Homework 4 Report
Calvin Schaul
IDSN542 Machine Intelligence

Background Information from Lab 5

This assignment is a continued analysis of the Single Precision General Matrix Multiplication (SGEMM) dataset. In short, this dataset has numerical and categorical attributes that capture technical specifications and processing settings of GPUs doing matrix multiplication. The result data, and the target for our ML models to predict, is the average runtime in milliseconds of a GPU with specific settings to complete its calculations.

Expert insight from the dataset source suggested that this dataset works well with linear regression and also works well by analyzing average runtimes on a normal and logarithmic scale.

Model Selection

Based on the author's insight and my in-class observations, Linear Regression and Random Forest Regression models were used to predict the GPU runtimes.

The dataset listing on kaggle [1] shows that this source data from the UC Irvine Machine Learning Repository [2] was used in other academic environments to test custom implementations of linear regression. Since we covered this model in our coursework, it made sense for me to use this as one of my models.

The second model used in my code was the Random Forest Regressor. I chose this model after our in-class discussion on Random Forest using a series of Decision Tree models to avoid overfitting. I also wanted to avoid overfitting and getting bad prediction results so I used Random Forest.

Performance Analysis

Below are the performance metrics of each model's base predictions before and after cross-validation training. For linear regression, cross validation used 10 folds. According to Machine Learning Mastery k=10 is a common low-bias k-value [3]. For cross validation training of the random forest model, 5 folds were used for performance reasons to avoid excessive training times.

Specifically the root mean square error (RMSE) of the base predictions and of the cross validation scores was used to analyze the model's performance.

Training Data Performance

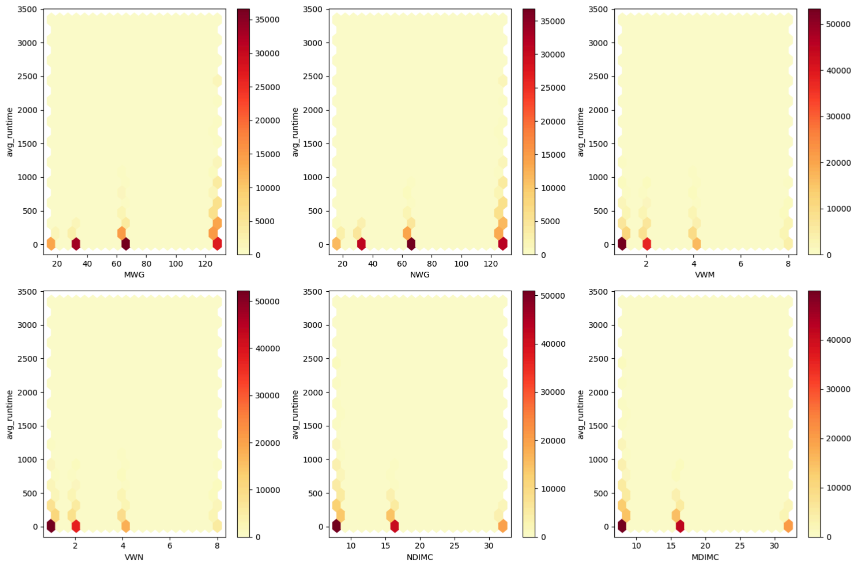
Evaluation	**Root Mean Square Error (ms)**
Training Linear Regression RMSE	283.8190353483322
Training Linear Regression Cross-Validation Scores	[284.89391106 281.20172164 284.88269672 289.02299268 273.41787776 281.5555231 286.49535851 279.58046662 277.97427655 298.81823879]
Training Random Forest Regression RMSE	1.4723369463869793
Training Random Forest Regression Cross-Validation Scores	[4.56799363 4.5232401 4.47803724 4.5232869 4.71859122]

Testing Data Performance

Final Linear Regression Test RMSE: 283.97052947029226 ms
Final Random Forest Regression Test RMSE: 3.972258198894714 ms

The SGEMM dataset contained runtimes ranging from 13.25 ms to 3,397.08 ms. With this context into how much our RMSE runtime across the models would impact our predictions, it is clear that the Random Forest model performed significantly better, specifically Random Forest had approximately 71 times lower RMSE.

In this case, Random Forest performed better due to the non-linear relationships between the GPU settings/specifications and average runtime across all of the possible configurations in the dataset. Below is a figure from my previous analysis of this dataset, showing a heatmap of runtimes and different settings. Low runtime was observed at a variety of different settings and not in a clear linear pattern.



SGEMM Runtime Heatmap - darker colors indicate more samples at a given configuration/runtime

Log Scale Observations

Further analysis was conducted on a logarithmic scale of average runtimes. In order to better understand our SciKit learn ML toolset, another model was trained using the natural logarithm of average runtimes, cross validated with 10 folds, and evaluated on training and testing data.

Training Data Performance

Evaluation	**Root Mean Square Error (ms)**
Training Linear Regression RMSE	0.7481006578225423
Linear RegressionAverage Cross Validation Score RMSE	0.7481604605666989

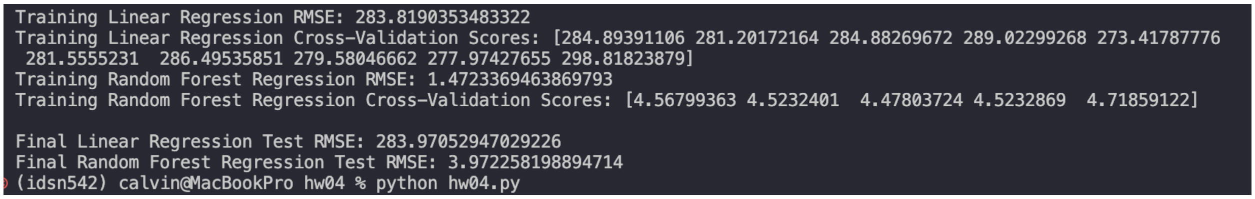
Testing Data Performance

Linear Regression Test RMSE (log space): 0.7453576690162044 ms

With a log of average runtimes ranging from 2.58 ms to 8.13 ms, this linear regression model in log space proves to be a much more reasonable option and can stack up with the Random Forest model from above.

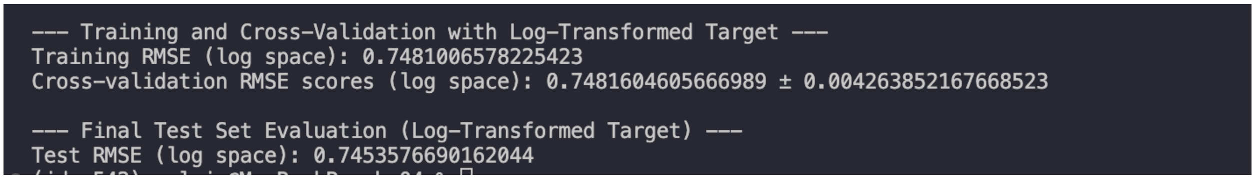
Appendix

Pasted Code output:



```
Training Linear Regression RMSE: 283.8190353483322
Training Linear Regression Cross-Validation Scores: [284.89391106 281.20172164 284.88269672 289.02299268 273.41787776
 281.5555231  286.49535851 279.58046662 277.97427655 298.81823879]
Training Random Forest Regression RMSE: 1.4723369463869793
Training Random Forest Regression Cross-Validation Scores: [4.56799363 4.5232401  4.47803724 4.5232869  4.71859122]

Final Linear Regression Test RMSE: 283.97052947029226
Final Random Forest Regression Test RMSE: 3.972258198894714
(idsn542) calvin@MacBookPro hw04 % python hw04.py
```



```
 --- Training and Cross-Validation with Log-Transformed Target ---
 Training RMSE (log space): 0.7481006578225423
 Cross-validation RMSE scores (log space): 0.7481604605666989 ± 0.004263852167668523

 --- Final Test Set Evaluation (Log-Transformed Target) ---
 Test RMSE (log space): 0.7453576690162044
```

References

[1] SGEMM Kaggle Listing (https://www.kaggle.com/datasets/rupals/gpu-runtime)
[2] UC Irvine Machine Learning Repository SGEMM Source data (https://archive.ics.uci.edu/dataset/440/sgemm+gpu+kernel+performance)
[3] Machine Learning Mastery. *How to configure K-Fold Cross Validation*. (https://machinelearningmastery.com/how-to-configure-k-fold-cross-validation/)