Applying Some ML algorithm on Hubble Tension Problem

Amorhossein Feiz Ahmad Ramezanpour Abolfazl Shirmahalleie

January 2022

Abstract

This project applied traditional machine learning techniques and neural networks on two different datasets to the Hubble Tension problem. We discussed the performance of models and compared them, and finally, we reported the Hubble parameters which resulted from our models.

1 Introduction

The difference between the prediction of the standard model for the Hubble parameter and the observational values is known as the Hubble Tension. Our approach is to use observational data and apply regression algorithms to find the optimal relationship between the Hubble parameter and cosmological quantities. In this project, we used two data sets: Cosmic Chronometers and Pantheon, which are described further. First, we will discuss about our data sets, and after that, we explain the efficiency of the models that we used in two different sections: Traditional ML algorithms and Neural Networks.

2 Data sets

There is two data sets for this problem, and we used just the first one in our project. Due to the low number of data points in the second data sets we were not able to use NN for them. However, we applied some traditional ML algorithm on the Cosmic Chronometers out of this project.

2.1 Pantheon

Each data point is a supernova and the main quantity which is our label is apparent magnitude in band B or m_b. Our features are Z_{cmb} which is redshift of supernova relative to CMB background, and error of m_b. The goal is finding the relation between m_b and z_{cmb}. Using m_b we can find luminosity distance and using luminosity distance we can find Hubble parameter.

2.2 Cosmic Chronometers

This data set contains 31 data points; each is a galaxy with old stellar populations and low star formation rates in redshift $z < 2$. Redshift and error of H are our features and Hubble parameter is label of each data point. These measurements are independent of the Cepheid distance scale and do not rely on any particular cosmological model. For instacne, we applied Cat Boost Regressor on this data set out of our project and get the Hubble value $H_0 = 70.13 \pm 20.36$ $(km/s)/Mpc$ with the MSE:0.167.

3 Traditional Models

3.1 Linear Regression

Linear Regression is the supervised Machine Learning model in which the model finds the best fit linear line between the independent and dependent variable.

- We used three different metrics and the value of loss for them was:

  Mean absolute error: 1.03534

  Mean squared error: 1.6905

  Root mean squared error: 1.3002

- Training time: 7.92 ms

  Test time: 4.08 ms

- Slope:9.3412

  Intercept:18.9794

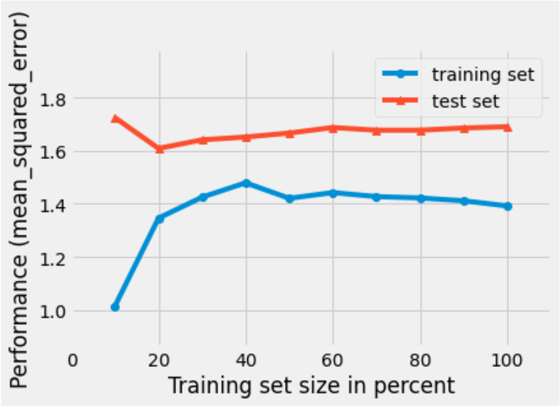


Figure 1: mean squared error in terms of training test size in percent

3.2 Random Forest Regressor

Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. A Random Forest operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees.

- We used three different metrics and the value of loss for them was:

  Mean absolute error: 0.57024

  Mean squared error: 0.5930

  Root mean squared error: 0.77011

- Training time: 588 ms

  Test time: 108 ms

  Grid time: 4min 57s

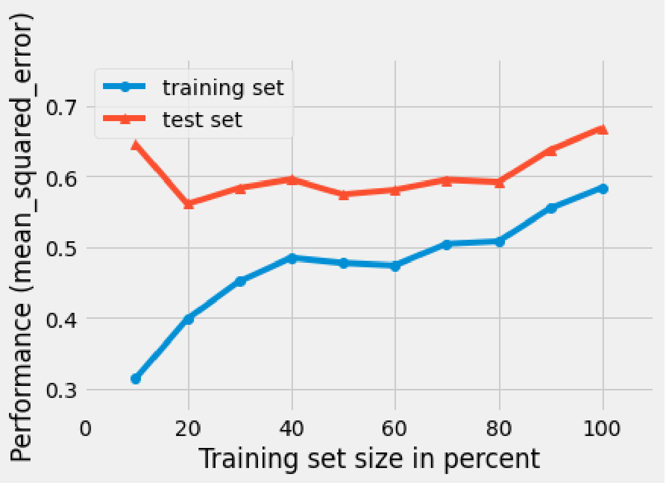


Figure 2: mean squared error in terms of training test size in percent

3.3 Xgboost regressor

Xgboost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

- We used three different metrics and the value of loss for them was:

Mean absolute error: 7.0253

Mean squared error: 51.5716

Root mean squared error: 7.1813

- Training time: 283 ms

  Test time: 3.9 ms

  Grid time: 3.67 s

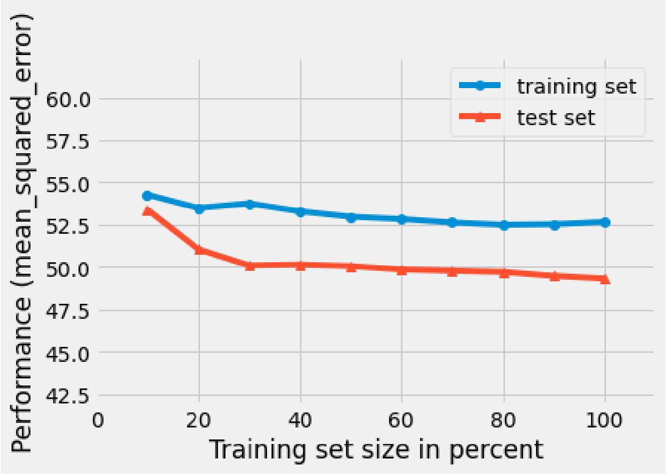


Figure 3: mean squared error in terms of training test size in percent

3.4 Ridge Regression

Ridge regression is a method of estimating the coefficients of multiple-regression models in scenarios where independent variables are highly correlated.

- We used three different metrics and the value of loss for them was:

  Mean absolute error: 1.0167

  Mean squared error: 1.5378

  Root mean squared error: 1.2400

- Training time: 502 ms

  Test time: 4.64 ms

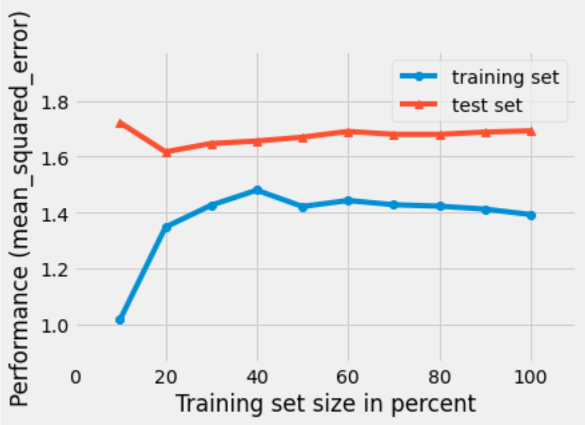


Figure 4: mean squared error in terms of training test size in percent

3.5 Lasso Regression

lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the resulting statistical model.

- We used three different metrics and the value of loss for them was:

  Mean absolute error: 1.0162

  Mean squared error: 1.5374

  Root mean squared error: 1.2399

- Training time: 504 ms

  Test time: 4.15 ms

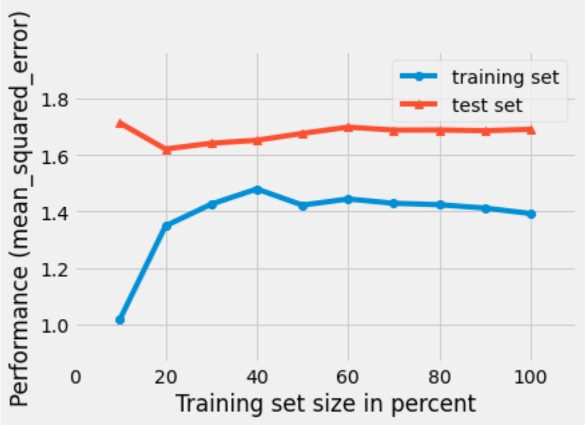


Figure 5: mean squared error in terms of training test size in percent

4 Neural Networks

We applied three NN model on the Pantheon data set.

4.1 First Model (Sequential)

- We used two different metrics and the value of loss for them was:

  Mean absolute error: 0.2678

  Mean squared error: 0.1251

- Training time: 8.23 s

  Test time: 135 ms

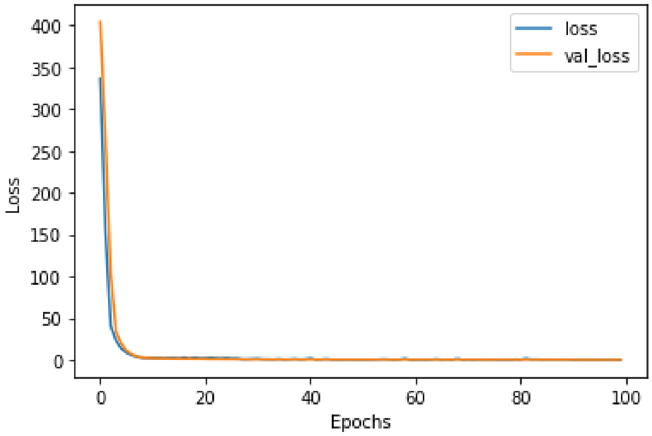


Figure 6: Loss in terms of epochs for the first model

4.2 Second Model (Sequential)

- We used two different metrics and the value of loss for them was:

  Mean squared error: 1.0115

  Root mean squared error: 0.6358

- Training time: 34.9 s

  Test time: 109 ms

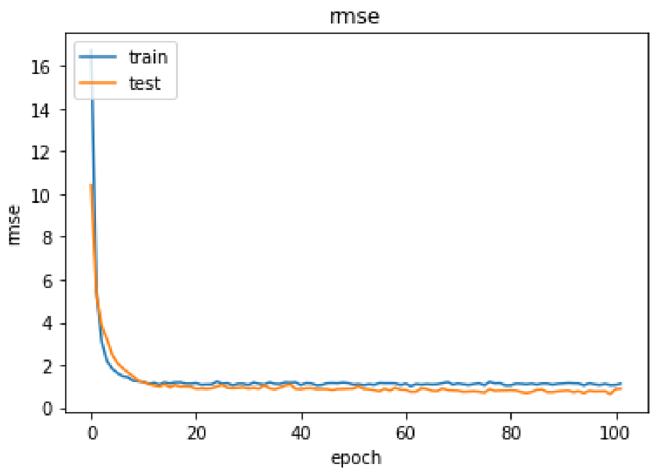


Figure 7: Loss in terms of epochs for the second model

4.3 Third Model (Functional)

- We used two different metrics and the value of loss for them was:

  Mean absolute error: 0.1416

  Mean squared error: 0.0389

- Training time: 5.89 s

  Test time: 70 ms

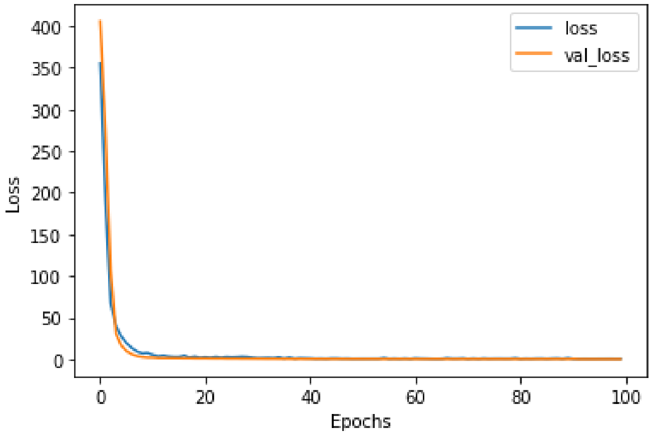
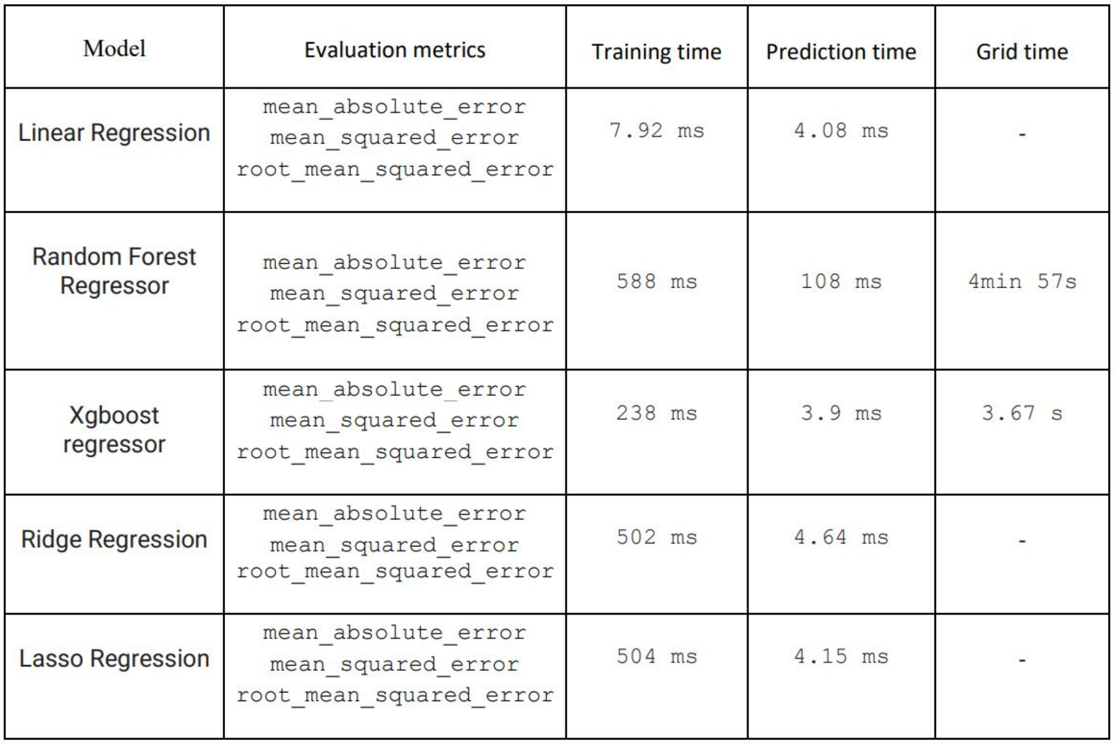


Figure 8: Loss in terms of epochs for the third model

5 Conclusion

5.1 Traditional Models

- Mean squared error and mean absolute error are appropriate metrics for regression.
- Obviously, Linear Regression is not a good model for this problem because it fits a linear curve.
- Random Forest Regressor returned a low value for error and it seems to be a good model for us.
- The large error for Xgboost Regressor can be for the complexity of this model that causes over fitting. For instance, this over fitting does not happen in Linear Regression because it is a simple model.



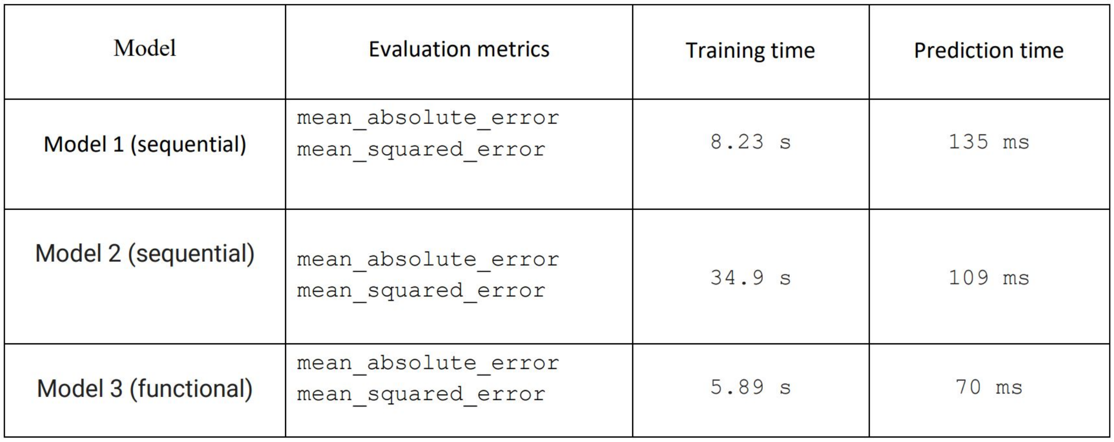
Model	Evaluation metrics	Training time	Prediction time	Grid time
Linear Regression	mean_absolute_error mean_squared_error root_mean_squared_error	7.92 ms	4.08 ms	-
Random Forest Regressor	mean_absolute_error mean_squared_error root_mean_squared_error	588 ms	108 ms	4min 57s
Xgboost regressor	mean_absolute_error mean_squared_error root_mean_squared_error	238 ms	3.9 ms	3.67 s
Ridge Regression	mean_absolute_error mean_squared_error root_mean_squared_error	502 ms	4.64 ms	-
Lasso Regression	mean_absolute_error mean_squared_error root_mean_squared_error	504 ms	4.15 ms	-

Figure 9: Compare table for traditional models

5.2 Neural Networks

- We used kernel initializer in these models to generates tensors with a normal distribution.
- Because the data is not very complex and does not happen over fitting, we did not use drop out layers and regularization.

- Mean squared error and mean absolute error are appropriate metrics for this task.



Model	Evaluation metrics	Training time	Prediction time
Model 1 (sequential)	mean_absolute_error mean_squared_error	8.23 s	135 ms
Model 2 (sequential)	mean_absolute_error mean_squared_error	34.9 s	109 ms
Model 3 (functional)	mean_absolute_error mean_squared_error	5.89 s	70 ms

Figure 10: Compare table for traditional models

References

- https://github.com/dscolnic/Pantheon
- https://arxiv.org/pdf/1802.01505
- https://arxiv.org/pdf/1710.00845
- https://arxiv.org/pdf/2104.00595
- https://hal.archives-ouvertes.fr/hal-02098120
- https://www.researchgate.net/publication/355912818_Hubble_Tension
- https://arxiv.org/pdf/2106.15656