Machine Learning Engineer Nanodegree

Capstone Proposal

Conor Donohue May 21st, 2019

Proposal

Project Overview

This project attempts to tackle and investigate the social and economic aspects of a students' life and see what correlation they have to their alcohol consumption and to their school grades. This problem will be tackled using a myriad of regression algorithms to predict continuous values for the number of classes they fail. The data will be taken from Kaggle and will be loaded/analyzed using tools such as Python/SkLearn/Juypter Notebooks. The hope of this investigation is to show how there is a correlation between students' grades and the amount of alcohol they consume. If this can be shown it might hopefully enlighten students to the adverse effects of alcohol on not only their body but their academic life as well. Using machine learning algorithms to help the lives a people is not a new idea. One application that can be seen of where this is implemented is the use of decision trees to detect cyber bullying [3].

Problem Statement

This study will be looking at the alcohol consumption of two separate classes. A deep dive will be undertaken to investigate the social and economic aspects of a Math's and Portuguese classes and how all of these can have an impact on how well a student might perform. Relationships between all of these can be looked at and will be determined which aspects of a student's life is relevant in predicting the outcome of a student's failure rate. This investigation will use regression techniques to attempt to predict numeric values for the final grade a student is expected to receive.

Metrics

This project will evaluate the scores of the algorithms using techniques such as RMSE. Optimization will be attempted through the use of cross-validation along with other techniques.

This is a popular open source dataset where numerous data scientists have come along and tried to solve this problem. One benchmark that could be used is found on Kaggle [2] (Kaggle, 2019). This notebook used techniques such as Linear Regression, Decision Trees and Lasso. The results of this could then be compared to the future results of this investigation.

Datasets Exploration

The project will be taking an open source csv dataset from Kaggle [1]. The dataset contains two csv files. Both files contain the same headers but the difference between them is that one csv file is for a Maths class while the other is for a Portuguese class. A more thorough description of the dataset can be seen below

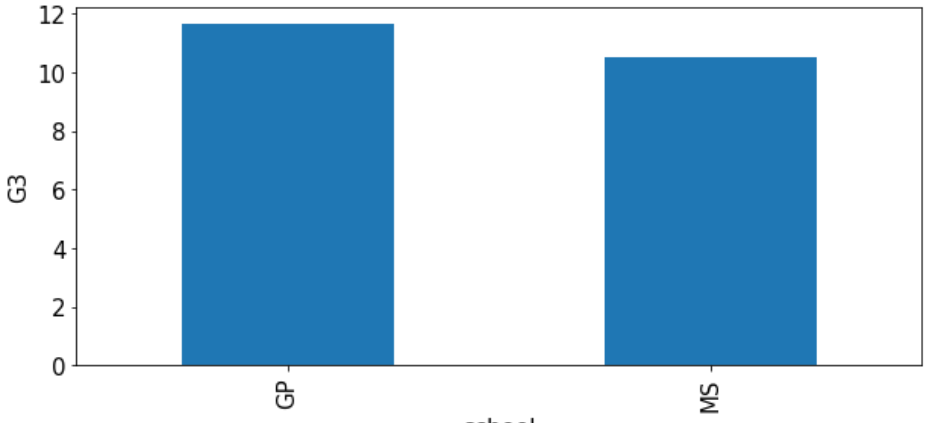
1. School: Student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. Sex: Student's sex (binary: 'F' - female or 'M' - male)
3. Age: Student's age (numeric: from 15 to 22)
4. Address: Student's home address type (binary: 'U' - urban or 'R' - rural)
5. Famsize: Family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus: Parent's cohabitation status (binary: 'T' - living together or 'A' - living apart)
7. Medu: Mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education, or 4 - higher education)
8. Fedu: Father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education, or 4 - higher education)
9. Mjob: Mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob: Father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. Reason: Reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12. Guardian: Student's guardian (nominal: 'mother', 'father' or 'other')
13. Traveltime: Home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. Studytime: Weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. Failures: Number of past class failures (numeric: n if 1<=n<3, else 4)
16. Schoolsup: Extra educational support (binary: yes or no)
17. Famsup: Family educational support (binary: yes or no)
18. Paid: Extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. Activities: Extra-curricular activities (binary: yes or no)
20. Nursery: Attended nursery school (binary: yes or no)
21. Higher: Wants to take higher education (binary: yes or no)
22. Internet: Internet access at home (binary: yes or no)
23. Romantic: With a romantic relationship (binary: yes or no)
24. Famrel: Quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25. Freetime: Free time after school (numeric: from 1 - very low to 5 - very high)
26. Gout: Going out with friends (numeric: from 1 - very low to 5 - very high)

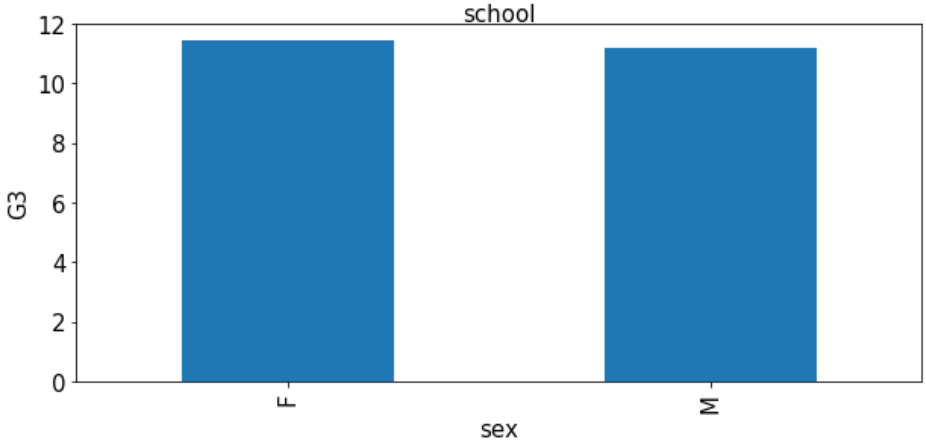
27. Dalc: Workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28. Walc: Weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29. Health: Current health status (numeric: from 1 - very bad to 5 - very good)
30. Absences: Number of school absences (numeric: from 0 to 93)
31. G1: First period grade (numeric: from 0 to 20)
32. G2: Second period grade (numeric: from 0 to 20)
33. G3: Final grade (numeric: from 0 to 20, output target)

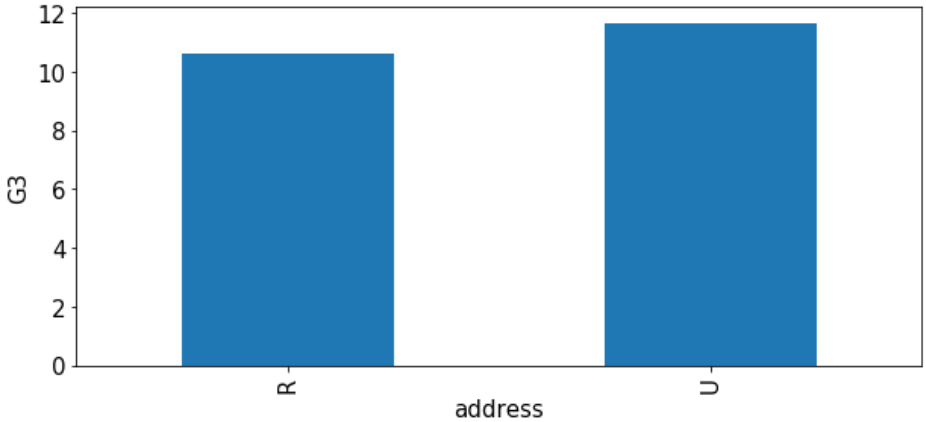
There are 395 datapoints for the Maths class and 649 data points for the Portuguese class. For splitting the data into their relevant group, train/test/validation, SkLearn's train_test_split method can be used to implement this for us.

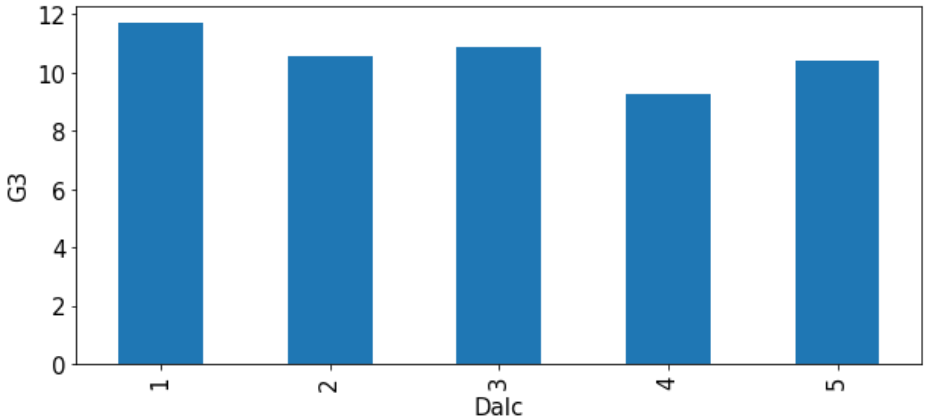
Exploratory Visualization

Below we can look at different attributes and let us have an insight into how they might affect the final grade. For example, we can see that there is little difference in final grades between males and females. However, we can see that the GP school seems to have a better average score than MS. One other tool which might give us a better insight would be a heat map.

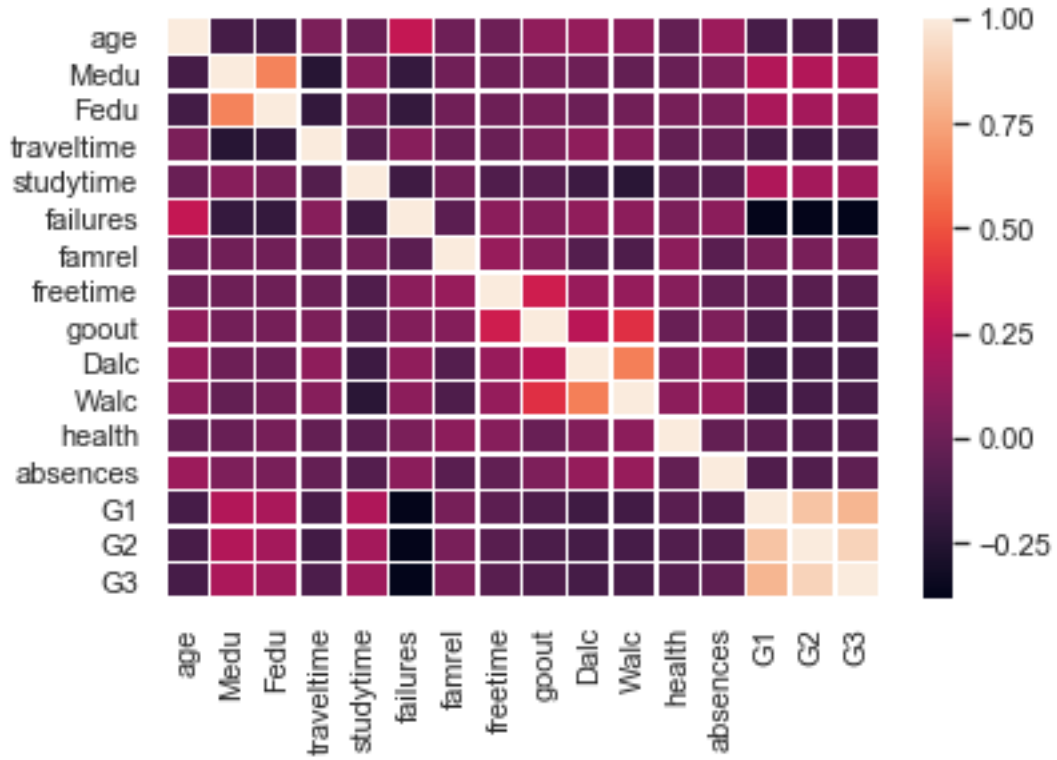

G3
0
2
4
6
8
10
12
GP
MS
school


G3
0
2
4
6
8
10
12
F
M
sex

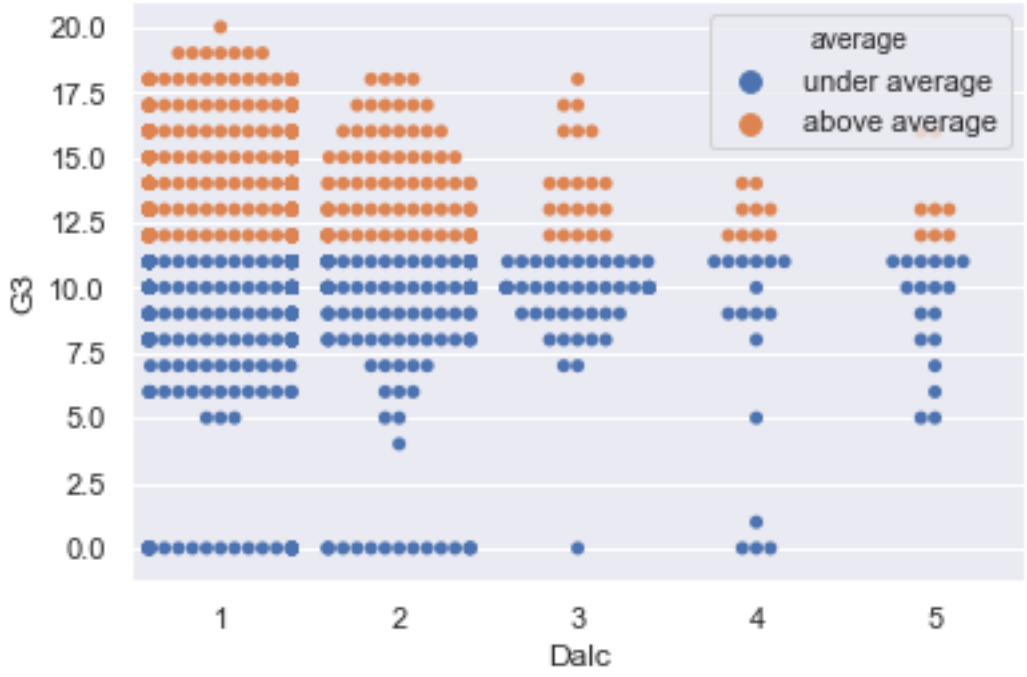

G3
0
2
4
6
8
10
12
R
U
address

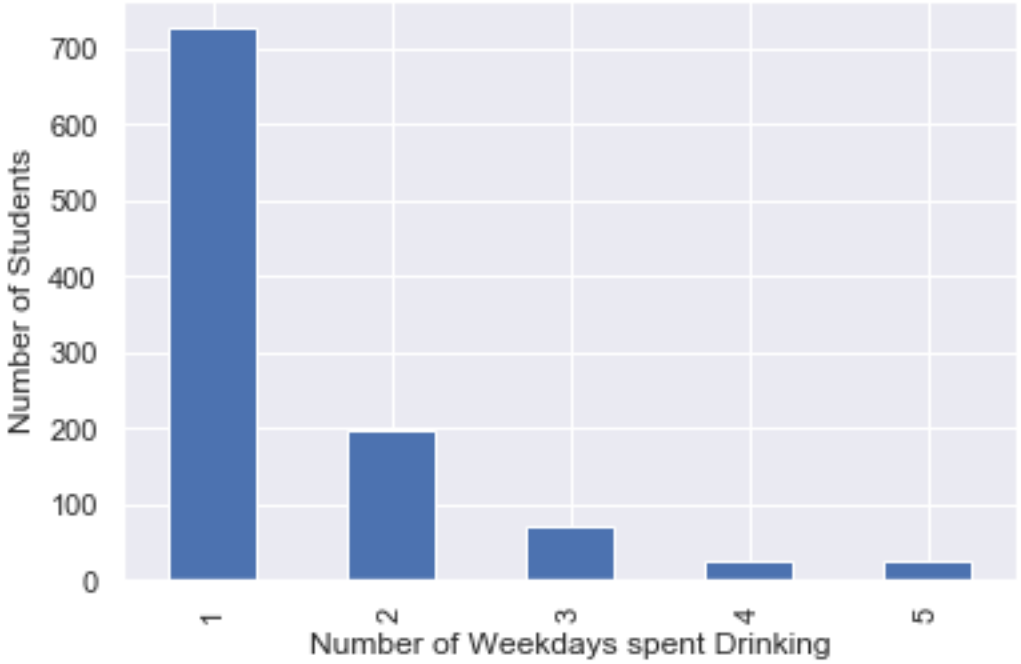

G3
0
2
4
6
8
10
12
1
2
3
4
5
Dalc

The heat map shows the correlation between each feature in the dataset. It can be seen how there are extremely high correlations between the final grades and the grades given in first and second period. The next features which seem to be most correlated are the level of education which the student's parents have attained.



We can see the number of students that drink on a weekday drops dramatically after 1. This is a good sign as it can be seen from the scatter plot that the number of below average students seem to outweigh the number of above average as the student drinks more and more during the week.





Algorithms and Techniques

The aim of this project is to use regression algorithms from the SKLearn package in order to predict the final grade a student will receive. This investigation will also hope to look at the correlation between the different features in the dataset and look at and visualize the relationship between them and the students' grades.

There was some data pre-processing that was required. SkLearn's preprocessing library was used for this. In the preprocessing stage the data was be checked to see if it contains a skewed distribution. A logarithmic transformation needed to be applied to certain features. Normalizing the numerical features of the dataset so that each feature is treated equally when applying the supervised learners was another technique which was done.
Both classes needed to be split into testing and training data and cross validation was used to help reduce the chances of overfitting the data with the models that were created. Grid search was implemented to help decide the most optimal hyper parameters.
Linear Regression will be used as the base model to see how well performance can be predicted. However. the aim of this project will be test 3 other algorithms to see if performance can be improved upon without overfitting the dataset. The algorithms to be tested were

1. Linear Regression
2. XGBoost
3. Support Vector Machines
4. LightGBM

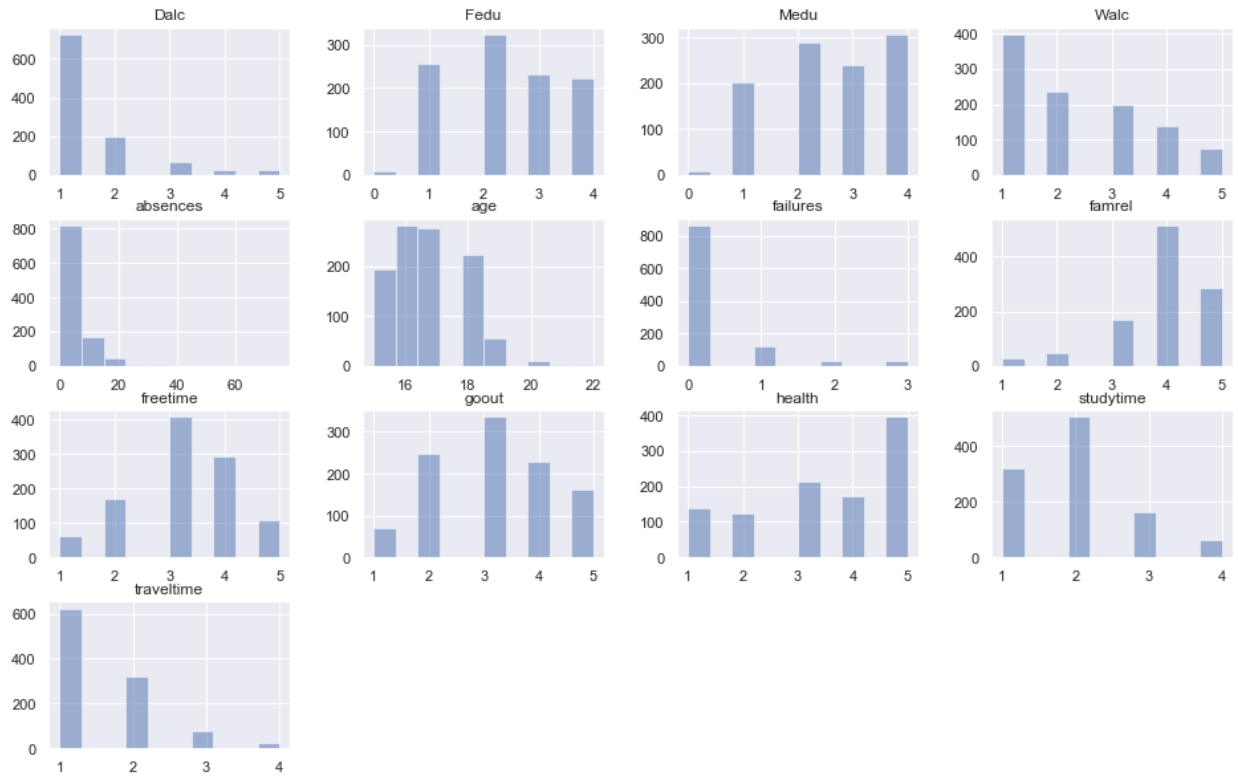
Benchmark Model

For this problem, Linear Regression will be the benchmark model and I will attempt to outperform it with other regression techniques mentioned above. Each algorithm will be compared based on their RMSE and MAE values.

Data Preprocessing

We will look at the dataset and investigate some preprocessing techniques and see which ones will be most appropriate for the data.

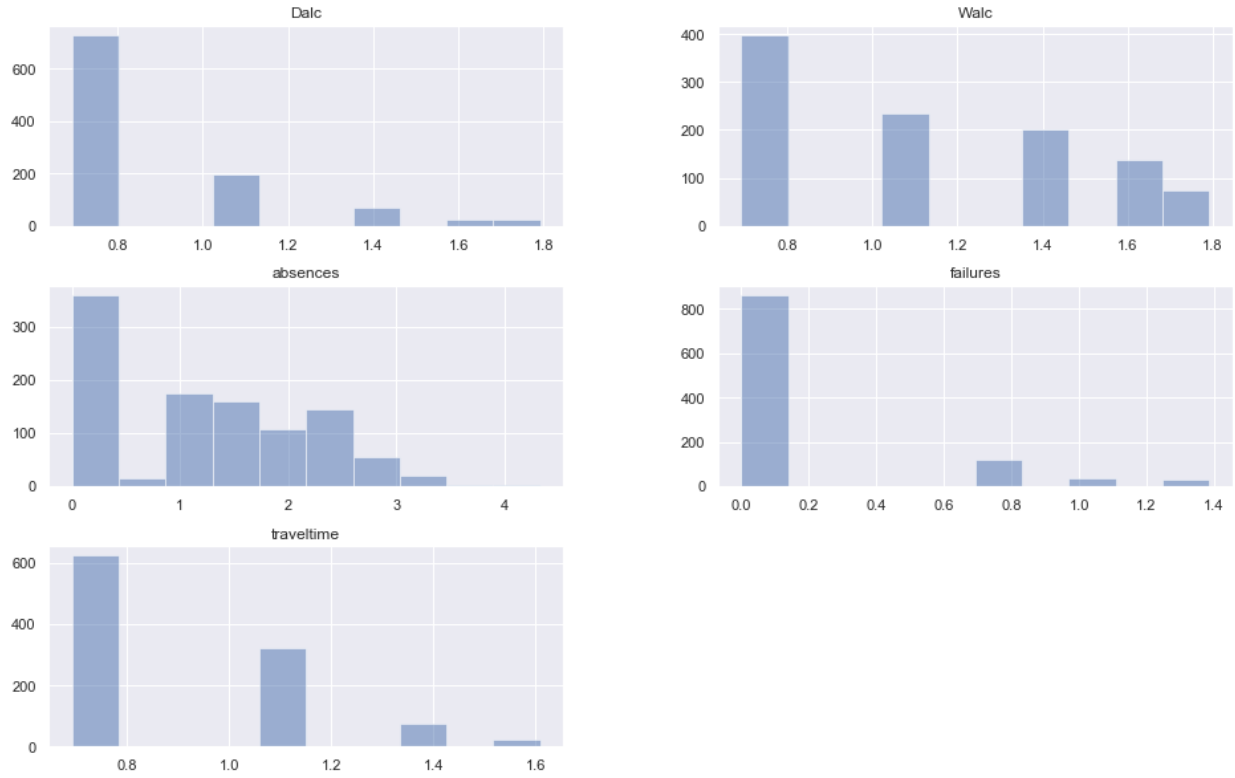
First, we will separate the feature we are trying to predict, G3, with the rest of the features. In order to make this a bit more challenging for the algorithms we will also drop G2 and G1. It will be interesting to see if final grades can be predicted without these intermediate grades.



We can see there are a few features which seem to have a skewed distribution. For example

1. Dalc
2. absences
3. failures
4. traveltime

These are some good choices which could use a logarithimic distribution applied to them. We can see their new distributions below.



It is also good practice to perform some scaling on numerical features. This will ensure that each feature is treated equally when performing supervised learning algorithms on the data. Even though most of the numerical feature were already within a range of 1-5 it was good to scale all features so they can be on the same range.

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	internet	romantic	famrel	freetime	goout	Dalc	Walc	he
0	GP	F	0.428571	U	GT3	A	1.00	1.00	at_home	teacher	...	no	no	0.75	0.50	0.75	0.00000	0.00000	
1	GP	F	0.285714	U	GT3	T	0.25	0.25	at_home	other	...	yes	no	1.00	0.50	0.50	0.00000	0.00000	
2	GP	F	0.000000	U	LE3	T	0.25	0.25	at_home	other	...	yes	no	0.75	0.50	0.25	0.36907	0.63093	
3	GP	F	0.000000	U	GT3	T	1.00	0.50	health	services	...	yes	yes	0.50	0.25	0.25	0.00000	0.00000	
4	GP	F	0.142857	U	GT3	T	0.75	0.75	other	other	...	no	no	0.75	0.50	0.25	0.00000	0.36907	

The final step of the data preprocessing stage was to perform one-hot encoding on the non-numeric features in the data set. Pandas was used to convert these to numerical values to make it easier for the algorithms to process the data.

Implementation

Finally, we can start getting to the good stuff We obviously can't train the model on the whole dataset so we will use SkLearn's cross validation implementation to split the data into training and testing sets. This gave us a training set of 835 samples and a testing set of 209 samples. The model was then fitted and the root mean squared error and mean absolute values were calculated. The results were as follows

1. Root Mean Squared Error for the training set was 2.429

2. RMSE for testing set was 2.7
3. Mean Absolute Error for the testing set was 1.8097
4. MAE for the testing set was 1.93

Since the code for testing each model was the exact same (except for the model itself) a function was made which would allow you to pass in the model and data set and it would print out the accuracy metrics. This was used to calculate the results for the other 3 Regression Algorithms. The accuracy metrics were imported from *sklearn.metrics* while Linear Regression and Support Vector Machines were taken from the SkLearn Package. The other packages were installed via pip. They were run through the same function to be evaluated

Refinement

There were two techniques used for model refinement/improvement. Cross-validation was used on the dataset to help reduce the possibility of overfitting the models to the training set. Also, the use of grid search was implemented to help find the most optimal hyper parameters for each algorithm. Grid Search, a package within SKLearn, allows hyper parameters of a model to be passed in with the options for each parameter specified. This allows the model to be tested with a user defined list of parameters and then the most optimal model can be used in the final comparison.
The list of parameters used for each model can be seen in the code snippet below.

```
'Linear_Regression':{
          'fit_intercept':[True,False],
           'normalize':[True,False],
           'copy_X':[True,False]
          },
'XGB':{
             'boster':['gbtree'],
             'eta':[0.05,0.1,0.25,0.5,0.8],
             'gamma':[0.05,0.1,0.25,0.5,0.8],
             #'reg_alpha': [0.05,0.1,0.25,0.5,0.8],
             #'reg_lambda': [0.05,0.1,0.25,0.5,0.8],
             'max_depth':[3,6,10],
             'subsample':[0.1,0.25,0.5,0.8]
      },
'SVM':{
          'C': [0.001, 0.01, 0.1, 1, 10],
          'gamma' : [0.001, 0.01, 0.1, 1],
          'kernel':['rbf','linear']
          },
'LGB':{'boosting_type': ['gbdt'],
             'num_leaves': [20,50,80],
             'learning_rate': [0.05,0.1,0.25,0.5,0.8],
             'subsample_for_bin': [10,100,500],
             'min_child_samples': [20,50,100],
```

```
    'reg_alpha': [0.05,0.1,0.25,0.5,0.8],
    'reg_lambda': [0.05,0.1,0.25,0.5,0.8]
}
```

Results

Model Evaluation and Validation

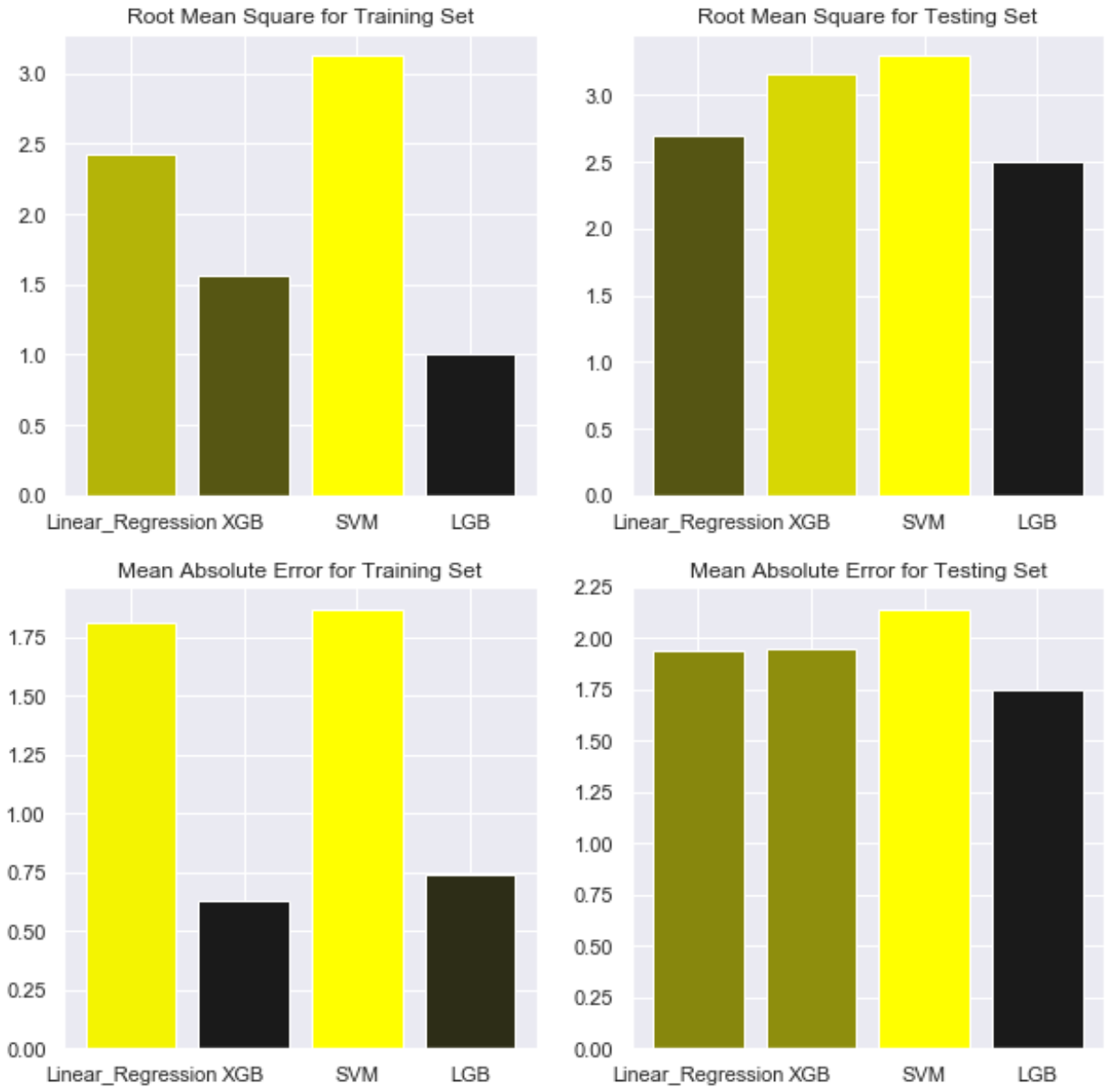
The final model chosen was LightGBM with the parameters of

1. boosting_type='gbdt',
2. class_weight=None,
3. colsample_bytree=1.0,
4. importance_type='split'
5. learning_rate=0.1
6. max_depth=-1
7. min_child_samples=20
8. min_child_weight=0.001
9. min_split_gain=0.0
10. n_estimators=100
11. n_jobs=-1
12. num_leaves=31
13. objective=None
14. random_state=None
15. reg_alpha=0.0
16. reg_lambda=0.0
17. silent=True
18. subsample=1.0
19. subsample_for_bin=200000
20. subsample_freq=0

Most of these parameters were dynamically chosen using GridSearch. This ensured the best values were selected for the parameters that were included in the GridSearch. It was not possible to include all parameters into the GridSearch as the algorithm would have taken hours to complete.

Justification

It can be seen below the scores of each algorithm and how they performed on the testing and training sets



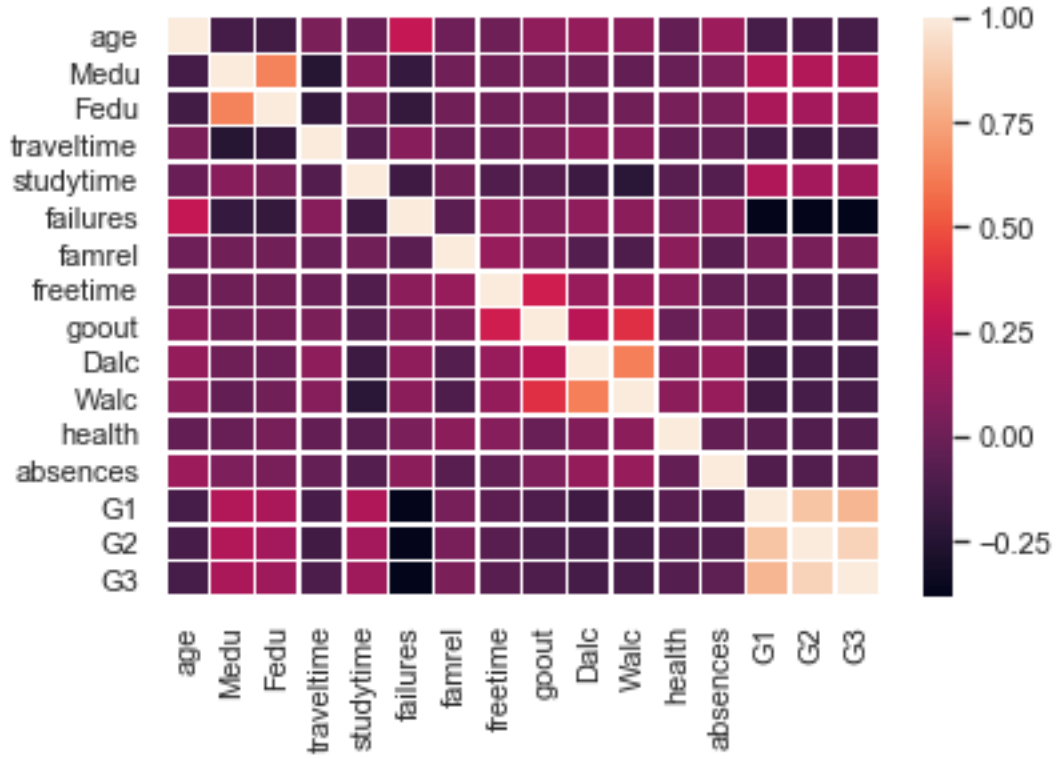
The selected model, LGB, performs in the top 2 for the training sets on both metrics, RMSE and MAE. It is also has the least error in both metrics for the testing set (which is what we really care about). XGB is ruled out as the selected model as it seems to overfit the training set. It can be seen that the training error is very low while the testing set is much higher. SVM is the worst performing algorithm in every case and rules itself out quickly.

Since Linear Regression was our base model and the aim of this was to find an improvement on it LGB is the only logical choice. It outperforms Linear Regression in every category and therefore justifies itself as the selected model.

Conclusion

Free-Form Visualization

Obviously, the graph just above is the most important part of the entire project but for arguments sake and so I am not repeating myself I will discuss another important item that was discovered. Below we can see the heatmap which shows the correlation between each feature in this dataset. G3, the final grade, is the feature we attempted to predict. It can be seen the highest correlated features to G3 are G2 and G1, which were removed from the dataset. Outside of that the next more correlated features are Mother/Father's education (which is interesting). However, these are not that highly correlated to G3. Removing G2 and G1 made this task much more difficult and I think the errors were within 1-2 marks of a students final grade is quite impressive.



Reflection

This project starts with two csv files which contain information on students in a math's class and a Portuguese class. These files contain social and economic insights into these students aswell their grades. The data is loaded into a pandas dataframe and techniques such a normalization, one-hot encoding, gridsearch and cross validation were all used to help prepare the data and optimize four different regression algorithms. These algorithms then were able to predict students' final grades using only their social and economic features.

Being able to write a function which only required the model and the data allowed a great level of flexibility which made testing different algorithms and different parameters much easier and quicker. This was a really enjoyable part of the project.

Improvement

One area of improvement that could be done for this project is to try and optimize the XGB algorithm, the training scores are extremely promising. While it does look like a case of overfitting the training set, it would be interesting to explore the idea of trying to lower the testing score through further data-preprocessing and the refining of parameters further.

Bibliography

[1] *Kaggle*. (2019, May 21). Retrieved from Kaggle: https://www.kaggle.com/uciml/student-alcohol-consumption

[2] Kaggle. (2019, May 21). Retrieved from Kaggle: https://www.kaggle.com/dmitriy19/basic-eda-and-final-grade-prediction

[3] Kelly Reynolds, A. K. (2019, May 21). *IEEE Xplore*. Retrieved from IEEE Xplore: https://ieeexplore.ieee.org/abstract/document/6147681