High School Student Test Score Performance

By: Channing Lee

DH 100 Theory and Methods
Instructor: Adam Anderson

Disclaimers about the Dataset

Data Description:

	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
...	...	...	...	...	...	...	...	...
995	female	group E	master's degree	standard	completed	88	99	95
996	male	group C	high school	free/reduced	none	62	55	55
997	female	group C	high school	free/reduced	completed	59	71	65
998	female	group D	some college	standard	completed	68	78	77
999	female	group D	some college	free/reduced	none	77	86	86

1000 rows × 8 columns

Why did I decide to pick this Dataset?

Questions addressed:

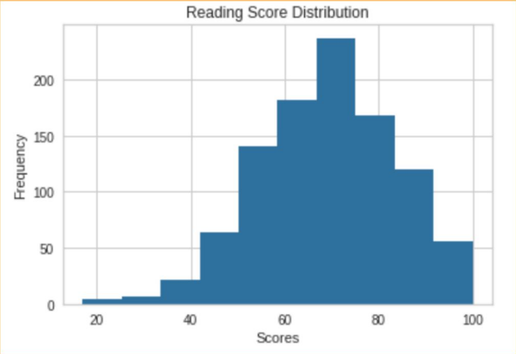
Main Question:

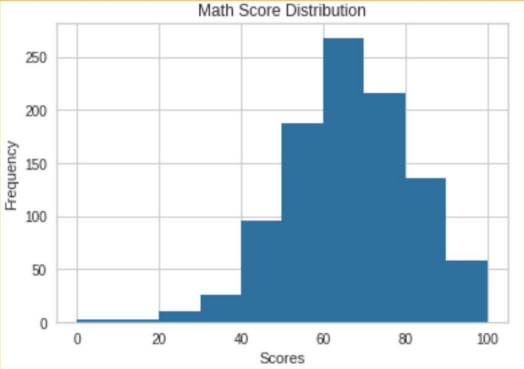
1. How can we increase the scores of the high school students?

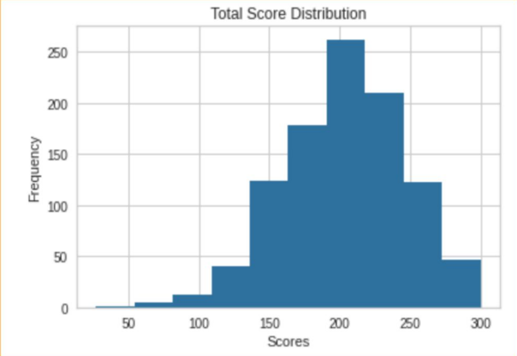
Other Questions:

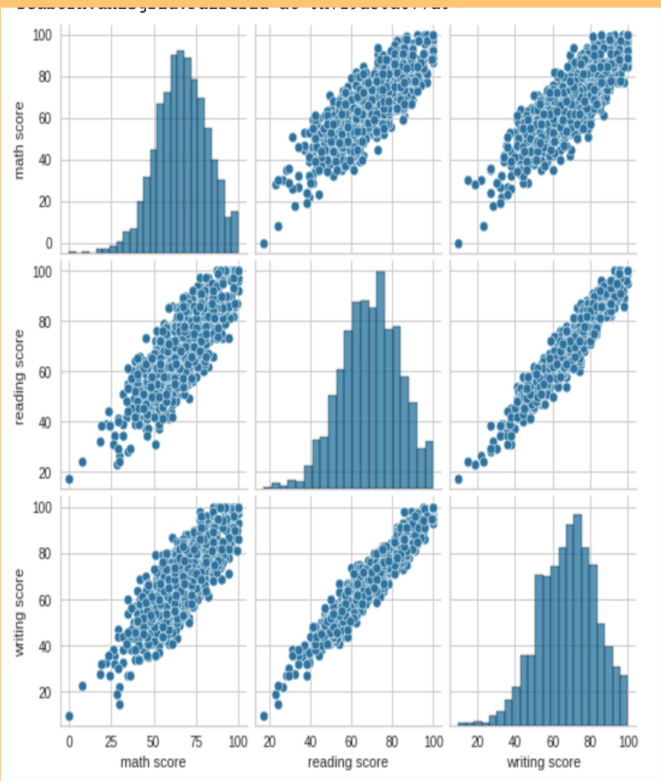
1. How does factors such as gender, race/ethnicity parental level of education, and etc affect student test performance?
2. What factors affect math score the most and the least?
3. What factors affect writing score the most and the least?
4. What factors affect reading score the most and the least?
5. Do the factors that affect each individual score (math, writing, and reading) the most have the same impact on the total score?

Visualizations









Machine Learning Algorithms used

1) Linear Regression
2) Ridge Regression
3) Lasso Regression
4) Random Forest Regression
5) Principal Component Analysis

Setup for Machine Learning

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score	total score	percent correct
0	1	2	5	2	1	72	72	74	218	0.726667
1	1	3	3	2	2	69	90	88	247	0.823333
2	1	2	6	2	1	90	95	93	278	0.926667
3	2	1	4	1	1	47	57	44	148	0.493333
4	2	3	3	2	1	76	78	75	229	0.763333
...	...	...	...	...	...	...	...	...	...	...
995	1	5	6	2	2	88	99	95	282	0.940000
996	2	3	2	1	1	62	55	55	172	0.573333
997	1	3	2	1	2	59	71	65	195	0.650000
998	1	4	3	2	2	68	78	77	223	0.743333
999	1	4	3	1	1	77	86	86	249	0.830000

1000 rows × 10 columns

```
[26] X_train, X_test, Y_train, Y_test = train_test_split(X, Y_math, test_size= 0.33, random_state = 2)
```

How did I compare models?

```
def rmse(actual_y, predicted_y):
    return np.sqrt(np.mean((actual_y - predicted_y)**2))
```

Linear Regression Code

Linear Regression using reading score as the response variable

```
[39] Y_read = dataset["reading score"]

[40] #Split TEST TRAINING

X_train, X_test, Y_train, Y_test = train_test_split(X, Y_read, test_size= 0.33, random_state = 2)

[42] #Running Linear Regression and fitting training data and predicting Y

Linearmodelread = LinearRegression(fit_intercept=False)
Linearmodelread.fit(X_train, Y_train)
Y_pred = Linearmodelread.predict(X_test)

[44] #ROOT MEAN SQUARED ERROR

[45] test_error_linear_read = rmse(Y_test, Linearmodelread.predict(X_test))

[46] #Printing the test error of reading from Linear

[47] print("Test RMSE:", test_error_linear_read)

     Test RMSE: 15.427214220485427

[48] #Displaying coefficient

[49] Linearmodelread.coef_

     array([ 0.29525313,  3.78775894,  3.60156623, 14.53928481, 15.2566129 ])

[50] #Displaying Intercepts

[51] Linearmodelread.intercept_

     0.0
```

Ridge Regression Code

```
[81] alphas = np.arange(0.0001,1,0.001)
     list = alphas.tolist()
```

```
#display alphas
```

```
[83] alphas
```

```
          1.210e-02, 1.310e-02, 1.410e-02, 1.510e-02, 1.610e-02, 1.710e-02,
          1.810e-02, 1.910e-02, 2.010e-02, 2.110e-02, 2.210e-02, 2.310e-02,
          2.410e-02, 2.510e-02, 2.610e-02, 2.710e-02, 2.810e-02, 2.910e-02,
          3.010e-02, 3.110e-02, 3.210e-02, 3.310e-02, 3.410e-02, 3.510e-02,
          3.610e-02, 3.710e-02, 3.810e-02, 3.910e-02, 4.010e-02, 4.110e-02,
```

Ridge Regression Math

```
[84] # Redefining Y_math

[85] Y_math = dataset["math score"]

[86] #Train test split

[87] X_train, X_test, Y_train, Y_test = train_test_split(X, Y_math, test_size= 0.33, random_state = 2)

[88] #Cross validation Ridge regression

[89] clifmath = RidgeCV(alphas = list, normalize = False, store_cv_values= True).fit(X_train, Y_train)
     clifmath.alpha_

     0.9991

[90] #Using optimal alpha in Ridge

     fullclifmath = Ridge(alpha = clifmath.alpha_, normalize= False).fit(X_train, Y_train)

     #displaying math coefficients ridge

[93] fullclifmath.coef_

     array([ 5.9030777 ,  2.6452348 ,  1.6911944 , 10.44593273,  6.47287188])

[94] #displaying math intercept ridge

[95] fullclifmath.intercept_

     17.88057058238278

[96] #RMS of ridge math

[97] test_errorridgemath = rmse(Y_test, fullclifmath.predict(X_test))

[98] #print error ridge math

[99] print("Test RMSE Ridge:", test_errorridgemath)

     Test RMSE Ridge: 13.072340055562838
```

Lasso Regression Code

Lasso Regression Math Score

```
[148] #TRAIN test plit

[149] X_train, X_test, Y_train, Y_test = train_test_split(X, Y_math, test_size= 0.33, random_state = 2)

[150] #Lasso cross validation and find optimal alpha. Run lasso with optimal alpha

[151] lasmath = LassoCV(alphas= list, normalize= False).fit(X_train, Y_train)
      lasmath.alpha_
      full_lasmath = Lasso(alpha = lasmath.alpha_ , normalize= False).fit(X_train, Y_train)

[152] #Lasso math coefficent

[153] full_lasmath.coef_

      array([ 5.93547374,  2.64615034,  1.69445615, 10.51400041,  6.51653276])

[154] #Lasso Math intercept

[155] full_lasmath.intercept_

      17.647334693298447

[156] #Lasso math rmse error

      test_errorlassomath = rmse(Y_test, full_lasmath.predict(X_test))

[158] #Lasso math error print

[159] print("Test RMSE Lasso:", test_errorlassomath)

      Test RMSE Lasso: 13.074523665318557
```

Random Forest Code

```
[196] X_train, X_test, Y_train, Y_test = train_test_split(X, Y_math, test_size= 0.33, random_state = 2)
```

```
[197] RFCmath = RandomForestRegressor(n_estimators= 5000, random_state= 2)
```

```
RFCmath.fit(X_train, Y_train)
```

```
RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                      max_depth=None, max_features='auto', max_leaf_nodes=None,
                      max_samples=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=2, min_weight_fraction_leaf=0.0,
                      n_estimators=5000, n_jobs=None, oob_score=False,
                      random_state=2, verbose=0, warm_start=False)
```

+ Code + Text

```
[199] RFCmathprediction = RFCmath.predict(X_test)
```

```
[200] RFCmatherror= rmse(Y_test, RFCmathprediction)
```

```
[201] print(RFCmatherror)
```

```
15.111820053140164
```

Principal Component Analysis Code

```
[210] X_train, X_test, Y_train, Y_test = train_test_split(X, Y_math, test_size= 0.33, random_state = 2)
```

Principal Component analysis on the entire dataset

```
[211] XPCA = StandardScaler().fit_transform(X)
```

```
pca = PCA(n_components=2)
```

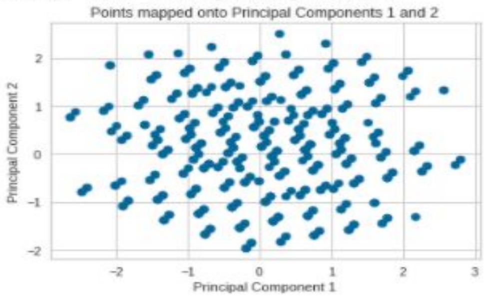
```
[213] principalcomp = pca.fit_transform(XPCA)
```

```
[214] principalDF = pd.DataFrame(data = principalcomp, columns = ['principal component 1', 'principal component 2'])
```

```
[215] finalDF = pd.concat([principalDF, Y_math], axis = 1)
```

```
[216] plt.title("Points mapped onto Principal Components 1 and 2")
      plt.scatter(data = principalDF, x = "principal component 1", y = 'principal component 2' )
      plt.xlabel("Principal Component 1")
      plt.ylabel("Principal Component 2")
```

```
Text(0, 0.5, 'Principal Component 2')
```



Results

Comparing different models

		Math Score		
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	13.630749359091372	13.072340055562838	13.074523665318557	15.111820053140164

		Reading Score		
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	15.427214220485427	12.887878273080332	12.887376769191185	14.861138065798864

		Writing Score		
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	14.717378952128481	12.637762224086966	12.636990889643377	14.471262075423892

		Total Score		
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	42.353325609155	37.3823489987393	37.383579282526064	42.991653577151006

Math Test Score

	Predictor	Linear math coefficients	Ridge math coefficients	Lasso math coefficients
0	gender	8.450756	5.903078	5.935474
1	race/ethnicity	3.483269	2.645235	2.646150
2	parental level of education	2.375814	1.691194	1.694456
3	lunch	13.534315	10.445933	10.514000
4	test preparation course	9.247094	6.472872	6.516533

Reading Test Score

	Predictor	Linear reading coefficients	Ridge reading coefficients	Lasso reading coefficients
0	gender	0.295253	-6.030107	-6.067670
1	race/ethnicity	3.787759	1.668664	1.669694
2	parental level of education	3.601566	1.876152	1.877670
3	lunch	14.539285	6.849392	6.897704
4	test preparation course	15.256613	8.293980	8.348255

Writing Test Scores

	Predictor	Linear writing coefficients	Ridge writing coefficients	Lasso write coefficients
0	gender	-2.600365	-8.157588	-8.208036
1	race/ethnicity	3.773361	1.905426	1.906679
2	parental level of education	3.980331	2.459127	2.461037
3	lunch	14.298928	7.510371	7.564256
4	test preparation course	16.995180	10.835996	10.906709

Total Test Scores

	Predictor	Linear total coefficients	Ridge total coefficients	Lasso total coefficients
0	gender	6.145644	-8.284617	-8.340255
1	race/ethnicity	11.044390	6.219324	6.222640
2	parental level of education	9.957712	6.026474	6.033271
3	lunch	42.372528	24.805696	24.976868
4	test preparation course	41.498888	25.602848	25.772400

Conclusion/Future steps:

Findings:

Lunch and Test Preparation has the biggest impact on a students test score performance

How to help:

- Offer lunch
- Test Preparation Course

Future step:

- Test the effect of running Random Forest Classification instead of Regression
- Using the fake Dataset during graduate school

Work Cited and Acknowledgements

Acknowledgement:

I would like to thank the course staff for helping me out with this project. The course staff's guidance and patience really helped me create a story and figure out the direction I wanted to take with the project. I am proud that this is my final project at Berkeley.

Work Cited:

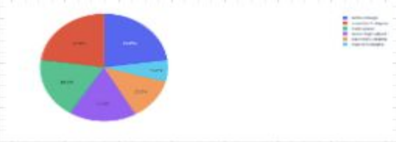
Seshapanpu, Jakki. "Students Performance in Exams." *Kaggle*, 9 Nov. 2018, www.kaggle.com/spscientist/students-performance-in-exams.

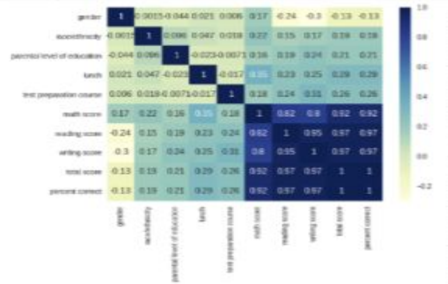
High Score Test Scores

DH 100 Theory and Methods | Channing Lee | 5/30

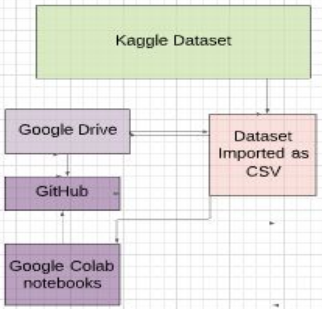
1) Dataset
https://www.kaggle.com/spscientist/students-performance-in-exams
(PDF, TXT, CSV)

2) How does factors such as gender, race/ethnicity, parental level of eduction, and etc affect student test performance

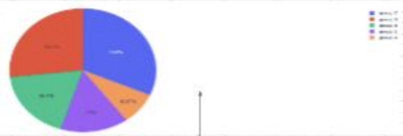


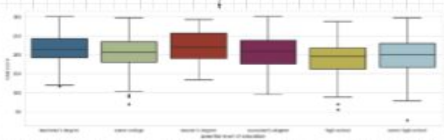


Introduction / "Hook"



1) High School Test Scores
2) DigHum 100 Theory & Methods in the Digital Humanities
3) Dr. Adam Anderson
4) Channing Lee





Descriptions:

This Dataset has 8 columns. It includes data on gender, race ethinicity, parent level of eduction, lunch, test preparations, and test score results for math, reading, and writing. I created additional columns such as total score and a percent correct.

I will be using Machine learning methods to figure out which factors affect the test score results the most. I will try methods such as linear regression, random forest, lasso regression, ridge regression, and principal components regression.

Discussion of results

Linear Regression:
Math Score: From the analysis we could see that the Parents level of eduction had the least amount of impact on Math scores while whether the students had lunch or not made the most amount of difference.
Reading Score: For reading score, gender had barely any impact at all while lunch and test preparation mattered the most
Writng Score: Gender had a negative impact on the scores and Once again Preparation had the most impact on Writing Performance.

Ridge Regression:
Math Score: Parental level of education had the least amount of impact while lunch had the most amount of impace
Reading Score: Gender had a negative impact on Reading and test preparation mattered the most
Writing Score: Gender also had a negative impact on Reading and Test preparation mattered the most.

Lasso Regression:
Math Score: Parent level of education had the least impact. Lunch had the most impact
Reading Score: Gender had a negative impact, Test prep most impact
Writing Score: Gender had negative impact, test prep most impact

Random Forest
Instad of using Random forest for classificiation, I used it for regression to find predictors of my model. Running Random Forest produced subpar results. Using Root Mean Squared Error as a metric, running Random Forest resulted in a lower Root Mean Squared Error than all the other methods.

PCA:
I was able to find split my dataset into components to reduce the deimisnsion of the data. Through PCA I saw a pattern in the dataset. I have to further examine more to produce substanctial resultes

Interpreting your results:

1) return to research question & how the results are answered by your methods.
2) It should explain how the visuals can be interpreted, and demonstrate your knowledge of the subject matter & corpus.

I have not ran my Machine learning methods yet but I created some visualizations so I can find trends in the data and hopefully develop some intuition.

Conclusions/Further steps

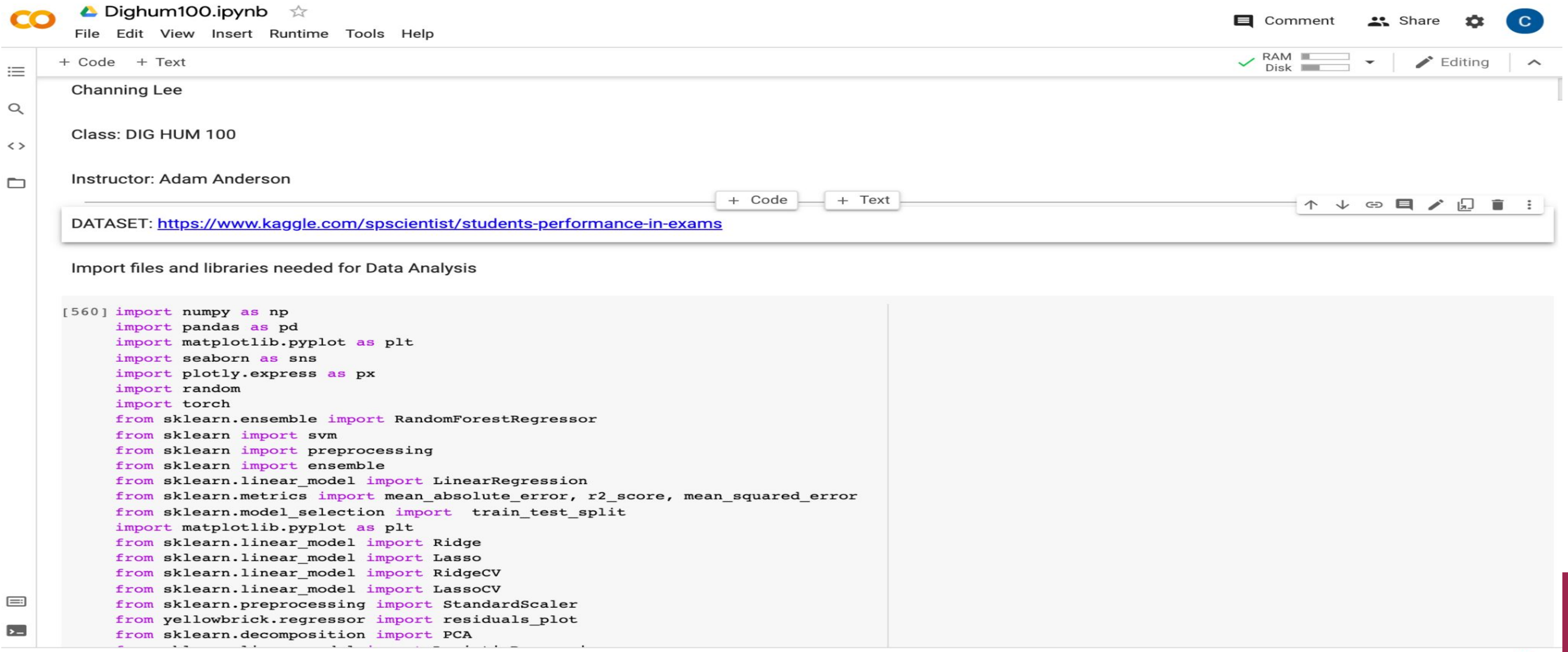
As a result of my research we are able to see that lunch and test preparation have the largest impact on test scores. To put my results into action I would recommend the school to offer lunch to students that may not have access to it. This would signficantly inhance a student's cognitive ability and allow them to focus in class resulting in high test scores. Another recommendation would be to put more time into standardize test preparation. Though standarize testing is dying out, it is almost impossible to receive admission to a top ranked school without a good ACT or SAT score. By providing students with a class period every week, students will be prepared for the test and have a bright future ahead of them

Work Cited
include the work you are citing:

1) The name of your project
2) DH 100
3) Instructor: Dr Anderson
4) Student: Channing Lee.
5) The works in the corpus you cite
6) Will include them later
7) Any helpful links?

If you fail to cite your work, you will be plagiarizing (whether intentionally or not).

Jupyter Notebook



Dighum100.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

Channing Lee

Class: DIG HUM 100

Instructor: Adam Anderson

DATASET: https://www.kaggle.com/spscientist/students-performance-in-exams

Import files and libraries needed for Data Analysis

```
[560] import numpy as np
      import pandas as pd
      import matplotlib.pyplot as plt
      import seaborn as sns
      import plotly.express as px
      import random
      import torch
      from sklearn.ensemble import RandomForestRegressor
      from sklearn import svm
      from sklearn import preprocessing
      from sklearn import ensemble
      from sklearn.linear_model import LinearRegression
      from sklearn.metrics import mean_absolute_error, r2_score, mean_squared_error
      from sklearn.model_selection import  train_test_split
      import matplotlib.pyplot as plt
      from sklearn.linear_model import Ridge
      from sklearn.linear_model import Lasso
      from sklearn.linear_model import RidgeCV
      from sklearn.linear_model import LassoCV
      from sklearn.preprocessing import StandardScaler
      from yellowbrick.regressor import residuals_plot
      from sklearn.decomposition import PCA
```

Google Colab link:
https://colab.research.google.com/drive/12d078fTSJF3AExwN5-tSGKZMdoCoStEs?usp=sharing

Github

earthimmortal Update README.md c0ca6b1 7 days ago 5 commits

Dighum100.ipynb	Created using Colaboratory	7 days ago
LICENSE	Initial commit	13 days ago
README.md	Update README.md	7 days ago
StudentsPerformance.csv	Add files via upload	7 days ago

README.md

DigHum100

What you will find:

- Dataset
- Jupyter Notebook
- Storyboard : https://drive.google.com/file/d/1KbSmSQX_mseeoyQNvFnfc3hLo4Yp7q_2/view?usp=sharing

Additional Information on items above:

- Dataset: Has code written for data exploration
- Jupyter Notebook: will find my code and documentation after each line of code explaining what I did
- Storyboard: Created using LucidChart. Used to display results/finding and analysis

topics provided.

Readme

MIT License

Releases

No releases published
Create a new release

Packages

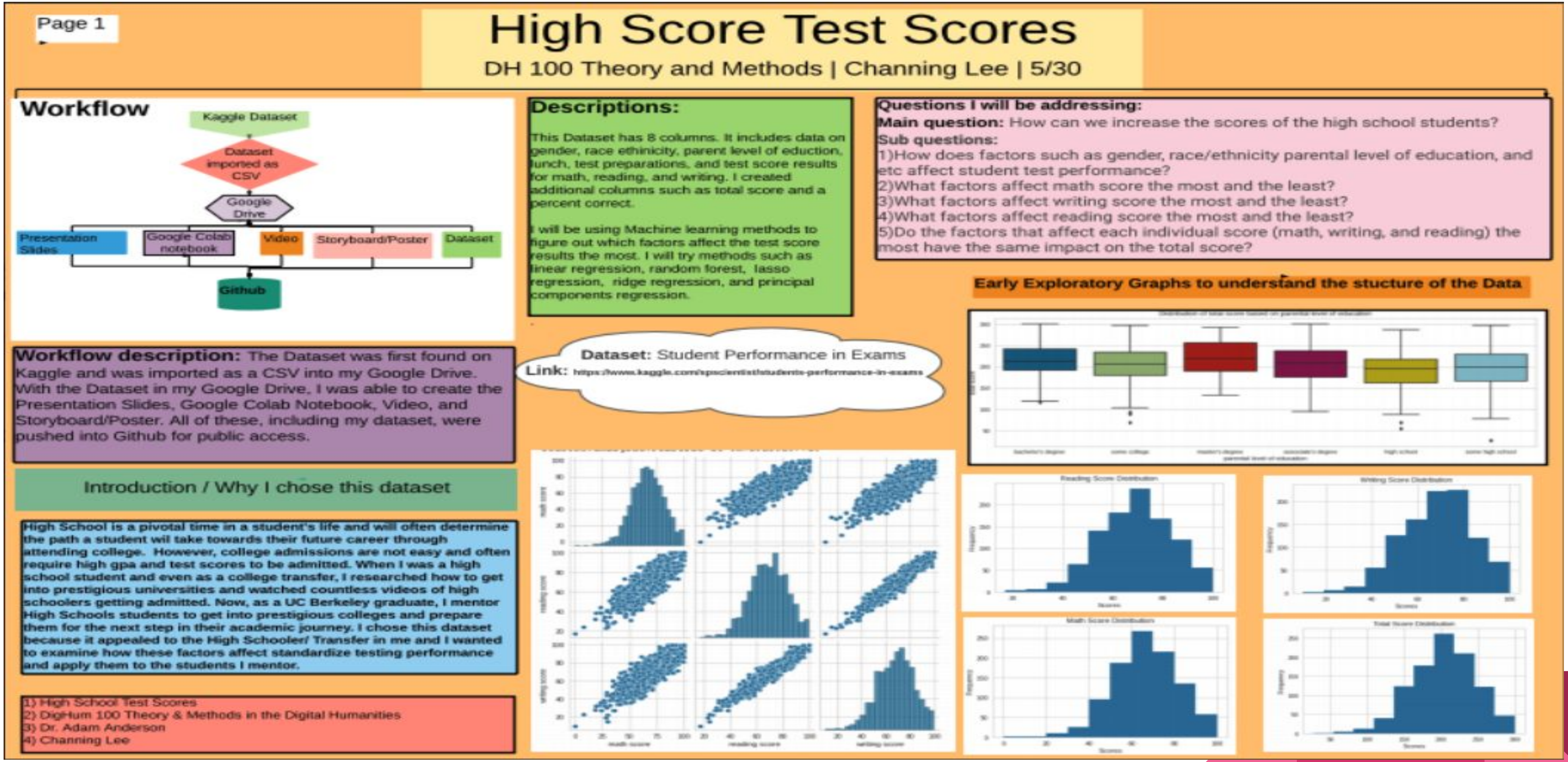
No packages published
Publish your first package

Languages

Jupyter Notebook 100.0%

Github link : https://github.com/earthimmortal/DigHum100

Poster Page 1



Google Drive Link:
https://drive.google.com/file/d/1Ke2e1E0u1W3dyRhfOIS1_nT_75YQqO1Y/view?usp=sharing

Poster Page 2

Page 2

Discussion of results/ Interpretation

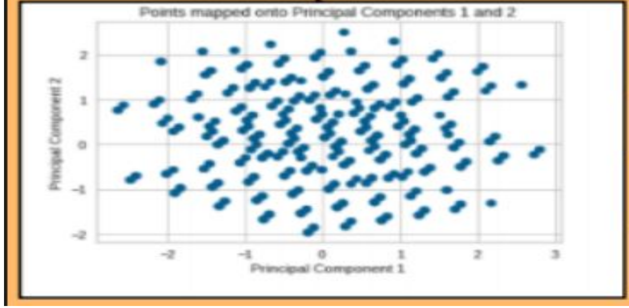
Linear Regression:
Math Score: From the analysis we could see that the Parents level of eduction had the least amount of impact on Math scores while whether the students had lunch or not made the most amount of difference.
Reading Score: For reading score, gender had barely any impact at all while lunch and test preparation mattered the most
Writng Score: Gender had a negative impact on the scores and Once again Preparation had the most impact on Writing Performance.

Ridge Regression:
Math Score: Parental level of education had the least amount of impact while lunch had the most amount of impace
Reading Score: Gender had a negative impact on Reading and test preparation mattered the most
Writing Score: Gender also had a negative impact on Reading and Test preparation mattered the most.

Lasso Regression:
Math Score: Parent level of education had the least impact. Lunch had the most impact
Reading Score: Gender had a negative impact, Test prep most impact
Writing Score: Gender had negative impact, test prep most impact

Random Forest
Instad of using Random forest for classificiation, I used it for regression to find predictors of my model. Running Random Forest produced subpar results. Using Root Mean Squared Error as a metric, running Random Forest resulted in a lower Root Mean Squared Error than all the other methods.

PCA:
I was able to find split my dataset into components to reduce the deimisnsion of the data. While plotting the the points on the first and second components of principal components, I found a pattern as seen on the left with the points forming three separate planes. This made it apparent to me that the dataset was fabricated and not real.



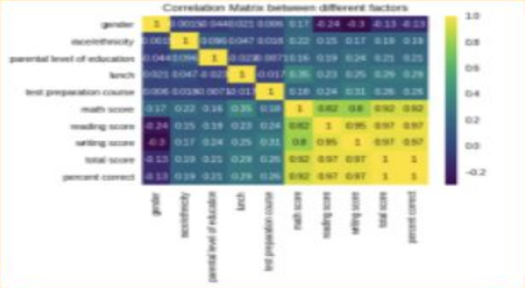
Predictor Coefficients

	Predictor	Linear math coefficients	Ridge math coefficients	Lasso math coefficients
0	gender	[illegible]	[illegible]	[illegible]
1	race/ethnicity	[illegible]	[illegible]	[illegible]
2	parental level of education	[illegible]	[illegible]	[illegible]
3	lunch	[illegible]	[illegible]	[illegible]
4	test preparation course	[illegible]	[illegible]	[illegible]

	Predictor	Linear total coefficients	Ridge total coefficients	Lasso total coefficients
0	gender	[illegible]	[illegible]	[illegible]
1	race/ethnicity	[illegible]	[illegible]	[illegible]
2	parental level of education	[illegible]	[illegible]	[illegible]
3	lunch	[illegible]	[illegible]	[illegible]
4	test preparation course	[illegible]	[illegible]	[illegible]

	Predictor	Linear reading coefficients	Ridge reading coefficients	Lasso reading coefficients
0	gender	[illegible]	[illegible]	[illegible]
1	race/ethnicity	[illegible]	[illegible]	[illegible]
2	parental level of education	[illegible]	[illegible]	[illegible]
3	lunch	[illegible]	[illegible]	[illegible]
4	test preparation course	[illegible]	[illegible]	[illegible]

	Predictor	Linear writing coefficients	Ridge writing coefficients	Lasso write coefficients
0	gender	[illegible]	[illegible]	[illegible]
1	race/ethnicity	[illegible]	[illegible]	[illegible]
2	parental level of education	[illegible]	[illegible]	[illegible]
3	lunch	[illegible]	[illegible]	[illegible]
4	test preparation course	[illegible]	[illegible]	[illegible]



Root Mean Squared Error Table (Used to compare model performance)

	Math Score			
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	13.630748359091372	13.072340055562838	13.074523465318557	15.111820053140184

	Reading Score			
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	15.427214223485427	12.887878273308032	12.887376769191185	14.861138065798864

	Writing Score			
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	14.717378952128481	12.637762224086966	12.636590889643177	14.471282075423892

	Total Score			
	Linear Regression	Ridge Regression	Lasso Regression	Random Forest
RMSE	42.353325609915 5	37.382348998 7393	37.383579282526064	42.991653571151036

Conclusions/Further steps

By looking at the coefficients for each response variable we are able to see that lunch and test preparation have the largest impact on test scores. To put my results into action I would recommend the school to offer lunch to students that may not have access to it. This would signficantly inhance a student's cognitive ability and allow them to focus in class resulting in high test scores. Another recommendation would be to put more time into standardize test preparation. Though standarize testing is dying out, it is almost impossible to receive admission to a top ranked school without a good ACT or SAT score. By providing students with a class period every week, students will be prepared for the test and have a bright future ahead of them.

Some other future steps I would like to try in the future is binning the responce variables so I can run Random Forest classification instead of regression. Often, Random Forest classification out performs Linear, Lasso, and Ridge. However, it did not out perform the rest.

Upon learning that my dataset is fake, I learned from the professor that it was probably used as a project proposal for research. Since I am interested in attending a Graduate School in Data Science, I may try to use /build off this dataset and build intituation on factors that really affect test score performance.

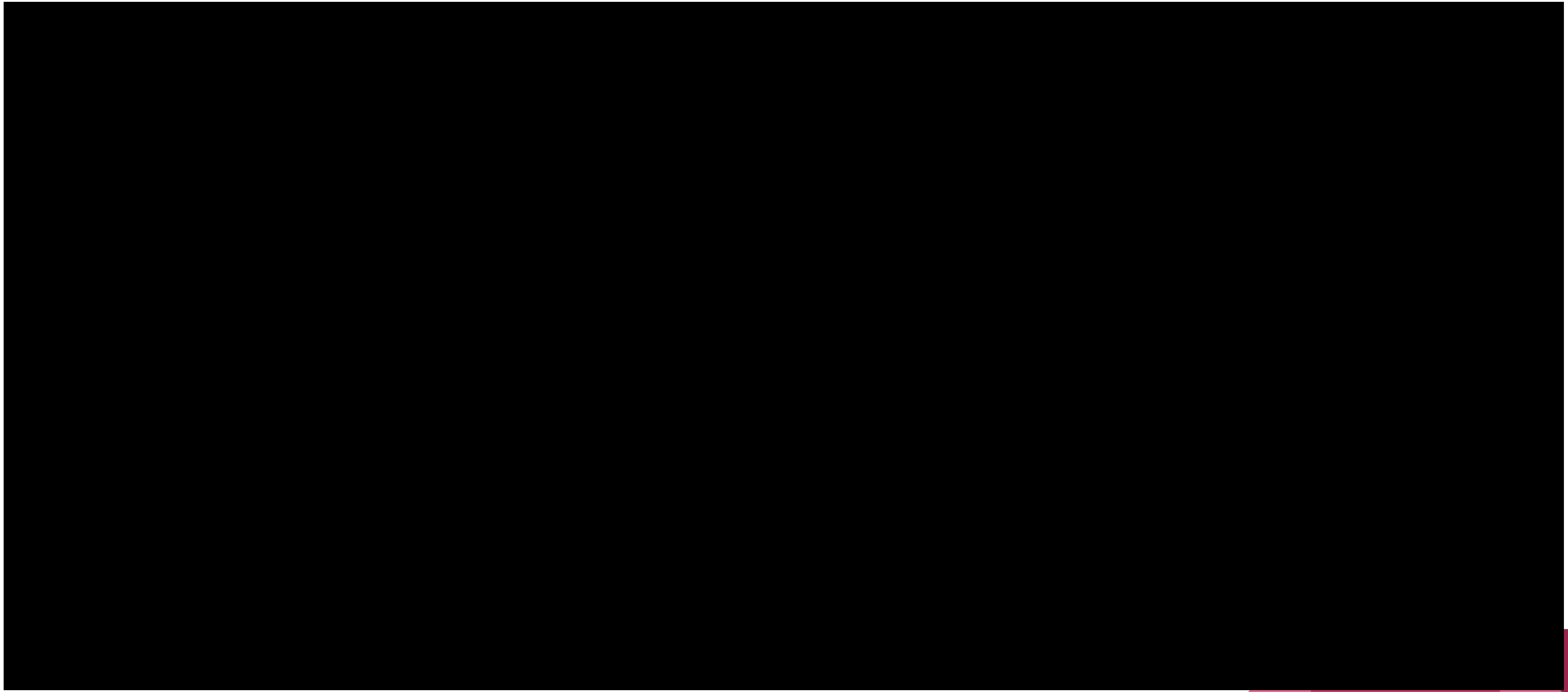
Work Cited

1) High School Test Scores
2) DH 100
3) Instructor: Dr. Anderson
4) Student: Channing Lee.

Work cited:
Seshapanpu, Jakki. "Students Performance in Exams." Kaggle, 9 Nov. 2018, www.kaggle.com/spscientist/students-performance-in-exams

Link to Github Repository:
https://github.com/earthimmortal/DigHum100

Video



Link to same video in Google Drive:
https://drive.google.com/file/d/1mPzi1ixj6NtrwQubXuUe5rMPe2BmYZ8f/view?usp=sharing