Predicting Stellar Mass

Anastasia Gorina and Andres Chaves

Introduction and Motivation

NASA's search for exoplanets and their host stars has been going on for over 30 years resulting in tons of data and over 4000 confirmed planets.

We decided to explore the information about planets and stars and build a multiple regression model that would predict the stellar mass using the information about the planetary systems.

Process

- Obtaining the data from NASA API
- Exploring with Pandas and Matplotlib
- Cleaning the data with numpy (deleting incomplete rows and columns with redundant information)
- Normalization
- Testing for linear model assumptions
- Building and adjusting the model with scipy and scikit learn
- Train-test splitting and cross-validation
- Ridge regularization
- Picking the best model

Choosing the predictors

- After cleaning the data we ended up with a table with following columns: planetary orbital period, eccentricity, planetary mass, planetary radius, number of planets in the system, stellar optical magnitude, stellar temperature, stellar radius, stellar mass.

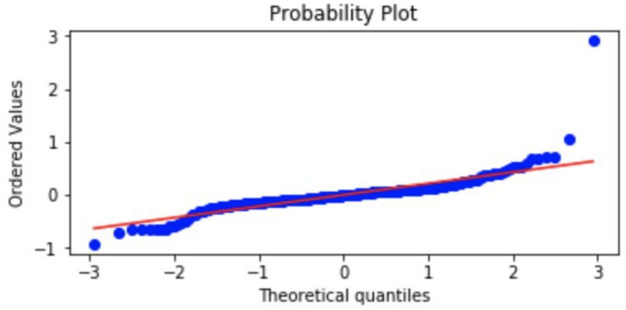
- We decided to set our dependent variable to stellar mass and the columns with information about planets to our independent variables. This gave us R-squared of 0.302 and high p-values

- We dropped the column with the highest p-value, number of planets, but our R-squared didn’t change

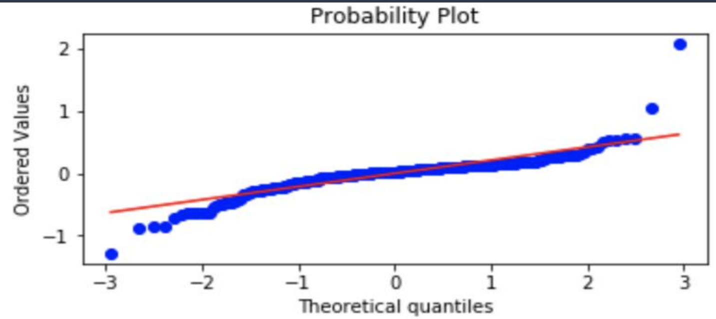
- Then we included the stellar columns and got R-squared of 0.795

- We figured out that the highest contributors were stellar optical magnitude, stellar temperature, and stellar radius and decided to proceed with this information as our X.

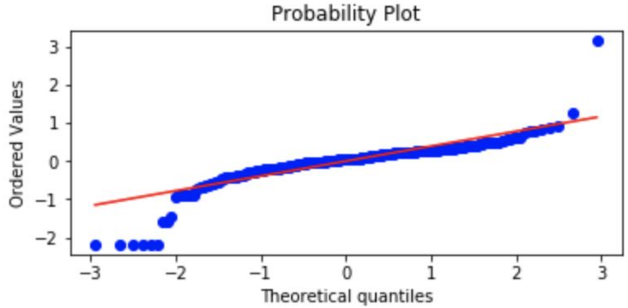
Q-Q plots: checking for normality



Stellar temperature/mass

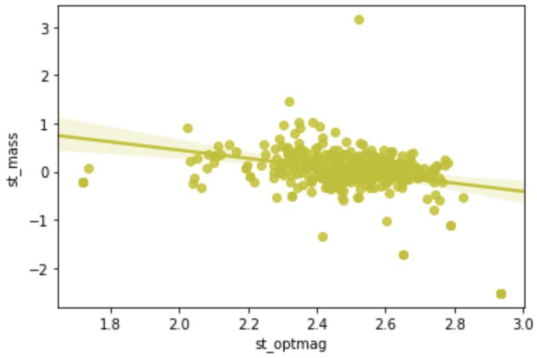


Stellar radius/mass

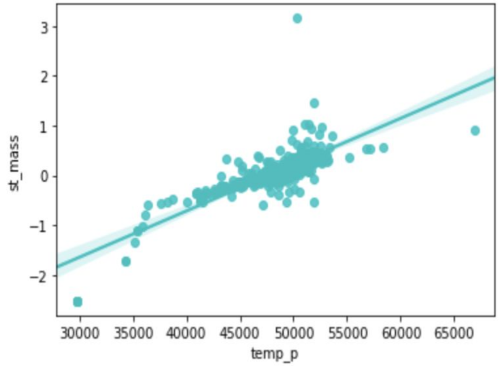


Luminocity/Mass

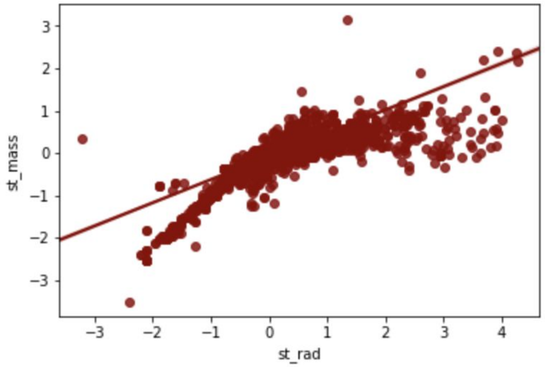
Pair by Pair simple models:



Luminocity/Mass
Pearson -0.3



Temperature/Mass
Pearson 0.8



Radius/Mass
Pearson 0.8

Adjusting the model

Model	R^2	Error Train	Error Test	Formula
Log X,y	0.792	9.6	9.9	$y = -9.6 + 0.5x_1 + 1.15x_2 - 0.15x_3$
Log + Ridge	0.860	-0.03	-0.06	
Log + x^2	0.722			$y = -9.26 - 0.09x_1^2 + 0.12x_2^2 - 0.04x_3^2$
Log + Polynomial	0.796	8.7	10.3	$y = -2.9 + 0.1x_1^3 + 6.23x_2^5 - 0.004x_3^5$
Log + Polynomial + Ridge	0.796	8.7	10.3	

Checking Kepler

r = The longest radius of an elliptic orbit
T = The time the planet crosses the stellar limb

$$M_s = \frac{4\pi^2 r^3}{GT^2}$$

$$ln(M_s) = ln(\frac{4\pi^2 r^3}{GT^2})$$

$$ln(M_s) = ln(4\pi^2 r^3) - ln(GT^2)$$

$$= ln(4\pi^2) - ln(G) + 3ln(r) - 2ln(T)$$

$$ln(M_s) = A + Bln(r) - Cln(T)$$

$$ln(M_s) = A + B * ln(Radius) - C * ln(Transit)$$

$$ln(M_s) = 2.8807 * ln(r) - 2.4902 * ln(T)$$

This model performed really well with really high R^2 (0.998), however, we lost the constant term A.

Conclusion and Future Thoughts

- Stellar mass can be predicted with good accuracy
- Bigger dataset can help improve the model as well as taking the measurement error into account
- Accurate model can help us find a perfect planetary system to move to after the sun turns into a red giant and Earth becomes unsuitable for life