Grade Classification using Random Forest Classifier

Debojjal Bagchi, Barnopriyo Dutta and Tsungrojungla Walling

About the Dataset & Problem definition

- https://www.kaggle.com/datasets/whenamancodes/student-performance
- Number of datapoints: 397

school	sex	age
address	famsize	Pstatus
Medu	Fedu	Mjob
Fjob	reason	guardian
traveltime	studytime	failures
schoolsup	famsup	paid
activities	nursery	higher
internet	romantic	famrel
freetime	goout	Dalc
Walc	health	absences

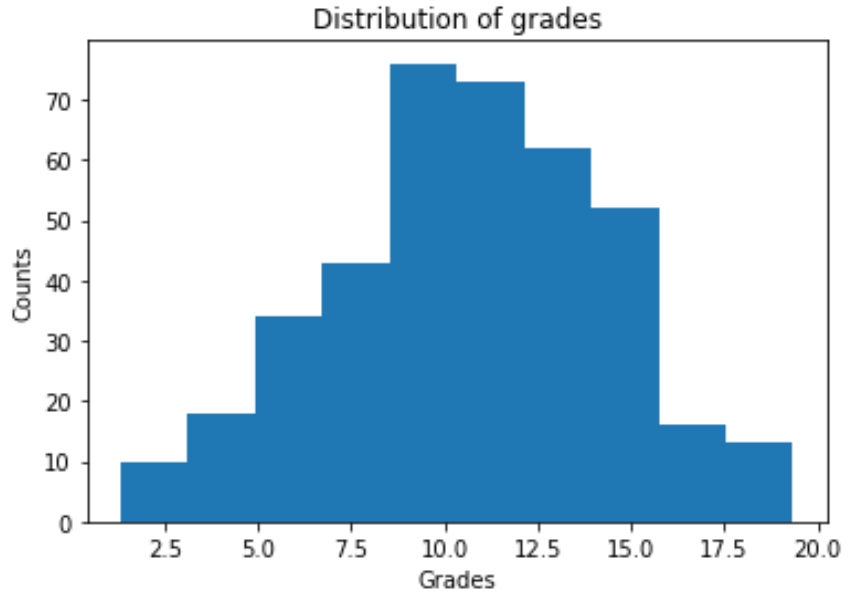
Methodology

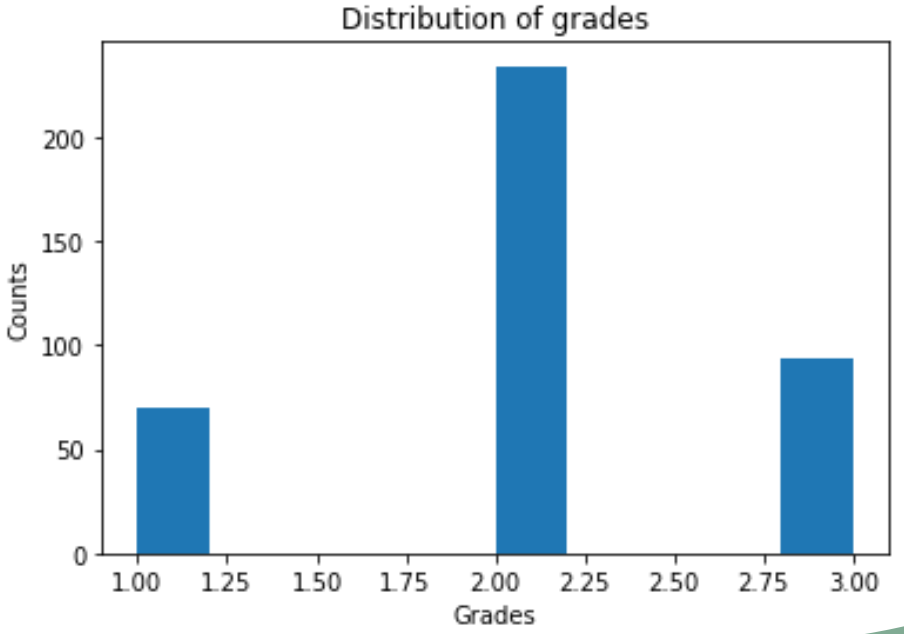
- Data engineering and feature creation
- One-Hot encoding of categorical features, NaN values were checked (None)
- Lasso regression to select features using tenfold cross validation (StandardScaling done beforehand)
- Remove features with zero coefficient in Lasso regression
- Remove features which are greatly correlated (>0.9)
- Bin the grades into categories – low, medium and high
- Model building – split into train and test data (80:20)
- Fit with the data using Random Forest Classifier with default hyperparameters
- Use GridSearchCV for hyperparameter tuning – tenfold cross validation
- Finally see the classification report, training accuracy and testing accuracy

Data Engineering

- Following new features were created:
- Relative study time = Study time/travel time
- Relative free time = study time/travel time
- Relative absences = absences/study time
- The travel time was normalized based on the address (Urban / Local)
- The school name column was dropped
- Grades were binned in three categories : Low(1), Medium(2) and High(3)

```
(7.3, 13.367]      234
(13.367, 19.433]    93
(1.233, 7.3]        70
```





Feature Selection using LASSO

```
clf.coef_
```

```
array([-0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -1.26851584e-01,
        0.00000000e+00, -1.24759518e+00, -0.00000000e+00,  0.00000000e+00,
       -4.78006044e-01,  0.00000000e+00,  0.00000000e+00, -1.65967086e-01,
        0.00000000e+00,  1.03977482e-01,  1.57837094e-01,  0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00,  3.07244217e-01, -9.23449043e-16,
       -2.64604647e-01,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  1.01654192e-01, -2.16592156e-01,  3.24705770e-01,
       -0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -6.25056385e-02,
       -0.00000000e+00,  1.58854503e-01, -6.95193166e-02,  0.00000000e+00,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  1.84061877e-01, -1.02605449e-16, -4.46908555e-03,
        7.69540869e-17, -0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00, -2.59038151e-01,  3.59119072e-16, -8.86995288e-02,
        0.00000000e+00,  8.74190323e-02, -5.13027246e-17])
```

- Lasso performs best when all numerical features are centered around 0 and have variance in the same order.
- If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected. [Ref: 3]
- This means it is important to standardize our features.
- To avoid data leakage, the standardization of numerical features is performed after data splitting and only from training data.
- We obtain all necessary statistics for our features (mean and standard deviation) from training data and also use them on test data.

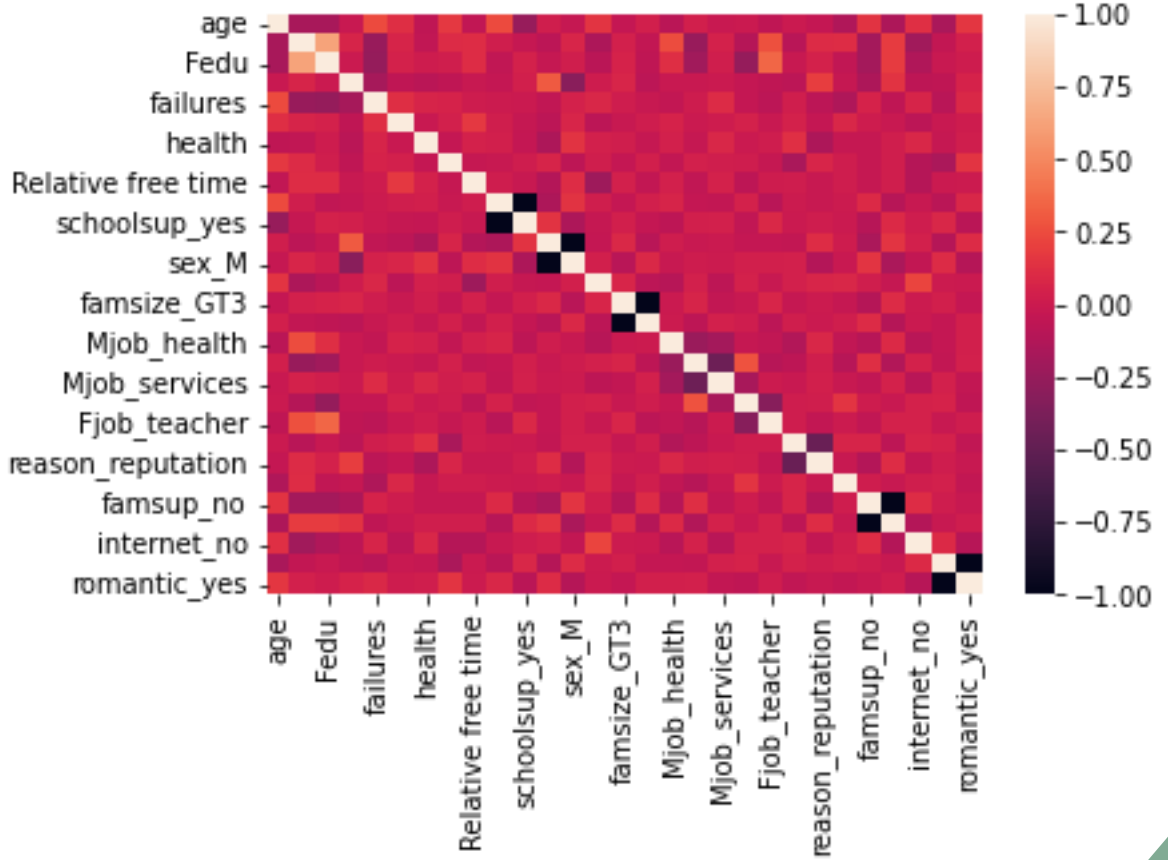
Results:
R squared training set 30.36
R squared test set 13.8
Alpha: 0.17552820323250823

Feature Selection using LASSO

school	sex	age
address	famsize	Pstatus
Medu	Fedu	Mjob
Fjob	reason	guardian
traveltime	studytime	failures
schoolsup	famsup	paid
activities	nursery	higher
internet	romantic	famrel
freetime	goout	Dalc
school	age	sex
Walc	health	absences

Feature selection using correlation

- For The features that had 0.95 or more correlation, only one such feature was kept. The rest were dropped
- Absolute values of correlation were used
- Dropped features:
- ['schoolsup_yes', 'famsup_yes', 'paid_yes', 'higher_yes', 'romantic_yes']



Hyperparameter Tuning

- GridSearchCV was used for finding optimal hyperparameters
- max_depth – longest path between the root node and the leaf node, maximum number of levels
- max_features – number of features for splitting at each leaf node
- n_estimators - number of trees in the forest
- `forest_params = [{'max_depth':[5,10,20,30], 'max_features':[10,15,20,23], 'n_estimators' : [100,200,400,800]}]`
- 10-fold cross-validation with accuracy scoring
- Thi64 s corresponds to 64 fits * 10cv = 640 fits
- Best Parameters were: {'max_depth': 5, 'max_features': 10, 'n_estimators': 100}
- The corresponding score was: 0.678629032258064 6
- Finally, the model was fitted with these hyperparameters and accuracy, etc was calculates

Results

- Model accuracy score: 0.6250
- Training-set accuracy score: 0.7476.
- Training accuracy was 0.96 without hyperparameter tuning, and testing accuracy was 0.5 => Severe overfitting
- After hyperparameter tuning :
- Training and testing accuracy are similar: Slight overfitting is seen
- Training: 0.75
- Testing: 0.62

Before GridSearchCV

```
Training set score: 0.9968
Test set score: 0.5375
Model accuracy score: 0.5375
Training-set accuracy score: 0.9968
```

	precision	recall	f1-score	support
1.0	0.67	0.29	0.40	14
2.0	0.64	0.78	0.71	46
3.0	0.56	0.50	0.53	20
accuracy			0.62	80
macro avg	0.62	0.52	0.54	80
weighted avg	0.63	0.62	0.61	80

```
Training set score: 0.7476
```

```
Fitting 10 folds for each of 48 candidates, totalling 480 fits
{'max_depth': 5, 'max_features': 10, 'n_estimators': 100}
0.6214717741935484
```

After GridSearchCV

Contributions

- Barnopriyo – feature engineering, coding Lasso, correlation, Analysis and interpretation of results
- Debojjal – data pre-processing, coding RandomForest and Lasso, Analysis and interpretation of results, study conception and design, ppt preparation
- Tsungrojungla – GridSearchCV, powerpoint preparation, Analysis and interpretation of results

References

- https://www.section.io/engineering-education/hyperparmeter-tuning/
- https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
- https://www.kirenz.com/post/2019-08-12-python-lasso-regression-auto/
- https://www.geeksforgeeks.org/random-forest-regression-in-python/

Thank You