Project Documentation

Part1: College Distance Dataset Analysis

Overview

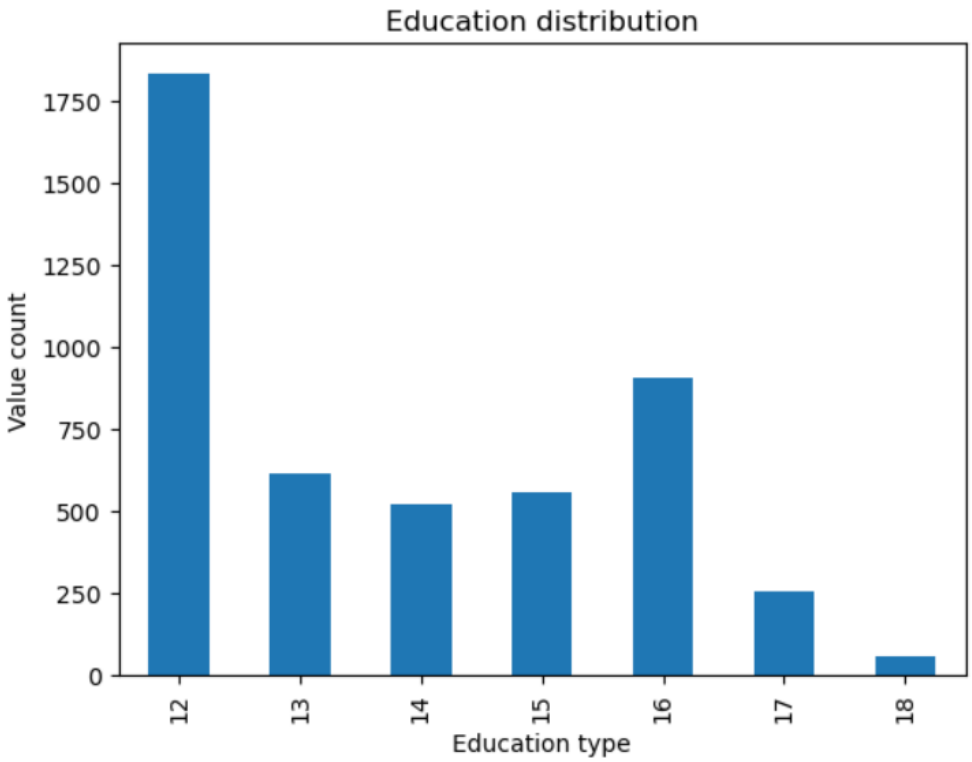
This script provides a data exploration, transformation, and encoding workflow for the College Distance dataset. The primary goals include visualizing data distributions, analyzing key statistics, handling categorical features, and saving a cleaned dataset and the encoder for future use.

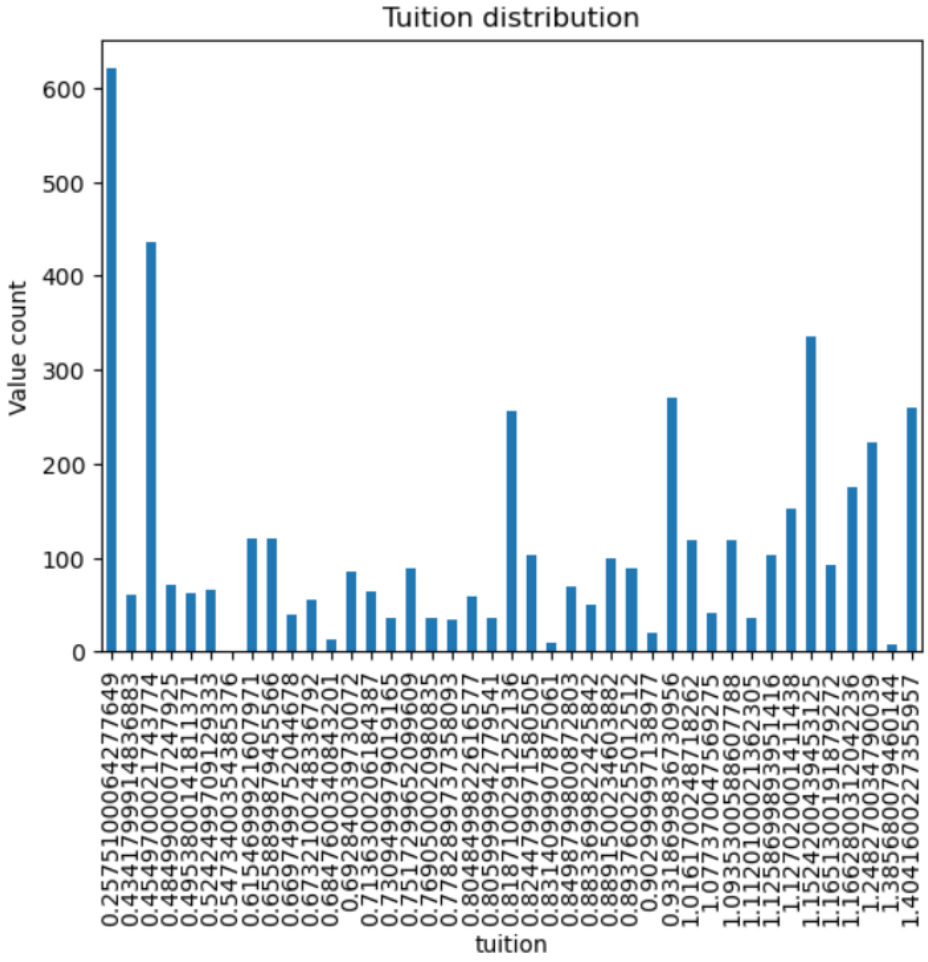
1. Initial Exploration

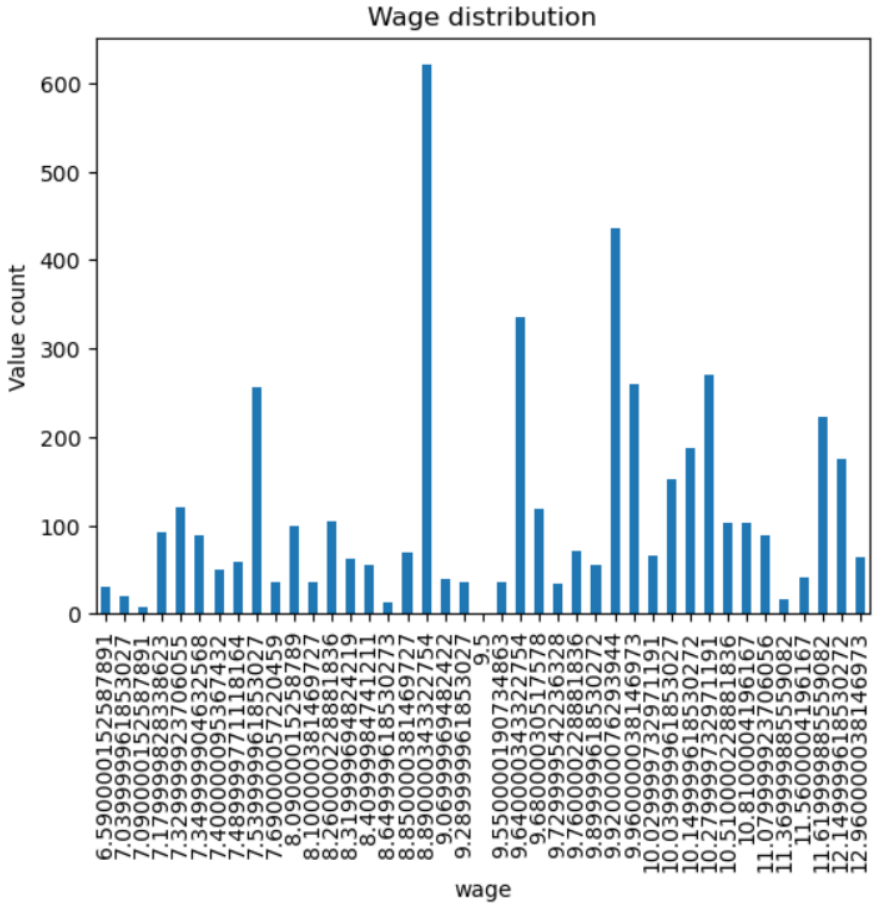
* Display the first few rows and statistical summary.
* Check for null values and data shape.
* Analyze unique values in each column.

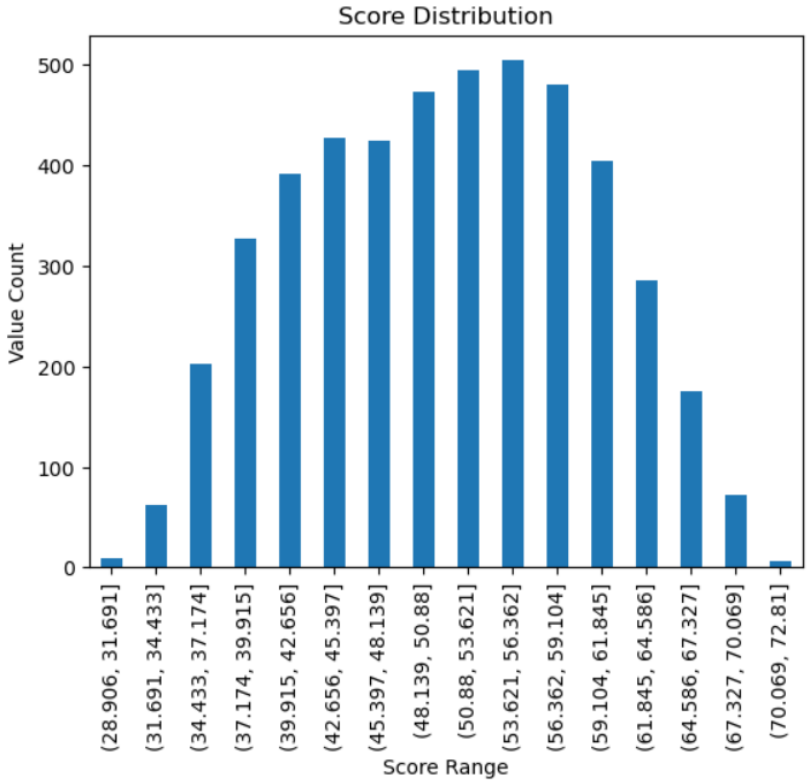
2. Distribution Visualizations

* Education Distribution: Bar plot of `education` levels.
* Tuition Distribution: Bar plot of `tuition` values.
* Tuition Distribution: Bar plot of `tuition` values.
* Wage Distribution: Bar plot of `wage` values.
* Score Distribution: Histogram of scores binned into ranges.




Tuition distribution
Value count
0
100
200
300
400
500
600
0.2575100064277649
0.4341799914836883
0.4549700021743774
0.4849900007247925
0.4953800141811371
0.5242499709129333
0.5473400354385376
0.6154699921607971
0.6558899875204678
0.6697499752044678
0.6732100248336792
0.6847600340843201
0.6928400039730072
0.7136300206184387
0.7309499979019165
0.7517299965209960
0.7690500020980835
0.7782899737358093
0.8048499982261657
0.8059999994277954
0.8187100029125213
0.8244799997158050
0.8314099990787506
0.8498799980087280
0.8833699982242584
0.8891500025501251
0.8937600025501251
0.9029999997138977
0.9318699983673095
1.0161700024871826
1.0773700047569275
1.0935300588607788
1.1120100002136230
1.1258699893951416
1.1270200001411438
1.1524200439453125
1.1651300191879272
1.1662800312042236
1.2482700347900390
1.3856800079460144
1.4041600022735595
tuition


Wage distribution
Value count
0
100
200
300
400
500
600
6.590000152587891
7.039999961853027
7.090000152587891
7.179999828338623
7.329999923706055
7.349999904632568
7.400000095367432
7.489999771118164
7.539999961853027
7.690000057220459
8.090000152587891
8.100000381469727
8.260000228881836
8.319999694824219
8.409999847412110
8.649999618530273
8.850000381469727
8.890000343322754
9.069999694824220
9.289999961853027
9.5
9.550000190734863
9.640000343322754
9.680000305175781
9.729999542236328
9.760000228881836
9.899999618530272
9.920000076293945
9.960000038146973
10.029999732971191
10.039999961853027
10.149999618530272
10.279999732971191
10.510000228881836
10.810000419616700
11.079999923706056
11.369999885559082
11.560000419616700
11.619999885559082
12.149999618530272
12.960000038146973
wage

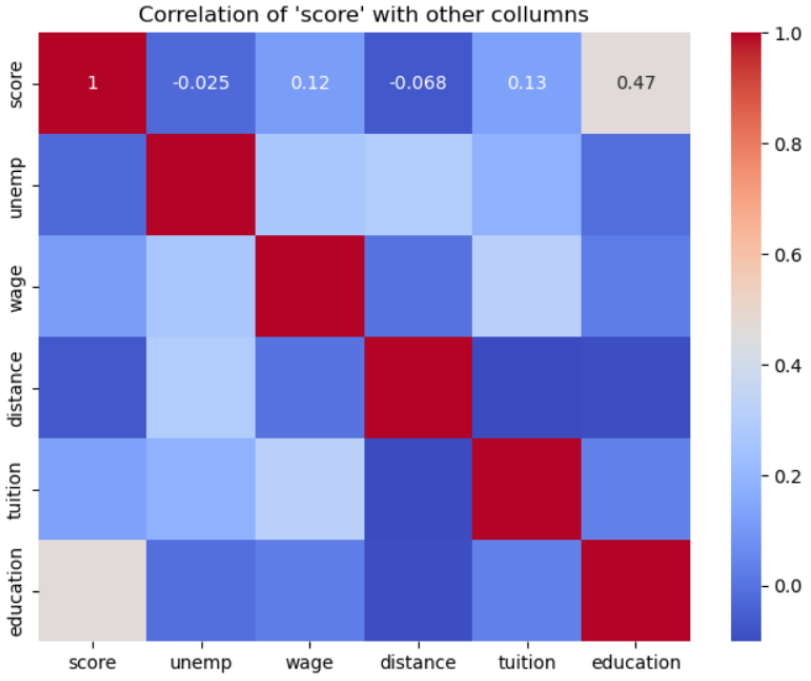


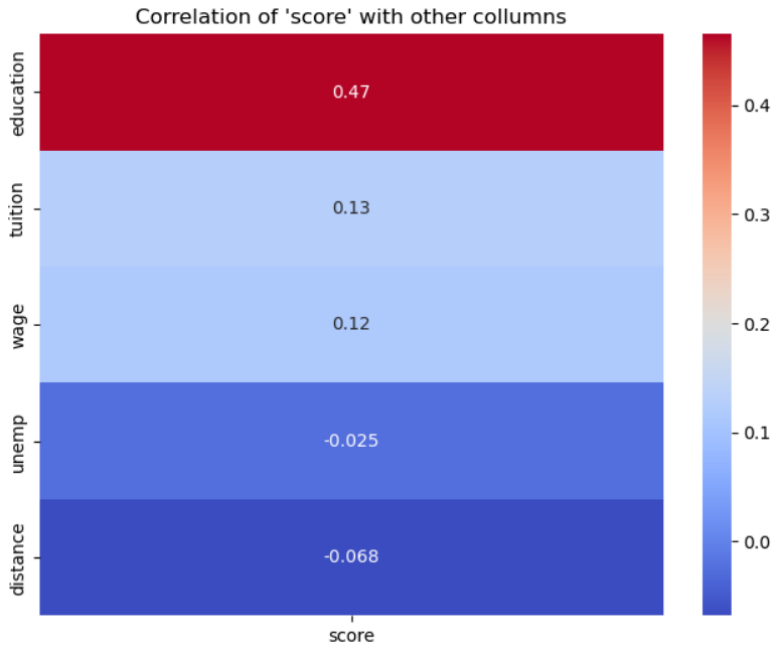
3. Statistical Analysis

* Compute skewness and kurtosis for numeric columns:

```
score: Skewness = -0.03, Kurtosis = -0.88
unemp: Skewness = 1.56, Kurtosis = 5.41
wage: Skewness = 0.09, Kurtosis = -0.26
distance: Skewness = 3.00, Kurtosis = 13.04
tuition: Skewness = -0.15, Kurtosis = -1.05
education: Skewness = 0.44, Kurtosis = -1.23
```

* Identify correlations between `score` and other columns and display correlation matrices:





4. Categorical Encoding

* Identify and encode categorical columns using OrdinalEncoder.
* Convert encoded columns to integer type and create a dictionary for encoding mappings:

{'gender': {'female': 0, 'male': 1},
'ethnicity': {'afam': 0, 'hispanic': 1, 'other': 2},
'fcollege': {'no': 0, 'yes': 1},
'mcollege': {'no': 0, 'yes': 1},
'home': {'no': 0, 'yes': 1},
'urban': {'no': 0, 'yes': 1},
'income': {'high': 0, 'low': 1},
'region': {'other': 0, 'west': 1}}

5. Saving Results

* Save the cleaned dataset as ‘CollegeDistanceCleaned.csv’.
* Save the encoder as ‘ordinal_encoder.pkl’ for reuse.

Part1: College Model Training and Evaluationt

Overview

This script builds, trains, and evaluates several regression models to predict the `score` variable in the College Distance dataset. The models explored include Decision Tree, Random Forest, Linear Regression, and Gradient Boosting Regressor, with a focus on optimizing model performance using evaluation metrics and hyperparameter tuning.

1. Data Import and

* Features (`X`) are defined by excluding the `score` column, while the target variable (`y`) is set to `score`
* Split data into training and testing sets with an 80-20 ratio.

2. Model Training and Evaluation

Decision Tree Regressor:
* Model 1: Default settings with `squared_error` criterion.
* Model 2 & 3: Modified models using `absolute_error` criterion with `max_depth` variations.
* Best scores:
    MAE: 6.1114398879843925
    MSE: 55.872398998729544
    r2 score: 0.26978943190142146

Random Forest Regressor:
* Model 1 & 2: Tested with `n_estimators` (100 and 150), different criterion values, and depth restrictions.
* Model 3: Lower `max_depth` to 4 for testing overfitting vs. underfitting.
* Best scores:
    MAE: 6.00735368290028
    MSE: 54.13116539506546
    r2 score: 0.29254605595389505

Linear Regression:
* Trained on all features as a baseline linear model.
* Best Scores:
    MAE: 5.999723186927585
    MSE: 53.029970954561485
    r2 score: 0.306937845681873

Gradient Boosting Regressor:
* Initial Models: Tested with various `n_estimators` and `learning_rate` values to assess boosting effect.
* Modified Feature Set: Dropped columns `ethnicity`, `mcollege`, `fcollege`, `urban`, and `home` and trained the model to assess feature importance.
* Custom Function: Defined `within_10_percent_accuracy` to evaluate accuracy within ±10% tolerance of actual values.
* Best scores:
    MAE: 5.642095694628343
    MSE: 48.19469718805719
    r2 score: 0.35878482763328656
    Accuracy within +-10%: 51.0548523206751%

3. Hyperparameter Tuning with RandomizedSearchCV
* Used a parameter grid on Gradient Boosting Regressor (`RandomizedSearchCV`) to identify the best model parameters based on negative MSE score.
* Output: Displayed best parameters, MAE, MSE, R^2 score, and accuracy within ±10% of the target variable.
* The search did not beat:
    MAE: 5.642095694628343
    MSE: 48.19469718805719
    r2 score: 0.35878482763328656
    Accuracy within +-10%: 51.0548523206751%

Key Results
* Decision Tree: Performed well with minimal tuning but sensitive to depth settings.
* Random Forest: Higher accuracy with more trees, though computationally intensive with larger depths.

* Linear Regression: Baseline model provided reference results with limited feature interactions.
* Gradient Boosting: Effective in improving accuracy, especially after tuning and dropping less influential features.

Best Overall scores:

MAE: 5.642095694628343
MSE: 48.19469718805719
r2 score: 0.35878482763328656
Accuracy within +-10%: 51.0548523206751%