Assignment 1: Linear Regression

Nathaniel Reeves - UTU CS 4320 Machine Learning

Assignment Description

Use this Student PerformanceLinks to an external site. dataset and the principles discussed in class to build a model that predicts student performance based on the available features.

Code submission will not be required for this assignment.

Also submit a report here of your findings. The report should include descriptions of the features and the label, data exploration plots, the process used, any insights found and used for fine-tuning, and the MSE and MAE loss for training data and testing data.

Big Picture

The goal of this project is to train a model that can accurately predict a students grade/preformance based on their; study hours, sleep hours, socioeconomic score, and attendance.

Get Data

The data for this project was pulled from the Kaggle link in the assignment description. This is the description and key features (columns) of the data on the Kaggle webpage:

Description

This dataset is a synthetic representation of student performance, designed to mimic real-world scenarios by considering key factors such as study habits, sleep patterns, socioeconomic background, and class attendance. Each row represents a hypothetical student, and the dataset includes both input features and the calculated target variable (grades).

The dataset can be used for predictive modeling, exploratory data analysis, or even as a beginner-friendly introduction to machine learning workflows.

Key Features

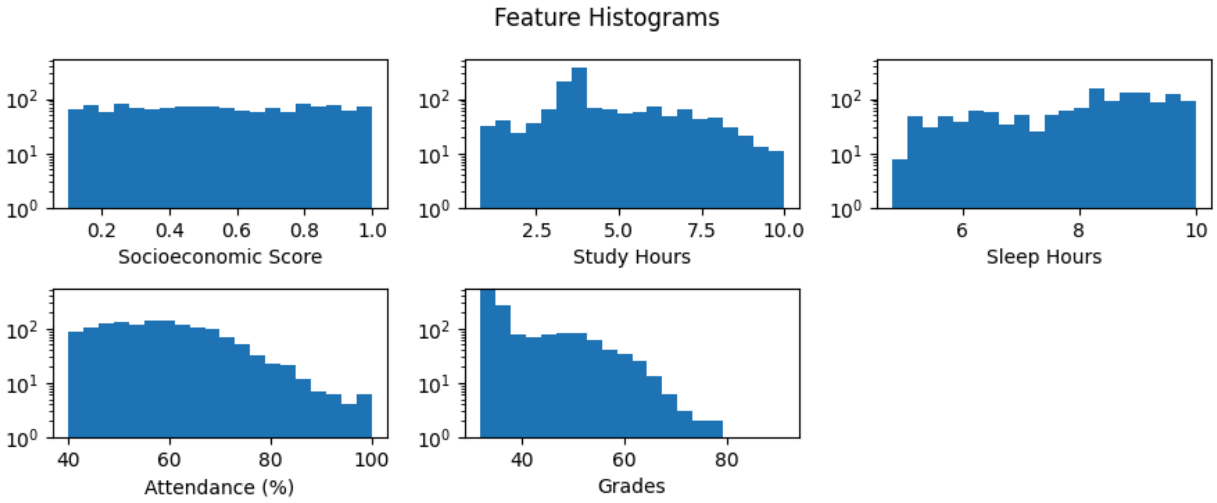
1. Study Hours
    a. Description: Average daily hours spent studying.
2. Sleep Hours
    a. Description: Average daily hours spent sleeping.
3. Socioeconomic Score

   a. Description: A normalized score (0-1) indicating the student's socioeconomic background.

4. Attendance (%)
   a. Description: The percentage of classes attended by the student.
5. Grades (TARGET)
   a. Description: The final performance score of the student, derived from a combination of study hours, sleep hours, socioeconomic score, and attendance.

Explore/Visualize

Histograms and scatterplots were made to identify any patterns in the data.



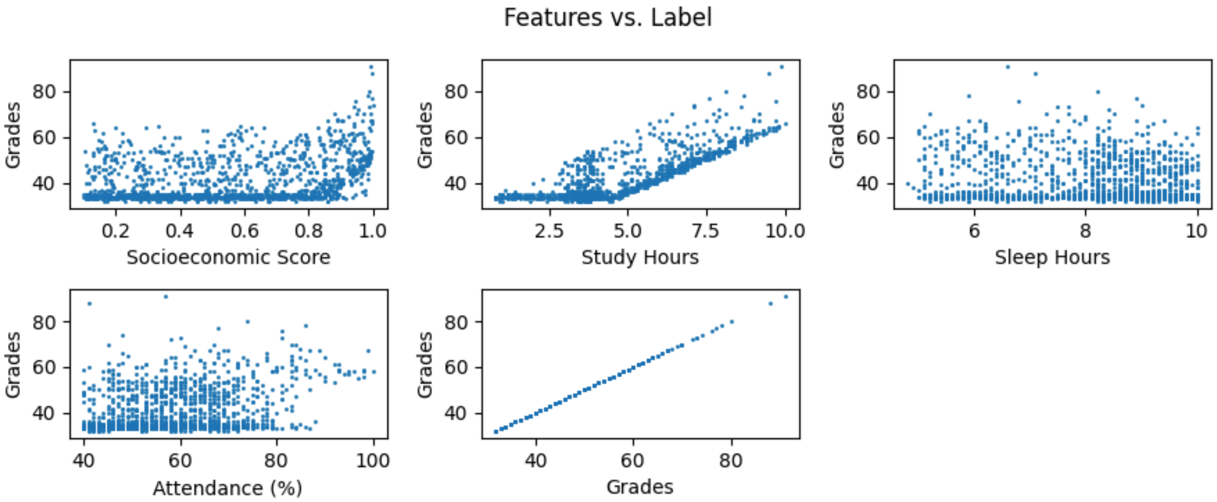
Socioeconomic Score: Seems the data has an even spread indicating students have an even distribution in this category. This variable will likely have little effect in the model.

Study Hours: This also seems to have an easy spread with a potential bias between the 2.5 and 5 hour marks indicating most students study at least 3 hours.

Sleep Hours: A slight but noticeable trend upwards indicating the students that more students get between 8-10 hours of sleep.

Attendance: A gradual slope downwards, indicating less students attend their classes 100% of the time.

Grades: A sharp slope downwards from about 50% indicating most students get below 50% grades.



Socioeconomic Score: Over all, the spread seems quite even except for a tick up near the end indicating students that have a better socioeconomic score get slightly better grades.

Study Hours: This chart seems to show a strong connection to study hours and student grades. Students who study for only 2.5 to 5 hours get lower grades.

Sleep Hours: This chart suggests students who get longer sleep (between 8-10 hours) also get better grades.

Attendance: This is the first chart to show possible outliers in my opinion. Over all students who don't attend class get lower grades with the exception of those two datapoints near the 40 (x) and 50 (x) mark.

Grades: Nothing to really see there.

Prepare Data

Over all, the data seems to be relatively clean. From my previous analysis, I only identified two potential points that might cause problems. After some thought, I decided to leave these points in the dataset. Their impact should be small compared to the data in the other columns.

The data was split using the standard 0.8 : 0.2 training to testing ratio. The seed 42 was used to make the models reproducible.

Select/Train Model

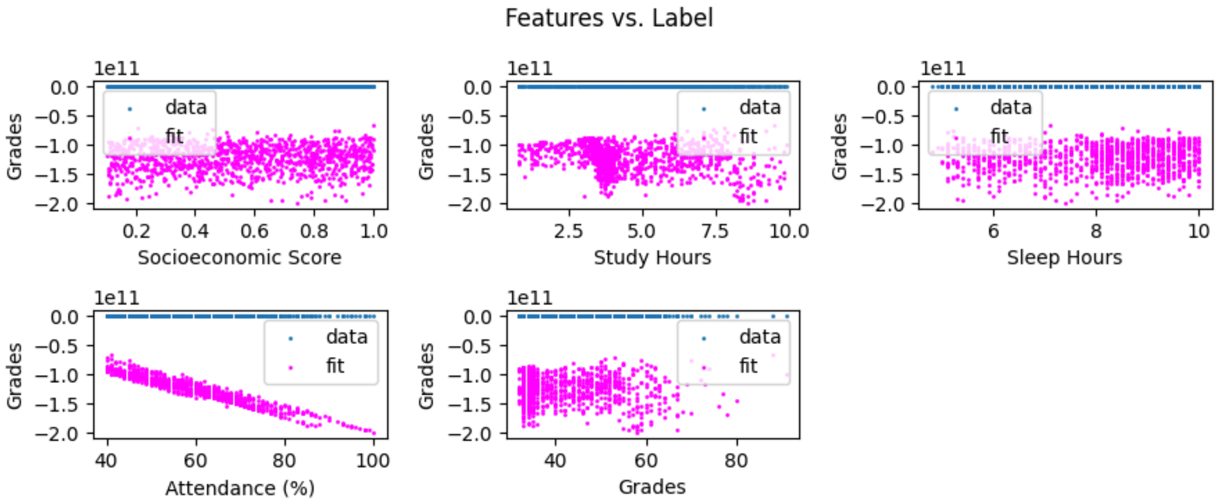
The submission for this assignment was a bit late, however this gave me the opportunity to note down the models other students used and to give them each a try. Here are the results of the classes most popular models along with the assignment default model. These models do not have any normalization/scaling steps. The data was just fed directly to training and testing.

	TRAIN			TEST		
	R^2	MSE	MAE	R^2	MSE	MAE
SGDRegressor	-1.258	1.169	29314733941	-1.511	1.140	29091669144
Ridge	0.779	20.46	3.492	0.744	19.27	3.385
Lasso	0.665	31.05	4.207	0.630	27.90	4.102
BayesianRidge	0.779	20.46	3.490	0.744	19.26	3.384
MLPClassifier	0.760	22.25	3.653	0.725	20.74	3.523
LinearRegression	0.779	20.46	3.492	0.744	19.27	3.386
RandomForestRegressor	0.997	0.261	0.360	0.980	1.451	0.898

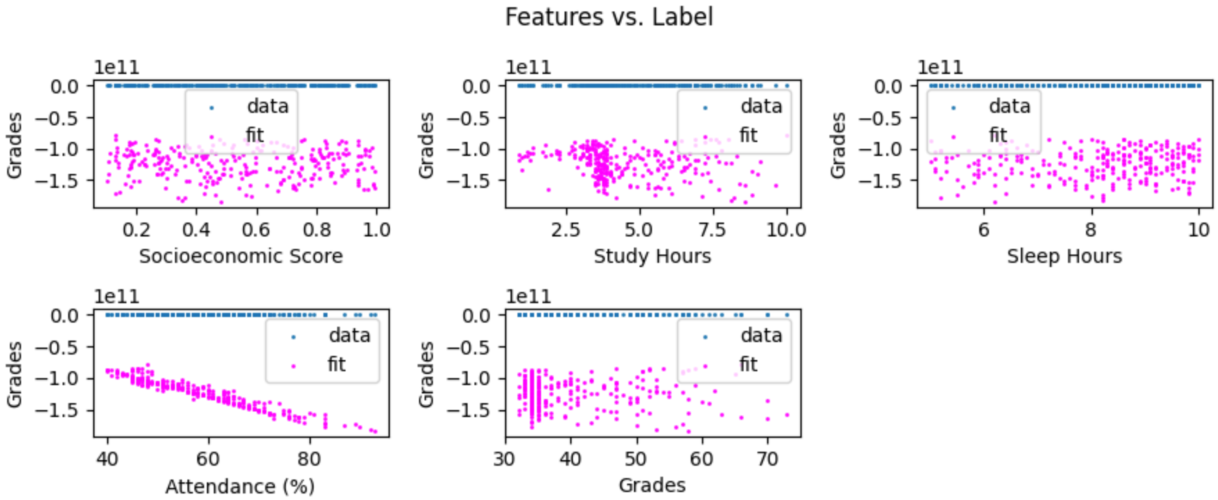
I noticed a huge difference in performance between the SGDRegressor and the RandomForestRegressor training models. I thought to plot the models training and testing to get a better look at their differences.

I was going mostly off the different R^2 values to make the observation.

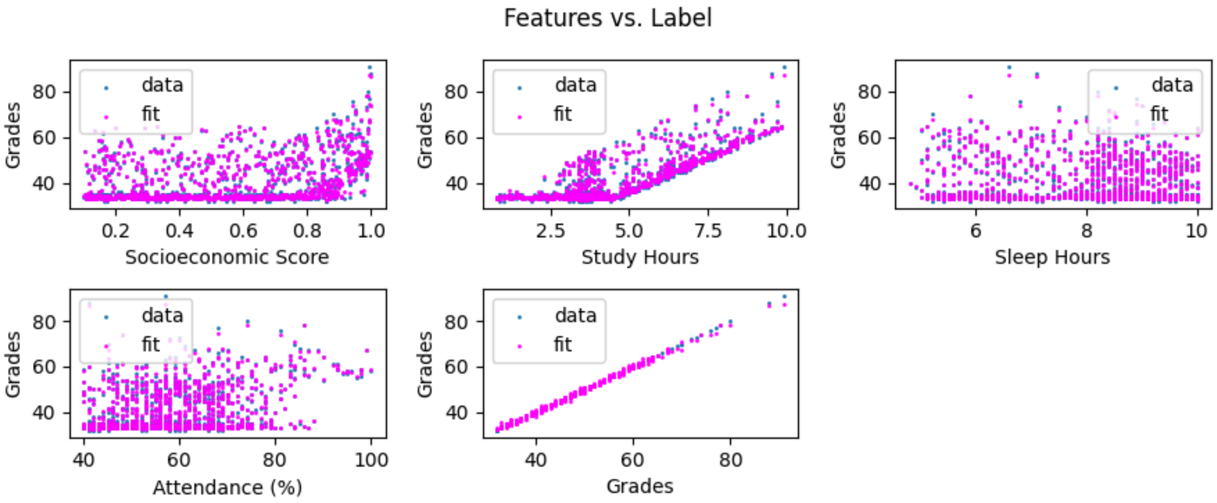
SGDRegressor - Train



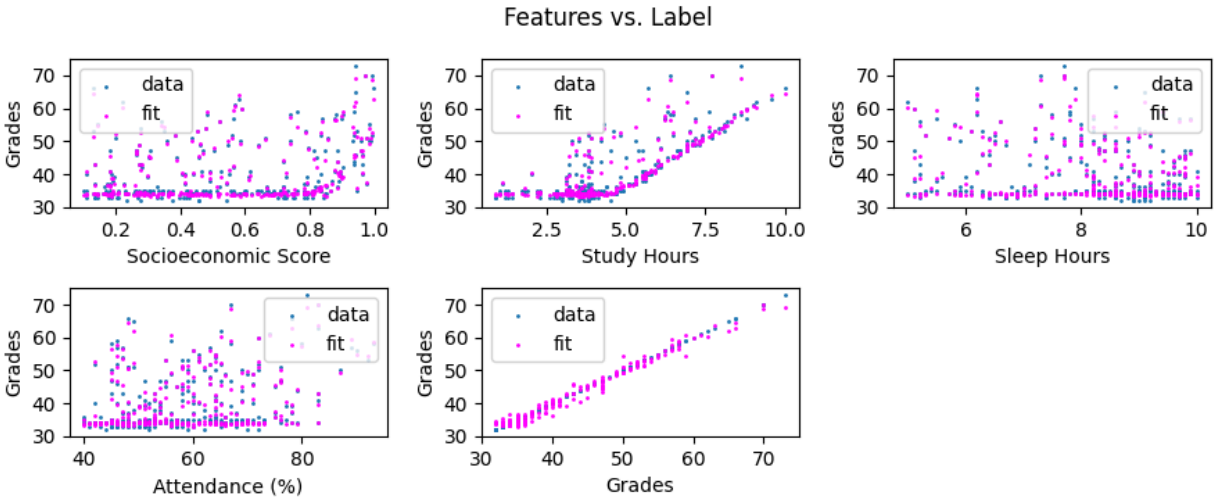
SGDRegressor - Test



RandomForestRegressor - Train



RandomForrestRegressor - Test



It was really cool to see the RandomForrestRegressor predict datapoints that lined up so close to the actual data while the SGDRegressor pick values so bad the real data became squished together in the plots.

Fine-tune the Model

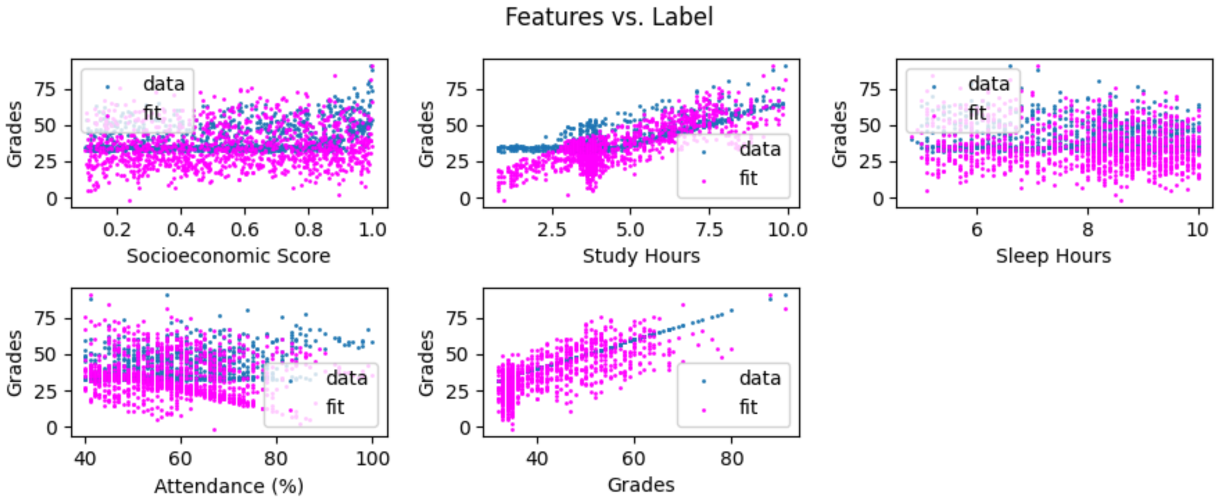
After implementing the normal fit code to the selected models, I tried scaling/normalizing the data before training to see if any models would improve. I particularly wanted to see the effect of scaling on the SGDRegressor. Here are the results I got scaling the data before training.

	TRAIN			**TEST**		
	R^2	MSE	MAE	R^2	MSE	MAE
SGDRegressor	-0.021	94.90	7.81	-0.294	97.68	7.87
Ridge	-0.095	101.83	8.08	-0.39	105.0	8.129
Lasso	-10.93	1108.81	32.44	-13.75	1113.69	32.610
BayesianRidge	-0.094	101.70	8.075	-0.389	104.87	8.124
MLPClassifier	-3179.3	295537	533.34	-3927.2	296506	534
LinearRegression	-0.126	104.63	8.187	-0.430	107.94	8.237
RandomForestRegressor	-5.391	593.97	22.75	-7.30	626.68	23.64

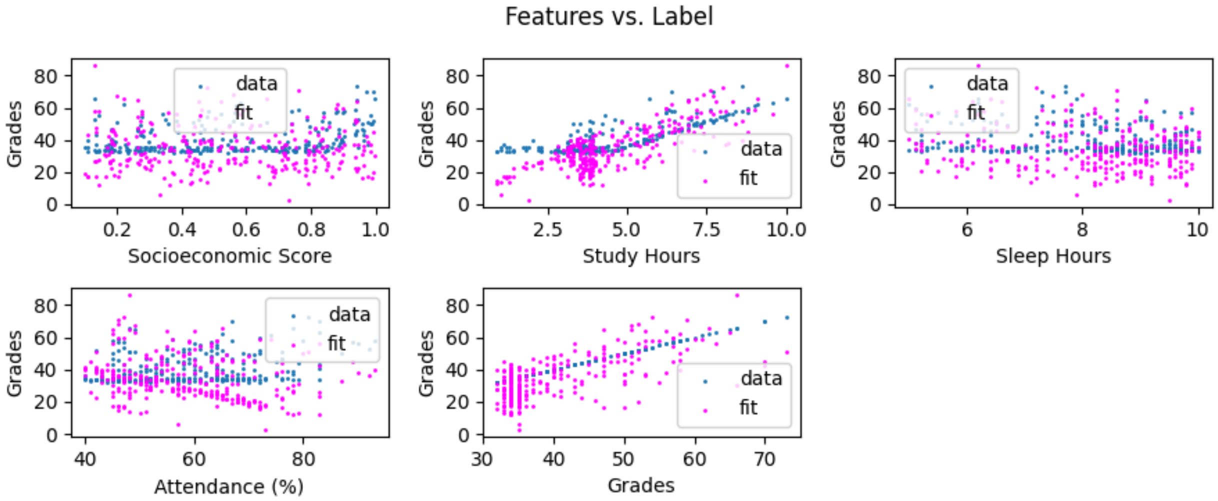
While scaling greatly improved the performance of SGDRegressor, it seems to have made all the other models worse. I am not sure why this is the case. Initially, I thought I mixed up my initial results with my scaled results, however after careful code review this wasn't the case.

It is interesting to note how much worse the MLPClassifier model performed. I decided to make new plots of the SGDRegressor vs the MLPClassifier to see if there was anything interesting.

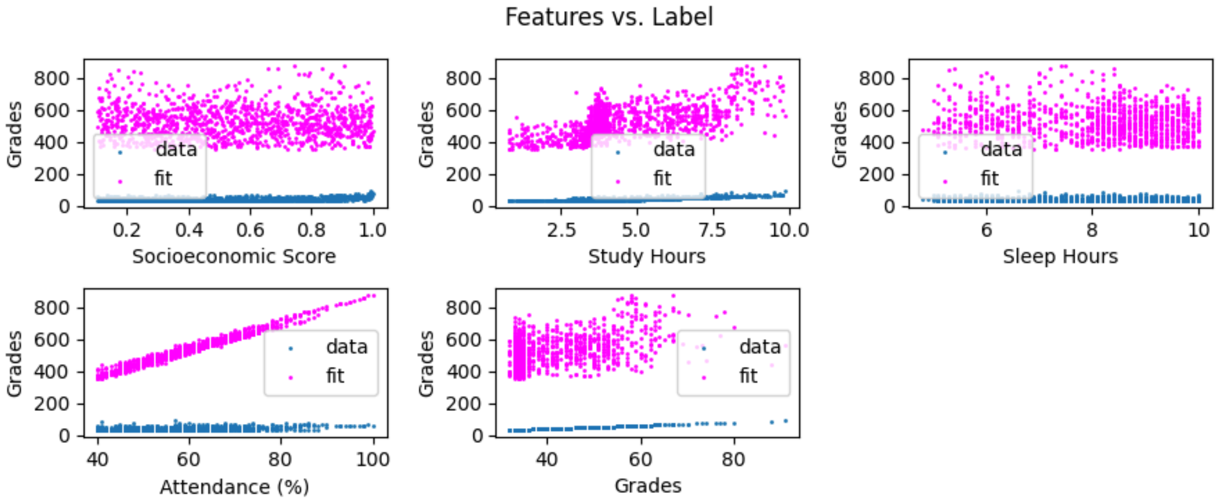
SGDRegressor - Train



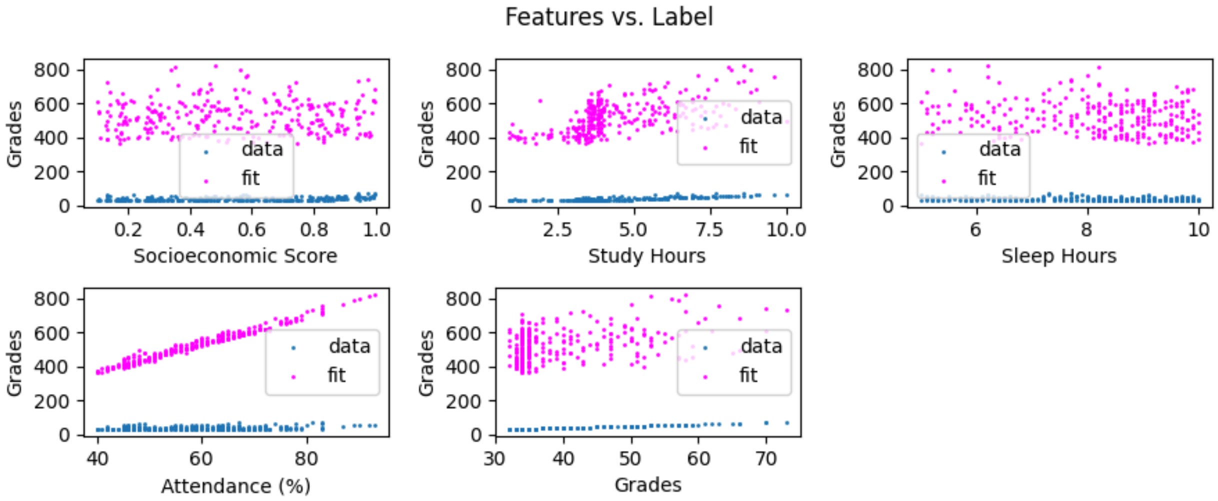
SGDRegressor - Test



MLPClassifier - Train



MLPClassifier - Test



Present Solution/ Production Ready?

Based on the performance of the raw data RandomForestRegressor model, with a R^2 of 0.98 on test data, it seems the RandomForestRegressor model could potentially be production worthy. All the other models, while interesting, do not preform well enough. Testing the RandomForestRegressor model on this data was a lucky pick.