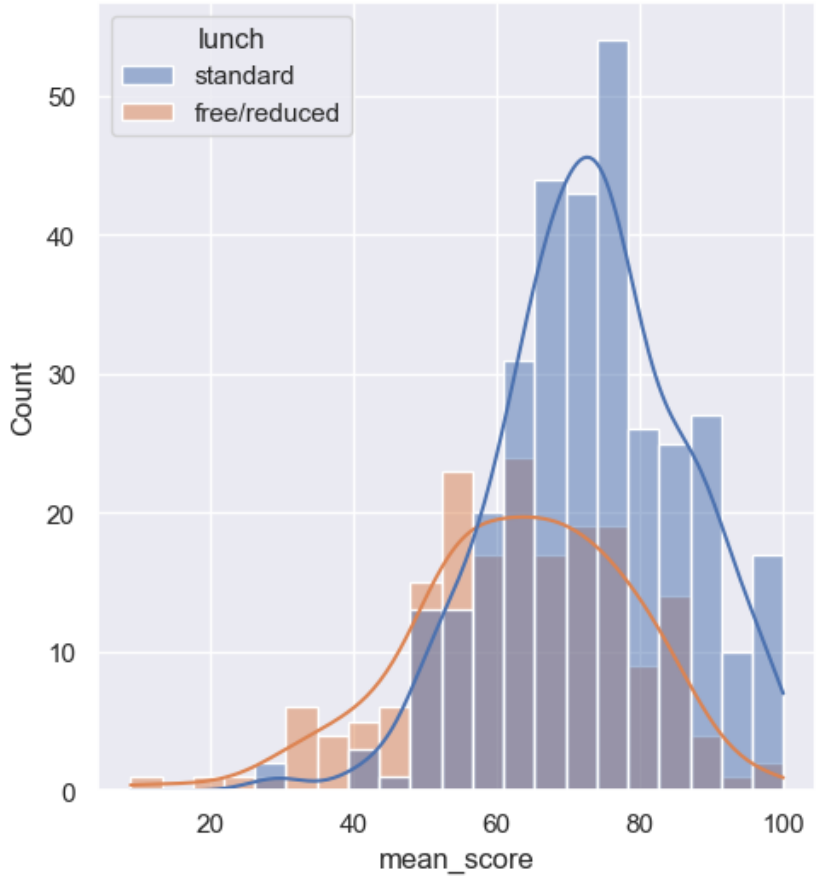


Data Observations:

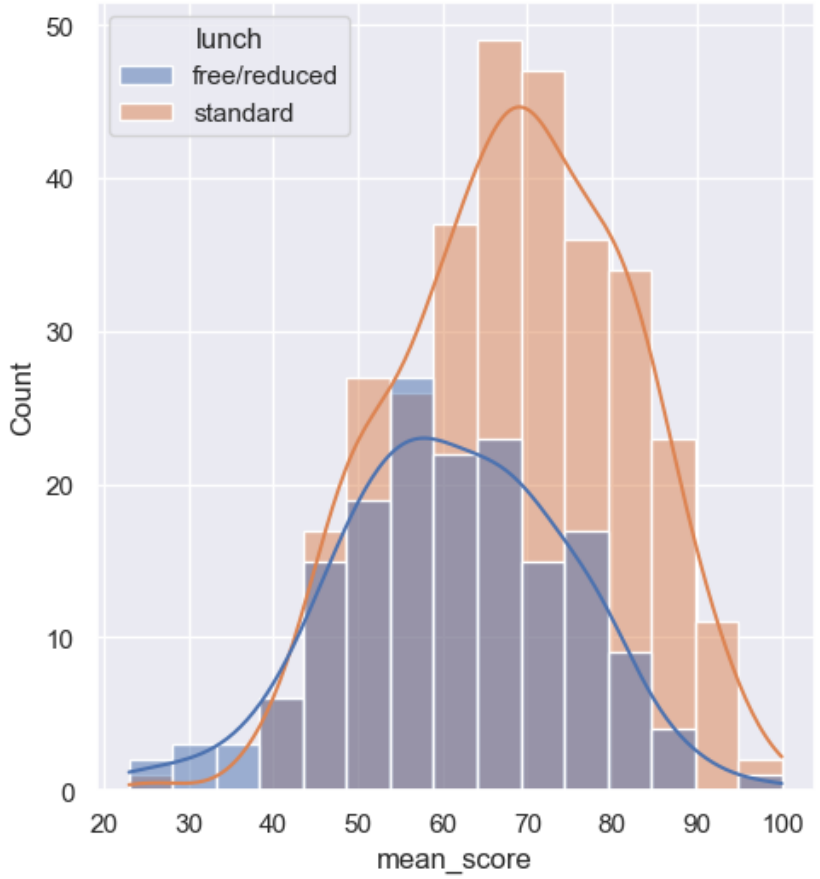
- Female students average score is higher
- Higher the education level of the parent, better the average score
- It is also observed that average score doesn't vary significantly between races



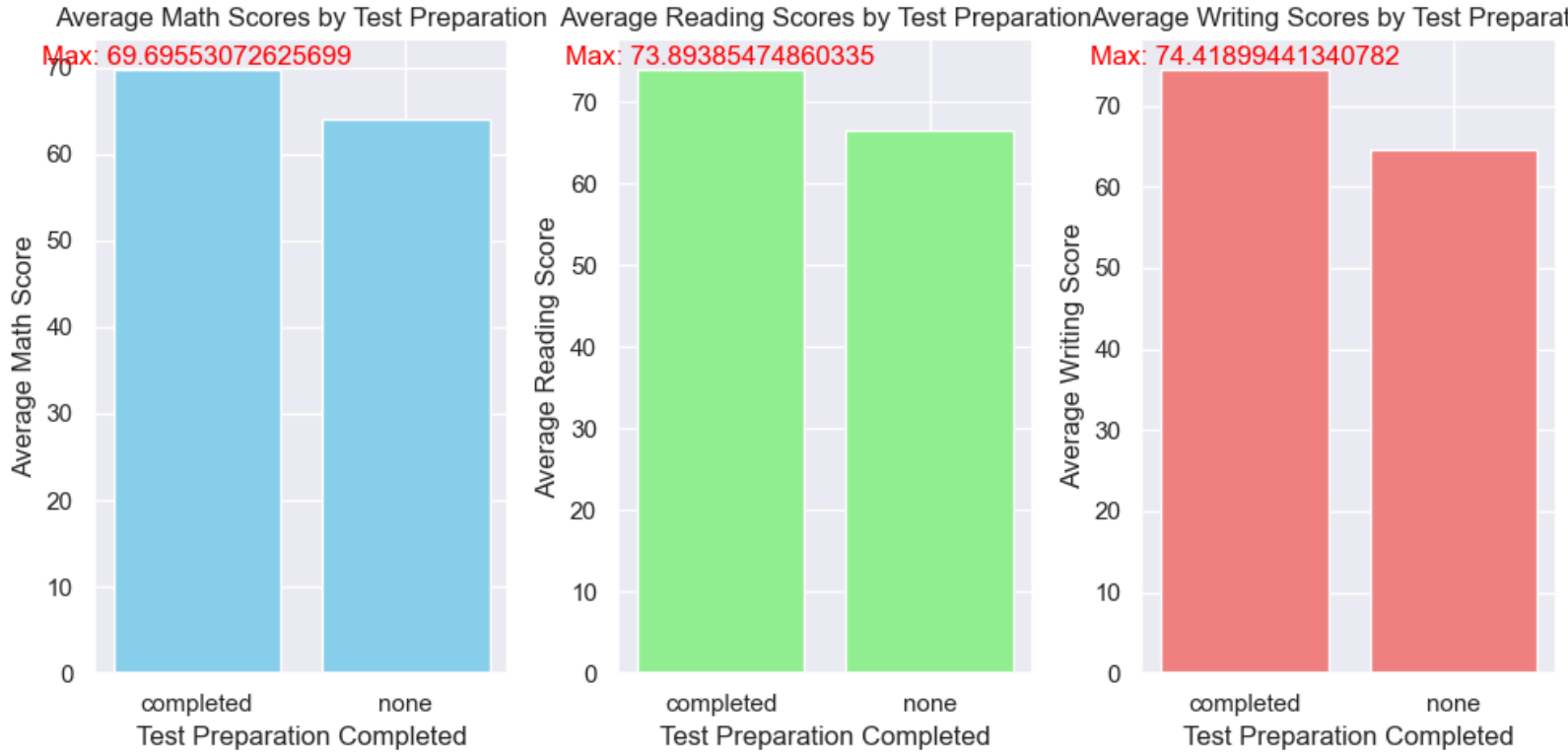
ALL



GIRLS



BOYS

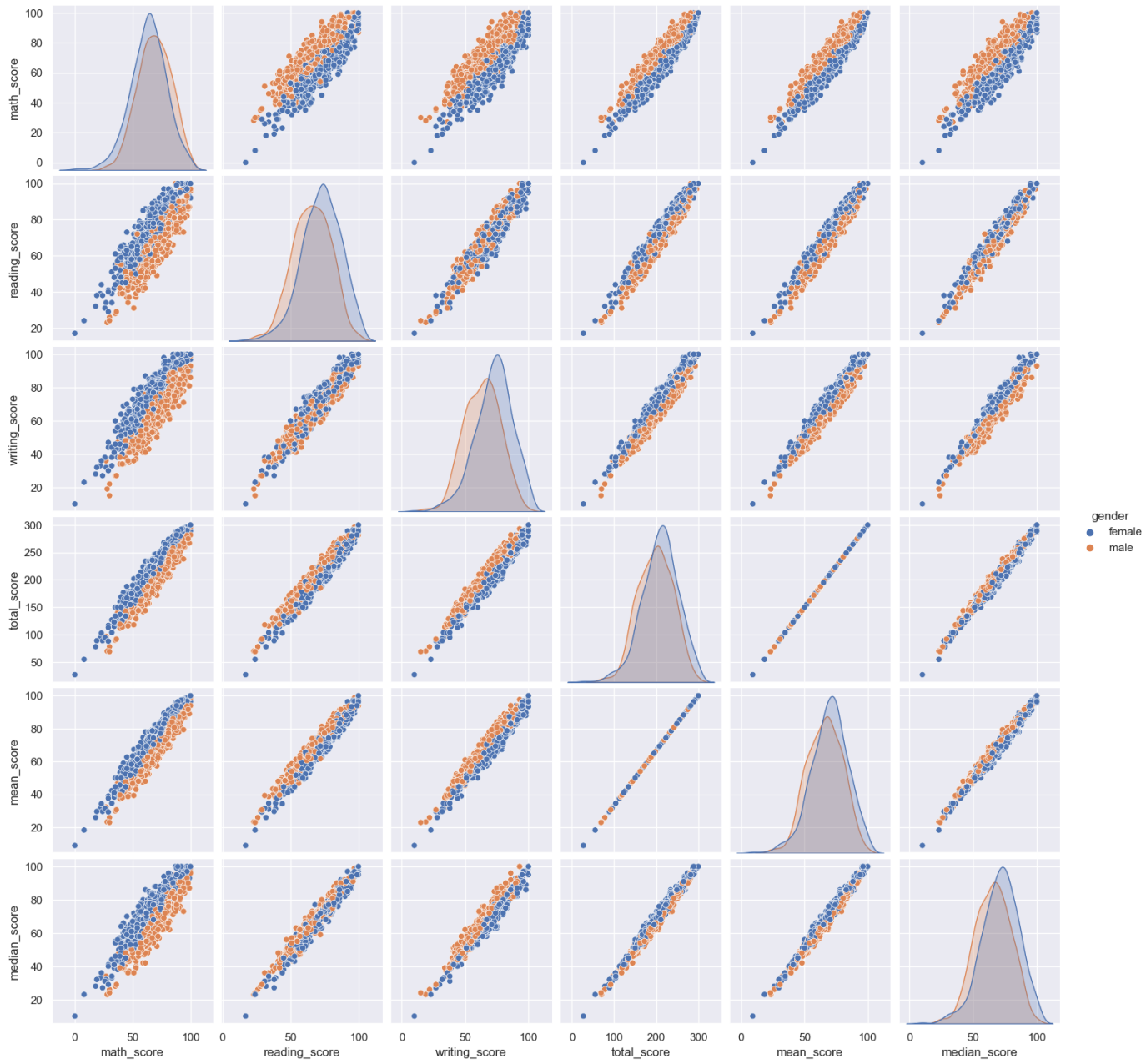


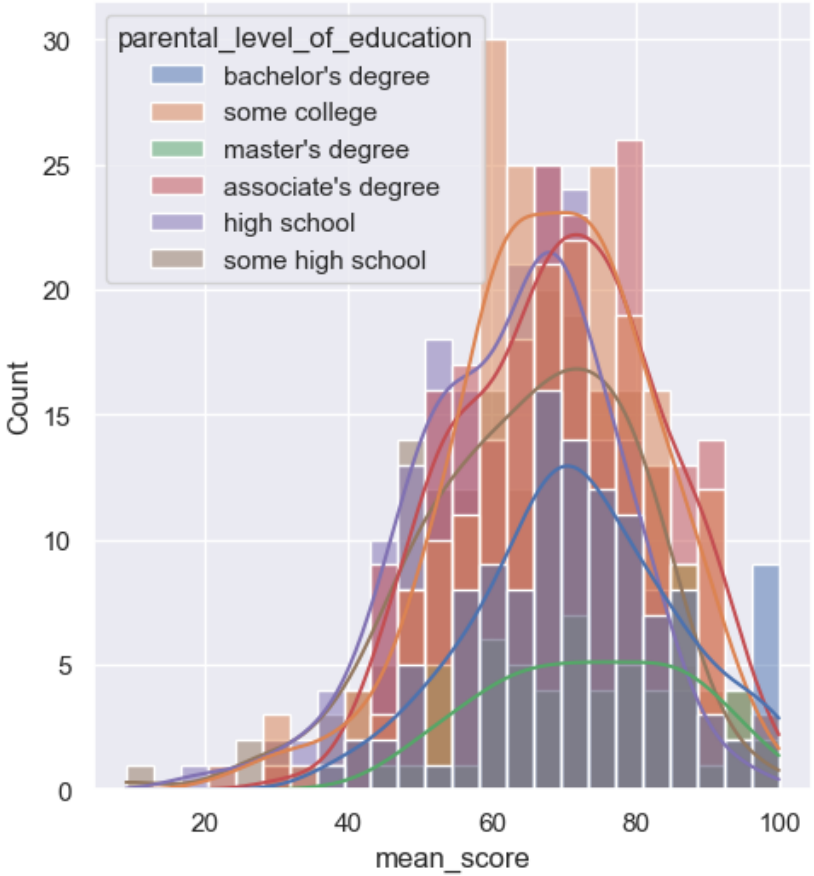
Data Observation:

- The students who have completed their test preparation have a better average score in all the subjects compared to the student.
- Although the difference between the average math score of students who have completed preparation v/s the students who didn't complete preparation is about 4 marks which is quite low, this is not the case with average reading score and writing score where the difference is larger.

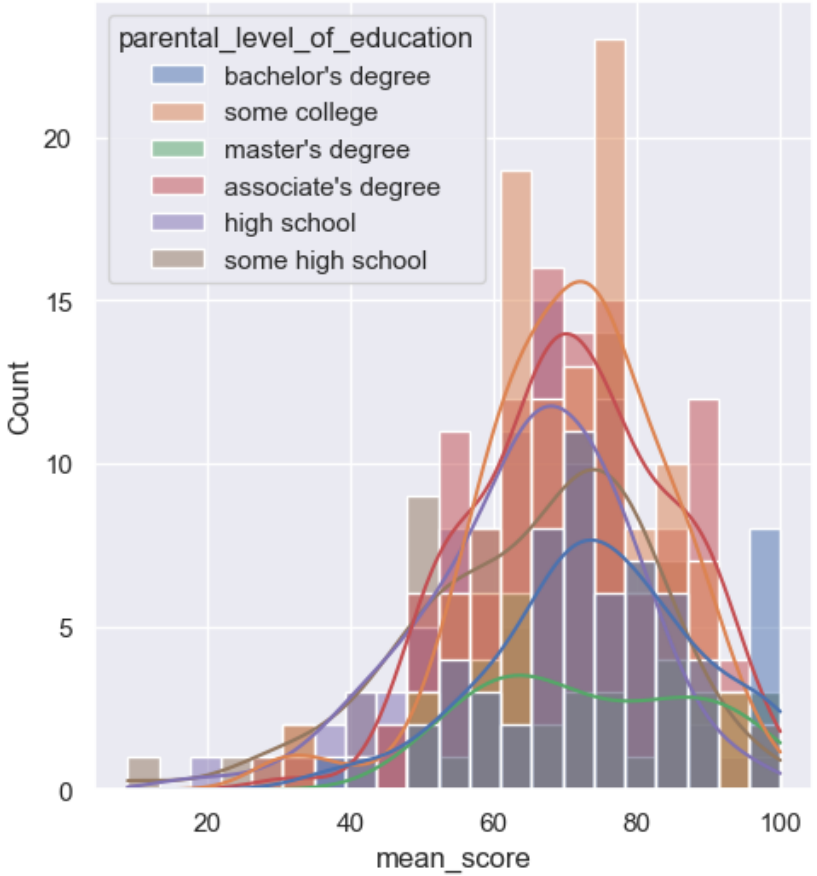
Data Observations:

- Female students average score is higher
- Higher the education level of the parent, better the average score.

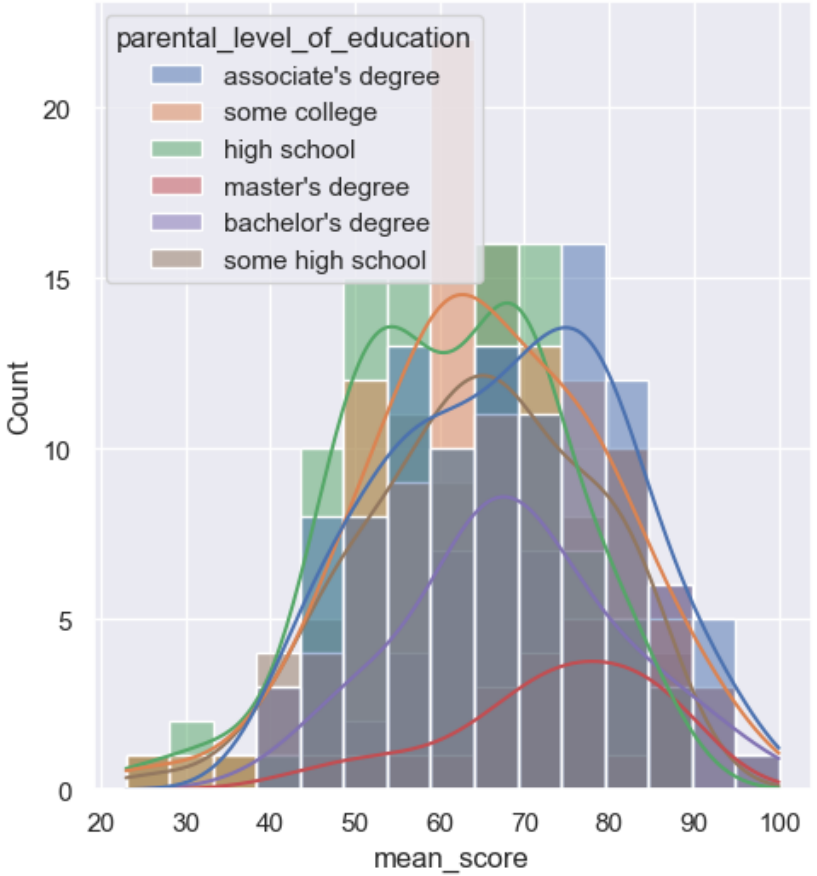




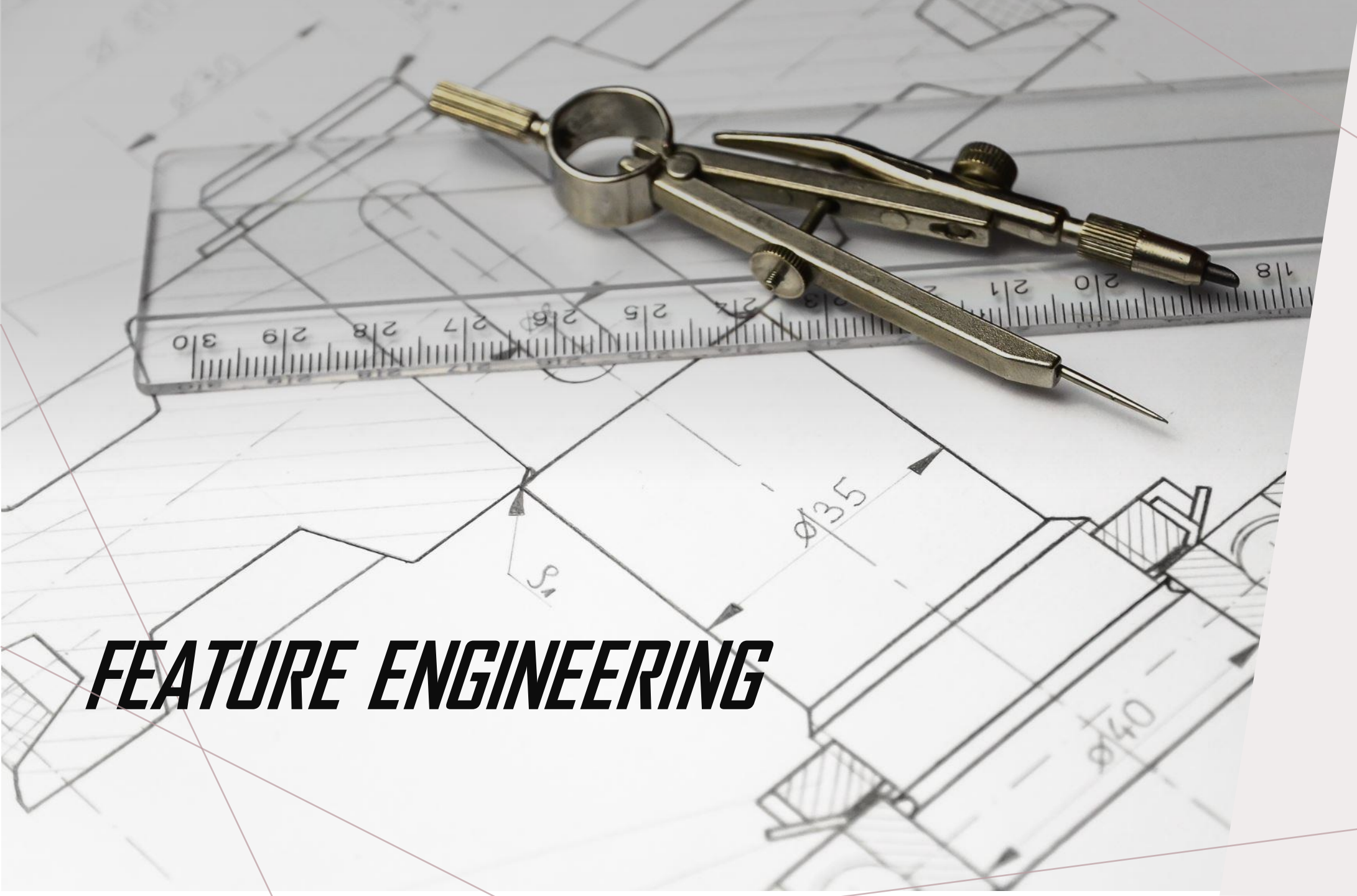
ALL



GIRLS



BOYS



FEATURE ENGINEERING

WHAT IS FEATURE ENGINEERING?

- Feature engineering refers to manipulation — addition, deletion, combination, mutation — of your data set to improve machine learning model training, leading to better performance and greater accuracy.

- Effective feature engineering is based on sound knowledge of the problem at hand and the available dataset.

- Feature engineering in ML is more than selecting the appropriate features and transforming them. Not only does feature engineering prepare the dataset to be compatible with the algorithm, but it also improves the performance of the machine learning models.

1. TOTAL SCORE

- Total_score = reading_score + writing_score + math_score
- Offers insights in student proficiency across multiple subjects
- Provides a more comprehensive picture than just individual subject scores
- Possible response variable in predictive modelling

2. MEAN SCORE

- mean_score = (total_score/3)
- Measure for central tendency of a student's score across 3 subjects
- Benchmark for comparing a student's performance
- Helps in analysing strengths and weaknesses of a student
- Possible response variable

3. BOOLEAN COLUMNS

- Columns with only two categories were transformed to boolean
- Eg (Lunch, Gender, Test_preparation_course) -> (lunchbool, genderbool, testbool)
- This makes it easier for the algorithm to interpret and process the data as it represents a clear dichotomy between 2 states
- Many ML algorithms such as linear regression and decision trees require numeric input
- Simplifies splitting splitting criteria at each node

3. BOOLEAN COLUMNS

```
studydata['lunchbool'] = studydata['lunch'].replace({'standard': True, 'free/reduced': False})
```

```
studydata['testbool'] = studydata['test_preparation_course'].replace({'completed': True, 'none': False})
```

```
studydata['genderbool'] = studydata['gender'].replace({'female': True, 'male': False})
```

	total_score	mean_score	median_score	race_ethnicity_encoded	lunchbool	\
0	218	72.67	72.0	1	True	
1	247	82.33	88.0	2	True	
2	278	92.67	93.0	1	True	
3	148	49.33	47.0	0	False	
4	229	76.33	76.0	2	True	

	testbool	genderbool
0	False	True
1	True	True
2	False	True
3	False	False
4	False	False

4. DIVISION

- It is impossible to predict the exact score of a student based on categorical variables
- However, it is easier to classify them into categories created on the basis of the range of marks they are scoring
- Creating the division column makes the data more granular and easy to interpret compared to a continuous scale

4. DIVISION

```python
import pandas as pd

# Load the dataset
df = pd.read_csv("studydata.csv")

# Create 'totalscore' column
df['totalscore'] = df['math_score'] + df['reading_score'] + df['writing_score']

# Calculate percentiles
percentiles = df['totalscore'].quantile([0.25, 0.4, 0.7])

# Function to assign division based on percentile
def assign_division(score):
    if score > percentiles[0.7]:
        return 'First Division'
    elif score > percentiles[0.4]:
        return 'Second Division'
    elif score > percentiles[0.25]:
        return 'Third Division'
    else:
        return 'Fail'

# Create 'division' column
df['division'] = df['totalscore'].apply(assign_division)

# Display the updated DataFrame
df.head()
```

4. DIVISION

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score	totalscore	division
0	female	group B	bachelor's degree	standard	none	72	72	74	218	Second Division
1	female	group C	some college	standard	completed	69	90	88	247	First Division
2	female	group B	master's degree	standard	none	90	95	93	278	First Division
3	male	group A	associate's degree	free/reduced	none	47	57	44	148	Fail
4	male	group C	some college	standard	none	76	78	75	229	First Division

ENCODING

- Categorical variables need to be converted to numerical data types because most ML algorithms require numerical input
- Unlike boolean columns, encoding is able to represent the ordinal relationship between the categories

Label Encoding	One Hot Encoding
Categorical variables with ordinal relationship	Categorical variables with multiple categories
Converts categories into numerical labels	Creates new binary columns for each category
Does not increase dimensionality	Increases dimensionality of the feature space

ENCODING

In [2]:
```
# Perform one-hot encoding on categorical variables
df_encoded = pd.get_dummies(df, columns=['gender', 'test_preparation_course', 'lunch'])

# Display the updated DataFrame
df_encoded.head()
```

Out[2]:

ıg_score	totalscore	division	gender_female	gender_male	test_preparation_course_completed	test_preparation_course_none	lunch_free/reduced	lunch_standard
74	218	Second Division	1	0	0	1	0	1
88	247	First Division	1	0	1	0	0	1
93	278	First Division	1	0	0	1	0	1
44	148	Fail	0	1	0	1	1	0
75	229	First Division	0	1	0	1	0	1

MACHINE LEARNING MODELS

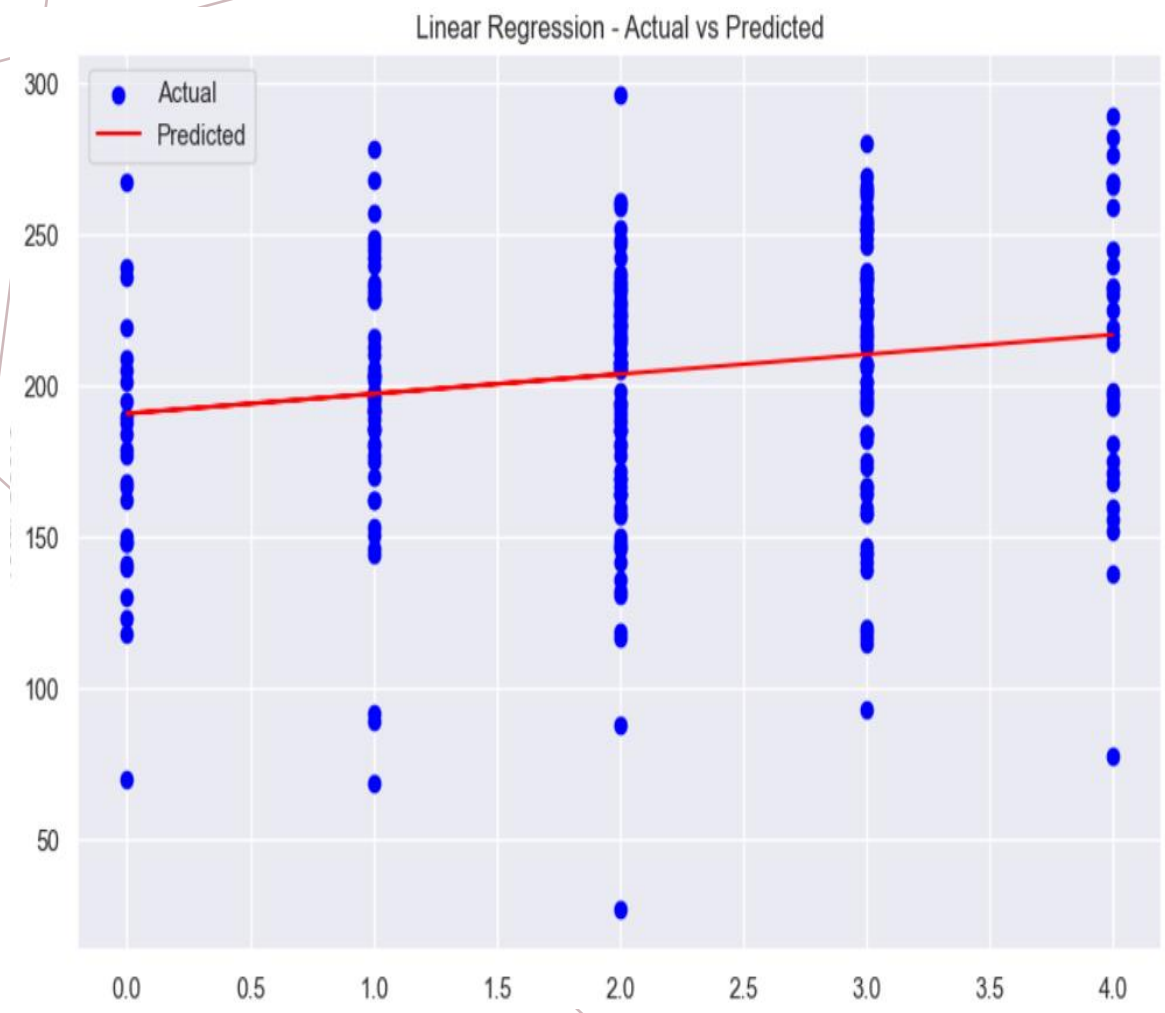
Linear Regression, Decision Tree, Random Forest, XGBoost, Support Vector Machine

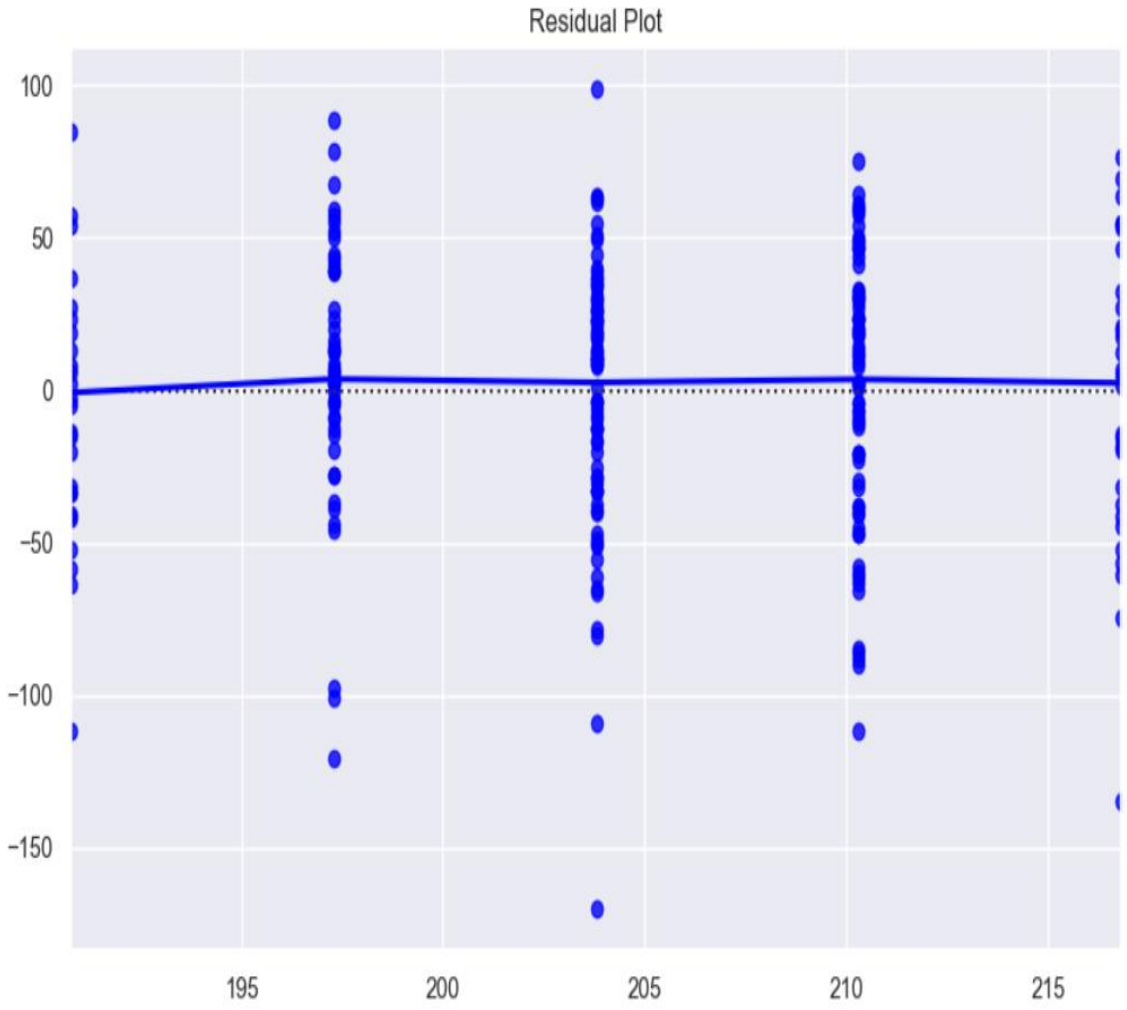
1. LINEAR REGRESSION

- Multivariate linear regression is a statistical technique used to model the relationship between multiple independent variables (predictors) and a single dependent variable (response).
- Regression Model: division = a1 x genderbool + a2 x lunchbool + a3 x testbool + b
- Metrics:

Mean Squared Error (MSE): MSE is a measure of the average squared difference between the actual values and the predicted values by the regression model.

Variance: Variance measures the variability or spread of the predicted values from the mean value of the target variable.


Linear Regression - Actual vs Predicted
Actual
Predicted
300
250
200
150
100
50
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Residual Plot
100
50
0
−50
−100
−150
195
200
205
210
215

ACCURACY RESULTS

- Metrics:

R squared (R^2): R-square is a goodness-of-fit measure for linear regression models.

$$R^2 = \frac{\text{Variance explained by the model}}{\text{Total variance}}$$

Mean Squared Error (MSE): measures how close a regression line is to a set of data points.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2$$

```python
# Evaluate the model
r_squared = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

print("R-squared:", r_squared)
print("Mean Squared Error:", mse)
```

```
R-squared: 0.016869192414713297
Mean Squared Error: 1979.565010926866
```

2. MULTI CLASS CLASSIFICATION TREE

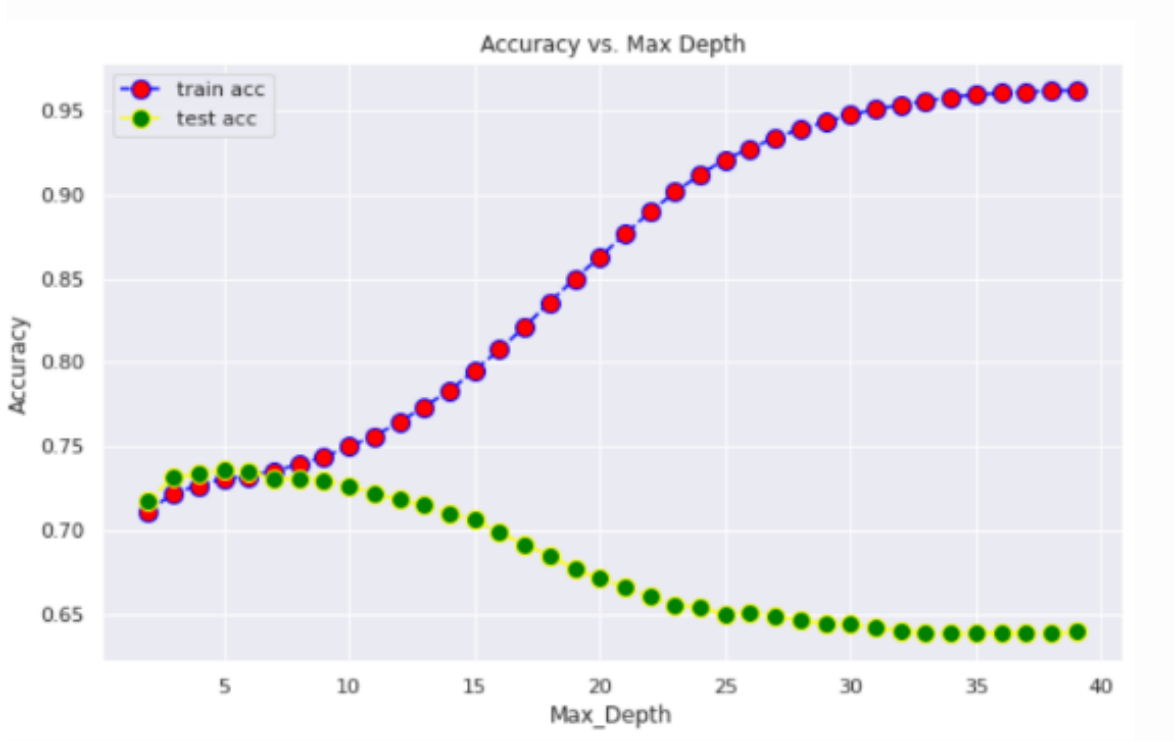
- A multi-class classification tree is a decision tree-based machine learning algorithm that recursively partitions the feature space to classify instances

It is made up of 2 entities:

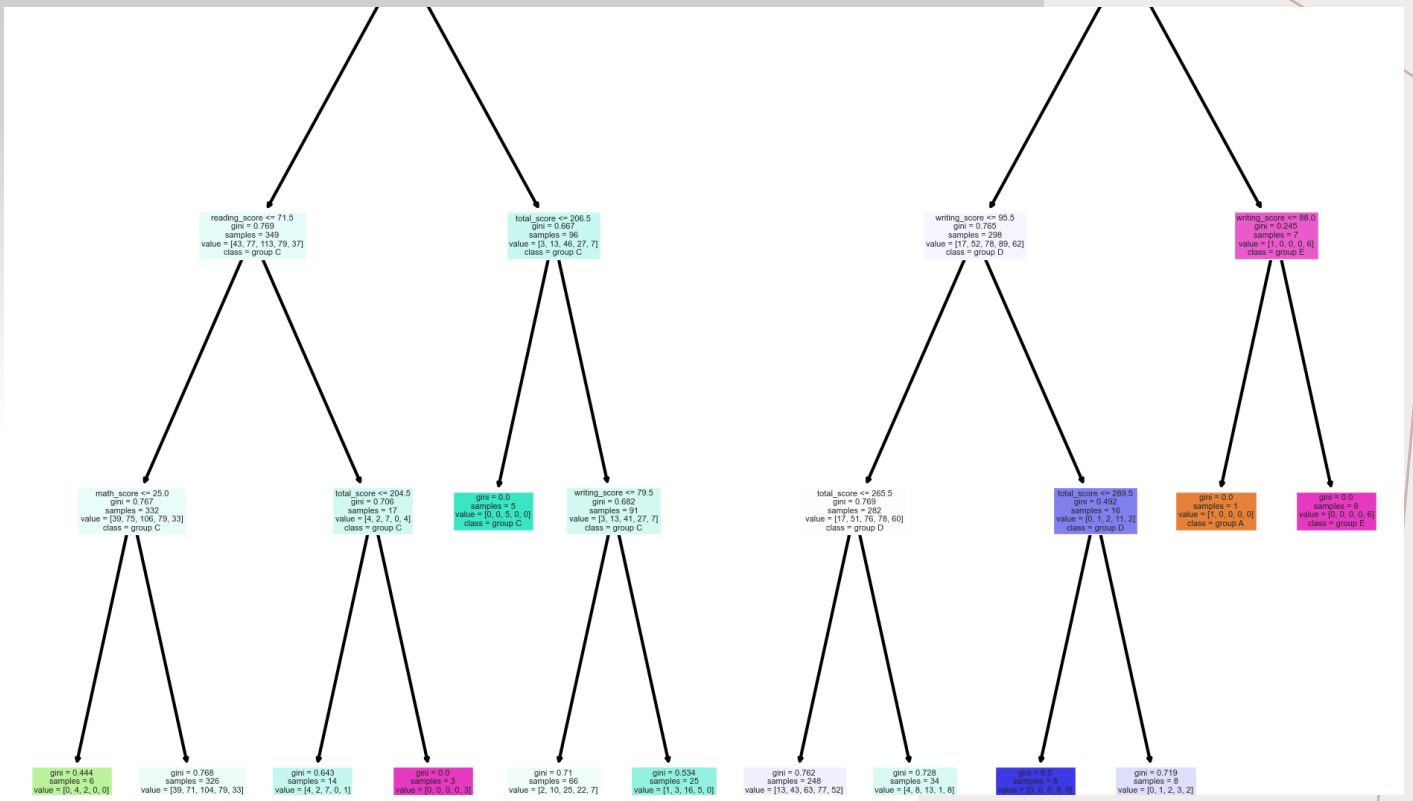
- Decision nodes: where data is split
- Leaves: Decisions/Final Outcomes
- We have made use of Label Encoding where necessary in order to simplify the input
- Metrics: gini score, False Positive/False Negative rates, Goodness of fit of tree

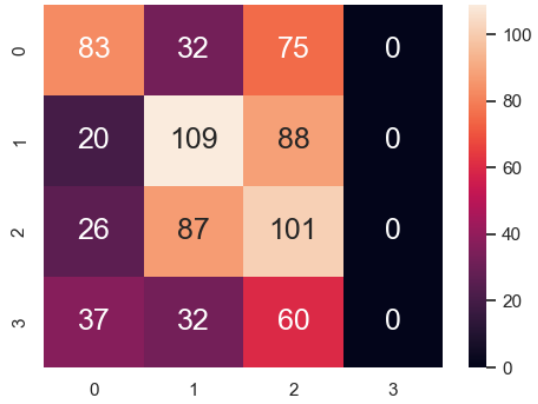
MDT VISUALISATION

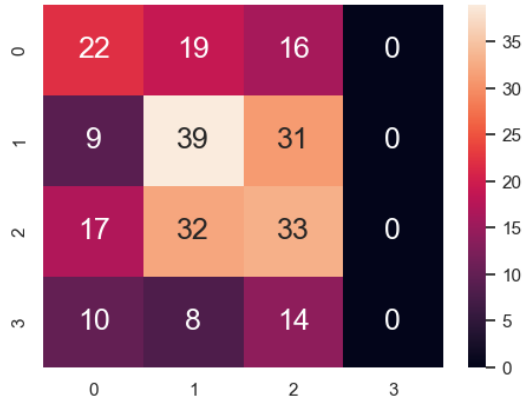
- After multiple combinations of predictor variables, it was determined which trees resulted in maximum accuracy.
- Optimal tree depth was determined to be 4



MDT VISUALISATION








```
Goodness of Fit of Model          Train Dataset
Classification Accuracy           : 0.7173333333333334

Goodness of Fit of Model          Test Dataset
Classification Accuracy           : 0.712
```

ACCURACY RESULTS

3. RANDOM FOREST

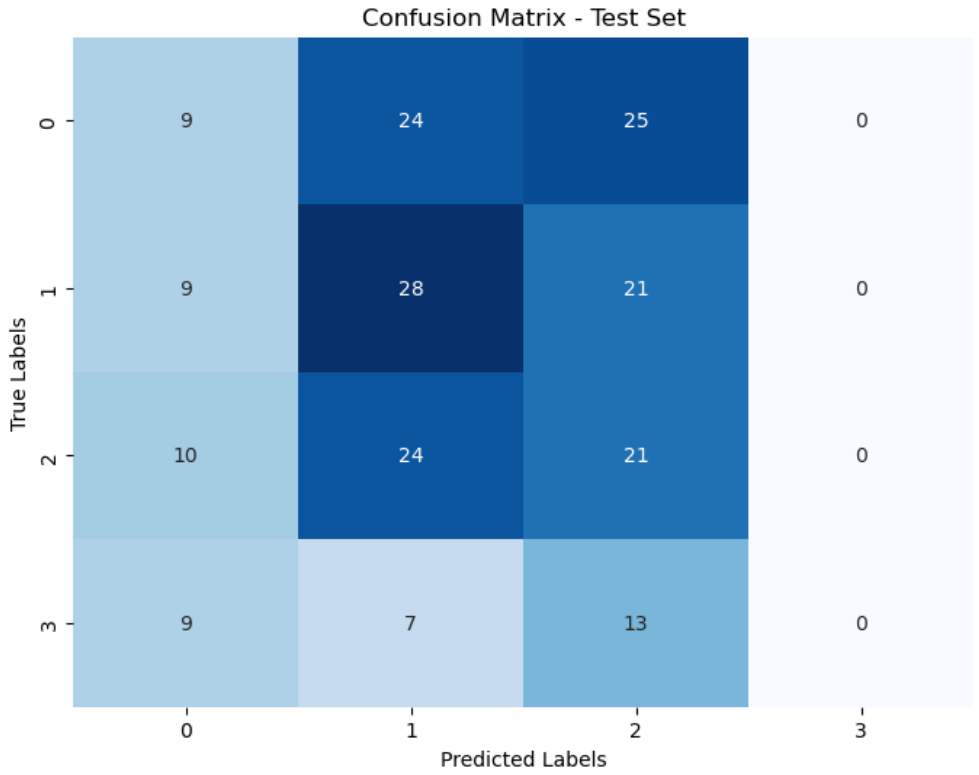
- Random forest is an ensemble learning technique that builds multiple decision trees and combines their predictions to improve accuracy and robustness.
- Each tree is trained on a random subset of features at each split, introducing diversity among the trees and reducing correlation between them.
- In classification tasks, the final prediction is made by majority voting among the individual trees' predictions
- We have made use of one hot encoding to convert categorical columns to forms which can be implemented as a random forest

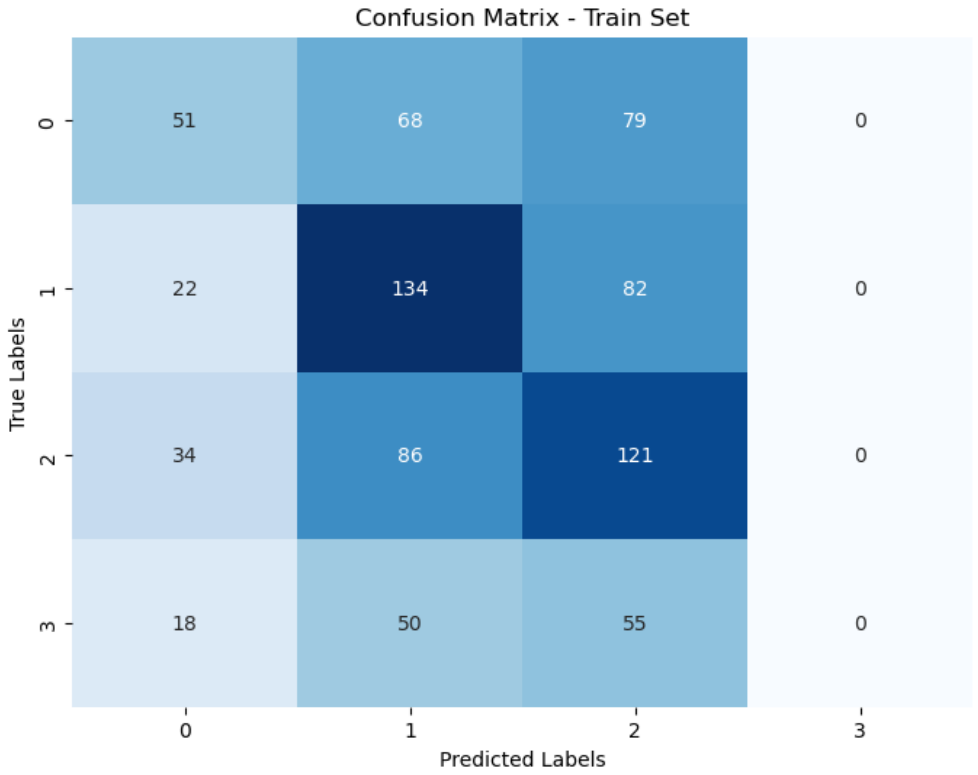
RANDOM FOREST VS DECISION TREE

- Decision Tree combines some decisions, whereas Random Forest combines several decision trees.
- Random Forest is long and slow, Decision Tree is fast and operates easily on large data sets
- Random Forest model needs rigorous training but are generally more stable and reliable in terms of predictions
- The main distinction between the two is that Random Forest does not rely on a single decision. It assembles randomized decisions based on many decisions and then creates a final decision depending on the majority.

METRICS

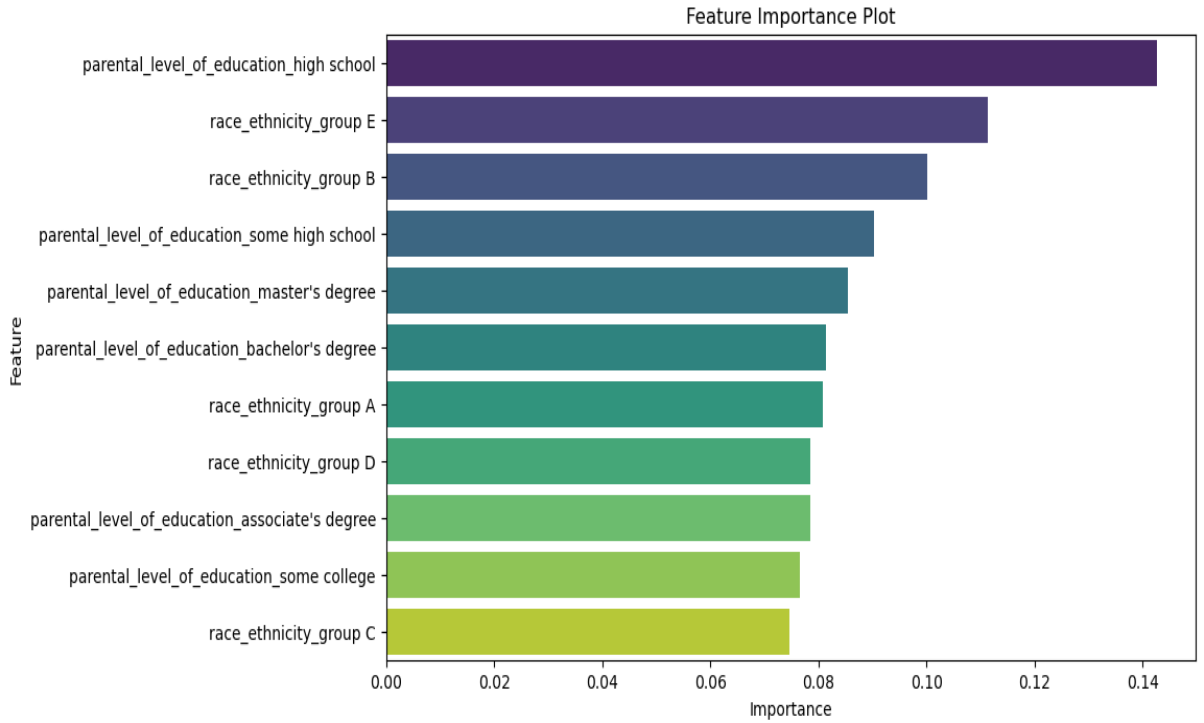
- Precision: Precision measures the accuracy of positive predictions made by the model. It is the ratio of correctly predicted positive observations to the total predicted positives, indicating how many of the instances classified as positive are actually positive.
- Recall: It is the ratio of correctly predicted positive observations to the actual positives in the data.
- F-1 score: $$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$
- Support





RANDOM FOREST RESULTS

```
Accuracy on Train Dataset: 0.3975
Accuracy on Test Dataset: 0.375

Classification Report on Test Dataset:
                precision    recall  f1-score   support

          Fail       0.40      0.60      0.48        58
 First Division      0.47      0.40      0.43        58
Second Division      0.27      0.31      0.29        55
 Third Division      0.00      0.00      0.00        29

      accuracy                           0.38       200
     macro avg       0.28      0.33      0.30       200
  weighted avg       0.33      0.38      0.34       200


Confusion Matrix on Test Dataset:
[[35  4 19  0]
 [17 23 18  0]
 [20 18 17  0]
 [16  4  9  0]]
```



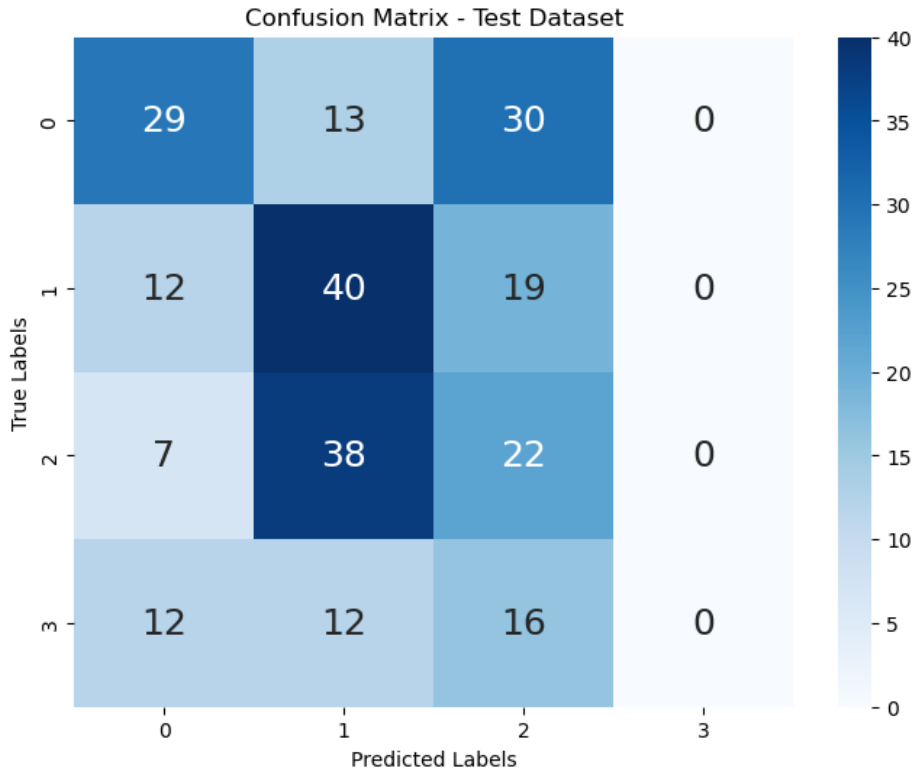
RANDOM FOREST RESULTS

4. XGBOOST

- short for Extreme Gradient Boosting
- It works by sequentially adding weak learners (decision trees) to the model and improving upon them by focusing on the instances that were previously misclassified.
- In our analysis, we utilized XGBoost to build predictive models for classifying student academic performance categories and total scores, leveraging its robustness and ability to handle complex datasets effectively.
- We have made use of one hot encoding to convert categorical columns to forms which can be implemented in XGBoost

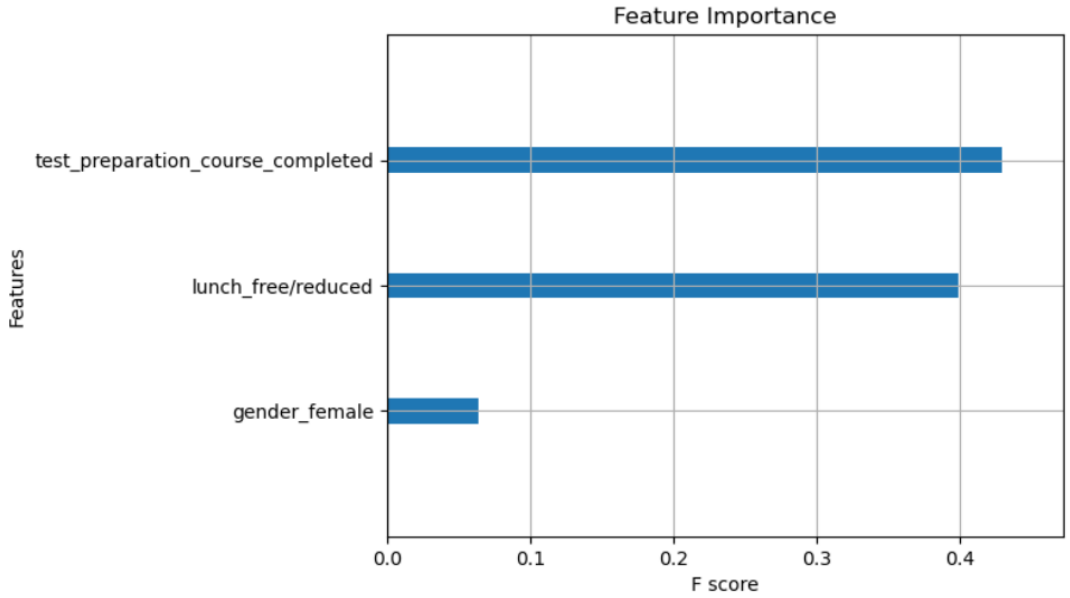
METRICS (SAME AS RF)

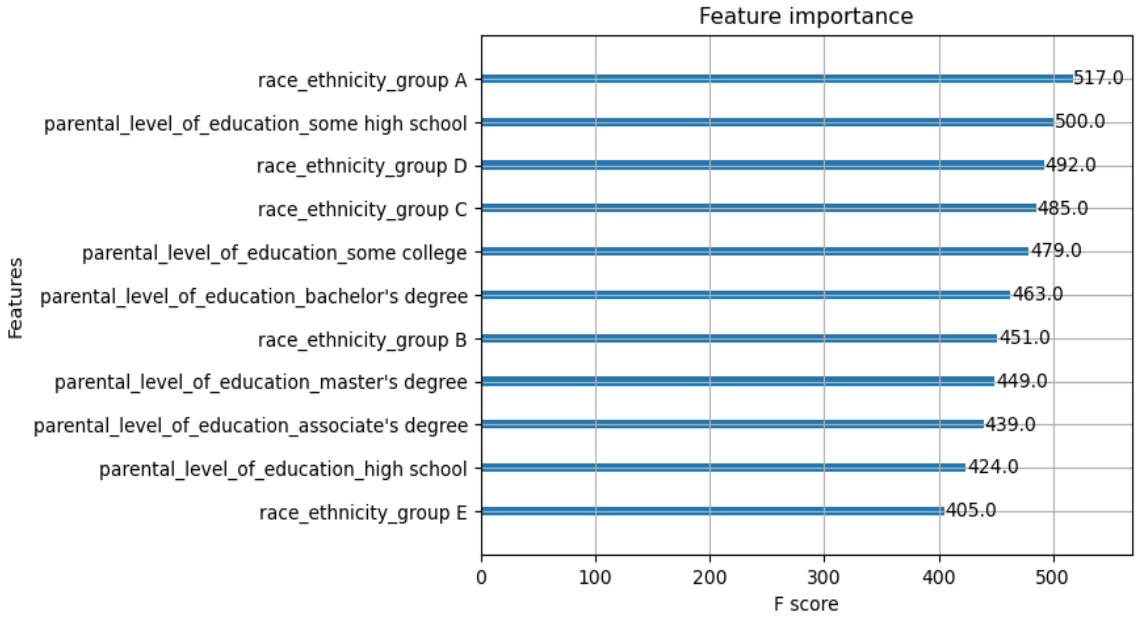
- Precision: Precision measures the accuracy of positive predictions made by the model. It is the ratio of correctly predicted positive observations to the total predicted positives, indicating how many of the instances classified as positive are actually positive.
- Recall: It is the ratio of correctly predicted positive observations to the actual positives in the data.
- F-1 score: $$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$
- Support




```
Accuracy on Train Dataset: 0.3973333333333333
Accuracy on Test Dataset: 0.364

Classification Report on Test Dataset:
              precision    recall  f1-score   support

           0       0.48      0.40      0.44        72
           1       0.39      0.56      0.46        71
           2       0.25      0.33      0.29        67
           3       0.00      0.00      0.00        40

    accuracy                           0.36       250
   macro avg       0.28      0.32      0.30       250
weighted avg       0.32      0.36      0.33       250


Confusion Matrix on Test Dataset:
[[29 13 30  0]
 [12 40 19  0]
 [ 7 38 22  0]
 [12 12 16  0]]
```

XGBOOST RESULTS





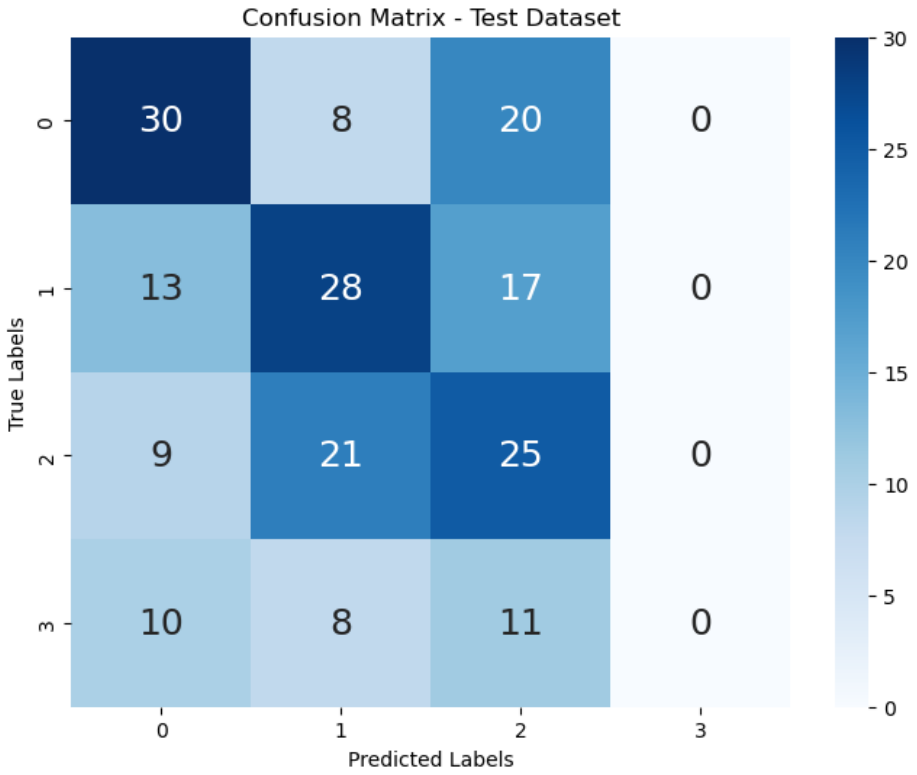
XGBOOST FEATURE IMPORTANCE PLOTS

5. SUPPORT VECTOR MACHINE

- Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.
- SVM works by finding the hyperplane that best separates different classes in the feature space, maximizing the margin between the classes while minimizing classification errors.
- In our analysis, SVM was employed to classify student academic performance categories based on socioeconomic variables such as race/ethnicity and parental education level.
- It is worth noting that SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation

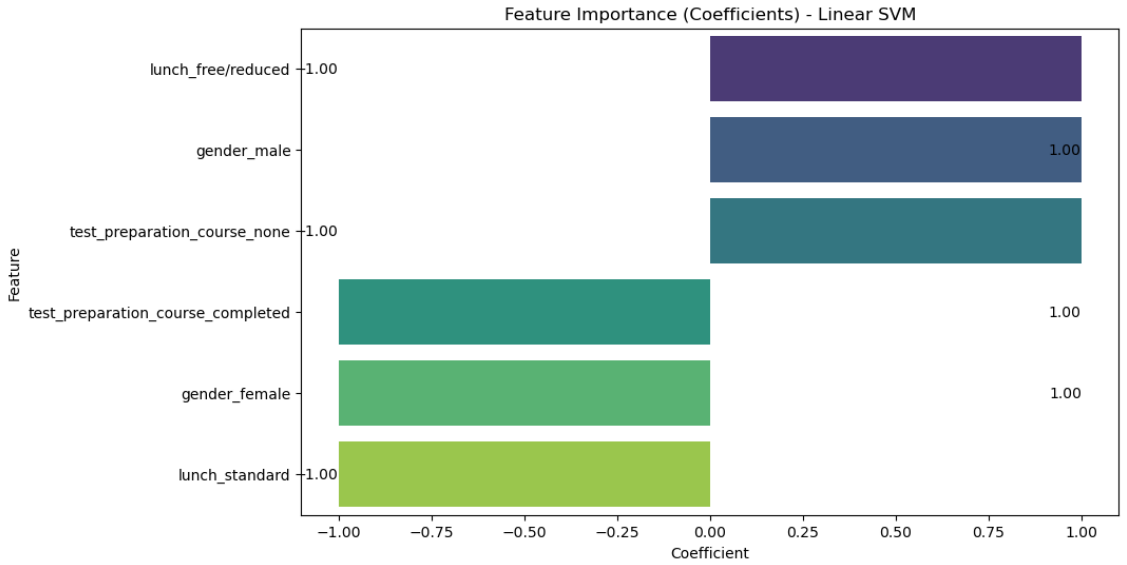
METRICS (SAME AS RF)

- Precision: Precision measures the accuracy of positive predictions made by the model. It is the ratio of correctly predicted positive observations to the total predicted positives, indicating how many of the instances classified as positive are actually positive.
- Recall: It is the ratio of correctly predicted positive observations to the actual positives in the data.
- F-1 score: $$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$
- Support



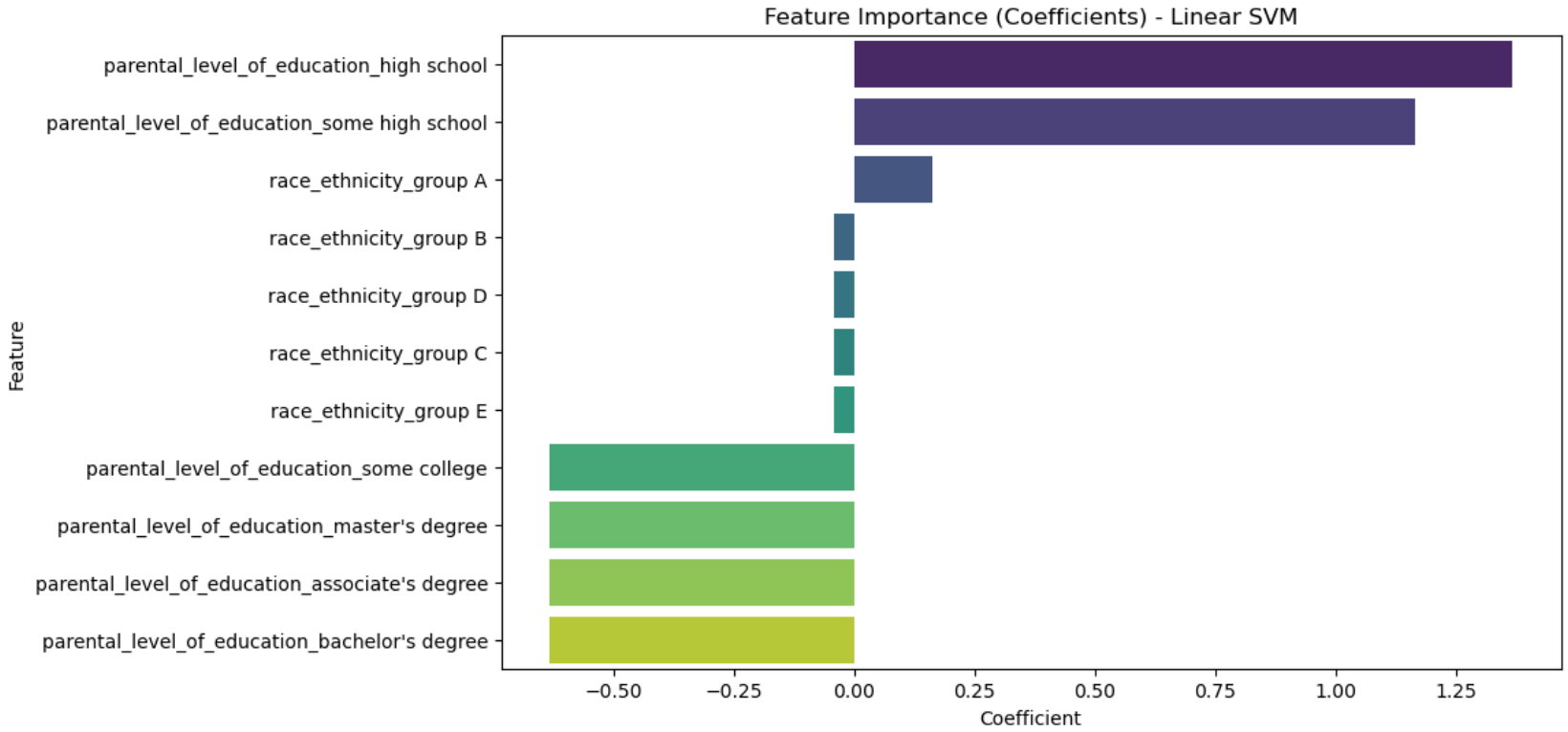

```
Accuracy on Train Dataset: 0.38875
Accuracy on Test Dataset: 0.415

Classification Report on Test Dataset:
                precision    recall  f1-score   support

          Fail       0.48      0.52      0.50        58
 First Division      0.43      0.48      0.46        58
Second Division      0.34      0.45      0.39        55
 Third Division      0.00      0.00      0.00        29

      accuracy                           0.41       200
     macro avg       0.31      0.36      0.34       200
  weighted avg       0.36      0.41      0.38       200


Confusion Matrix on Test Dataset:
[[30  8 20  0]
 [13 28 17  0]
 [ 9 21 25  0]
 [10  8 11  0]]
```

SVM RESULTS




```
Relevance of each variable (sorted by coefficient):
lunch_free/reduced: 1.00
gender_male: 1.00
test_preparation_course_none: 1.00
test_preparation_course_completed: -1.00
gender_female: -1.00
lunch_standard: -1.00
```

SVM FEATURE IMPORTANCE PLOTS



SVM FEATURE IMPORTANCE PLOTS

MODEL SUMMARY

MODEL EVALUATION

Based on the analysis of our model performance, we observed consistent accuracy scores between the training and testing datasets, suggesting presence of slight overfitting.

Among the models evaluated, the Multi-Variate Decision Tree exhibited the highest test accuracy score of 73.61%. Alternative models such as XGBoost are comparable for similar predictor and response variables

Accuracy significantly drops as we increase the number of parameters which underscores the need for additional data refinement and feature engineering.

Model	Accuracy (Max)
Linear Regression	0.17
Decision Tree	0.71
Random Forest	0.47
XGBoost	0.40
Support Vector Machine	0.42

OBJECTIVES ACHIEVED WITH ML

- Perform multivariate analysis using different ML algorithms
- We have compared the influence different socio-economic factors have on the performance of the students
- Evaluate and compare the accuracy of the different models
- Determine most accurate model
- Determine the best combination of predictors to predict student performance.

BACK TO OUR PROBLEM

- The analysis revealed a significant association between socioeconomic factors such as parental education level, lunch status, and test preparation courses, and student academic performance categories.
- By leveraging machine learning techniques such as Decision Trees, Random Forest, XGBoost, and SVM, we were able to develop predictive models to categorize students into performance divisions based on their socioeconomic attributes and academic scores
- From XGboost, it is seen that race is a very important factor to predict scores, but contrary to common belief, parents having masters degree isn't reflected to be very important, contradicting above point of them having higher median.

CONCLUSION

- Level of disparity between the train set and the test set in some algorithms reveals that the models possibly underfitted the data
- The best model was determined to be the decision tree combined with one hot encoding on the variables – "gender", "lunch" and "test preparation course". This is possibly because the high amount of non-linear relationships in the dataset. However, it is worth noting that for similar variables, random forest and XGBoost algorithms also gave satisfactory results
- The worst model was determined to be Logistic Regression. We believe this is because the label encoding model assigns higher values to certain variables randomly which does not reflect the true nature of the dataset.

ANSWER TO PROBLEM

- The most relevant variables to predict the student's academic performance were determined to be parental level of education, the type of lunch consumed and the test preparation course that the student completed.
- Contrary to popular belief, race didn't play a major role in the determination of a students academic performance
- Despite our efforts, we encountered challenges in significantly improving model accuracy beyond a certain threshold. This suggests that additional factors beyond the ones considered in this study may influence academic performance, highlighting the complexity of the issue. Further exploration and refinement of the models, as well as the inclusion of additional relevant variables, may be necessary to enhance predictive accuracy.

WORK DISTRIBUTION

Sanhith: Dataset Selection, Presentation, Slides, Exploratory Data Analysis

Shreyas: Data Cleaning, Machine Learning Models, Presentation, Slides, Analysis

Siddhant: Script Writing

Suraj: Slides, Presentation, Analysis

THANK YOU

With special thanks to our T.A. Mr Runzhong
for guiding us through this project