Jason Long

Dr. A. Jafari

DATS6203: Machine Learning II

29 April 2019

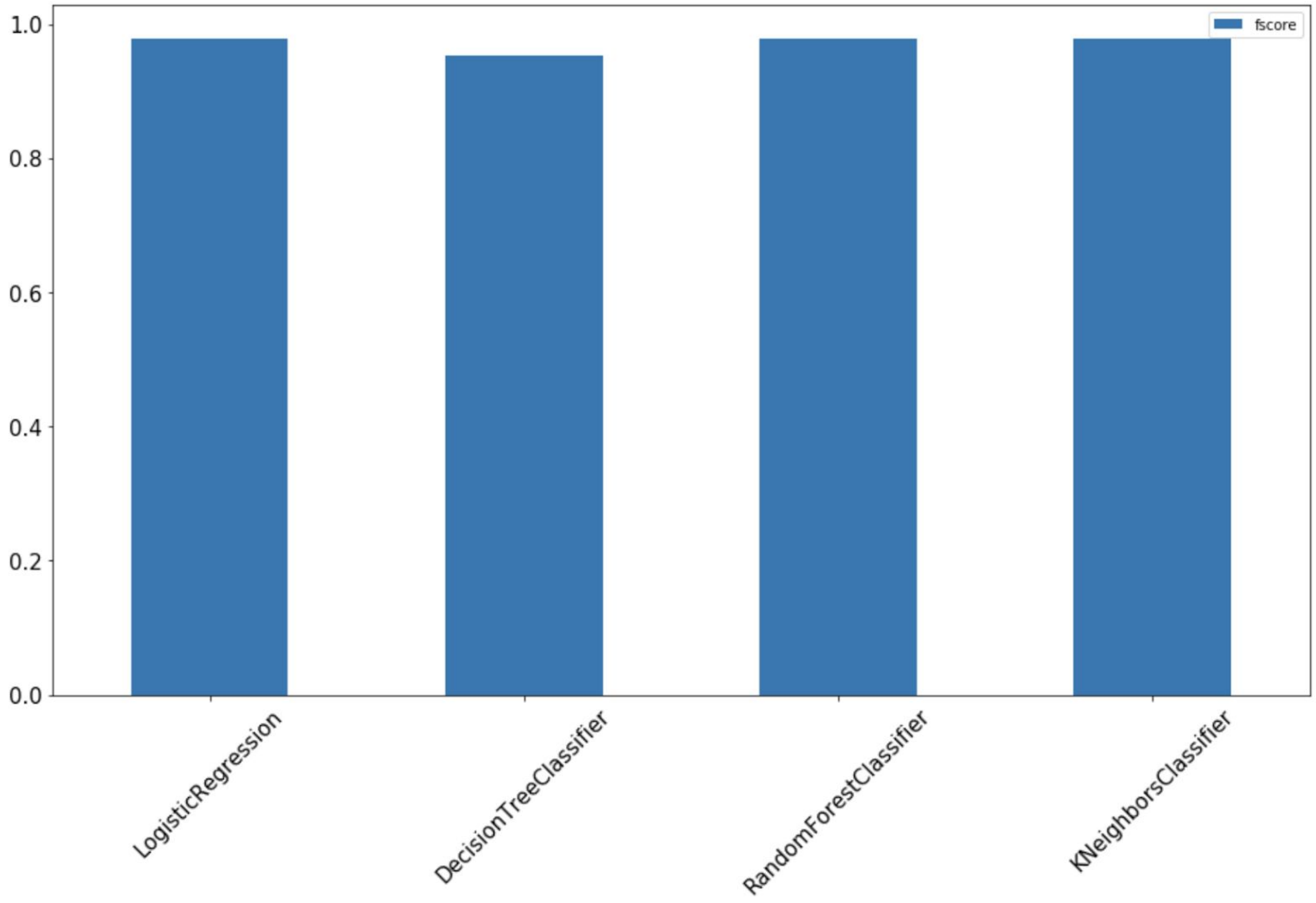
Individual Final Report

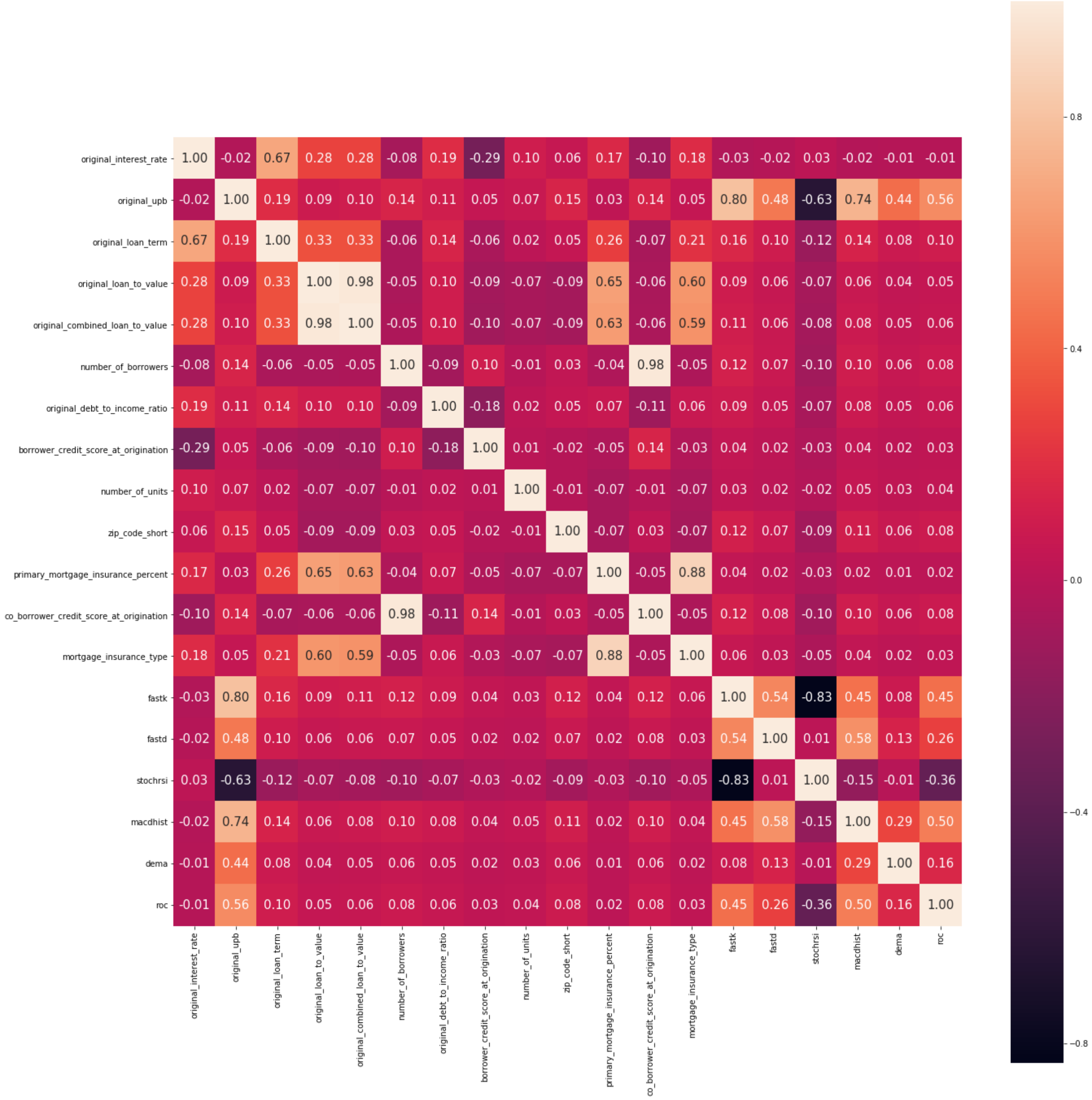
Using Pytorch and a Multi-Layer Perceptron Network, our goal is to predict whether a mortgage loan for a single-family home will fall delinquent by one or more months. The work was distributed as follows; Luke Bogacz would address getting the data to the model, Vishal Sinha would create and test models, and Jason Long would create visualizations and address project deliverables/management. All would experiment with models as time permits and share high performance models.

Individual Work

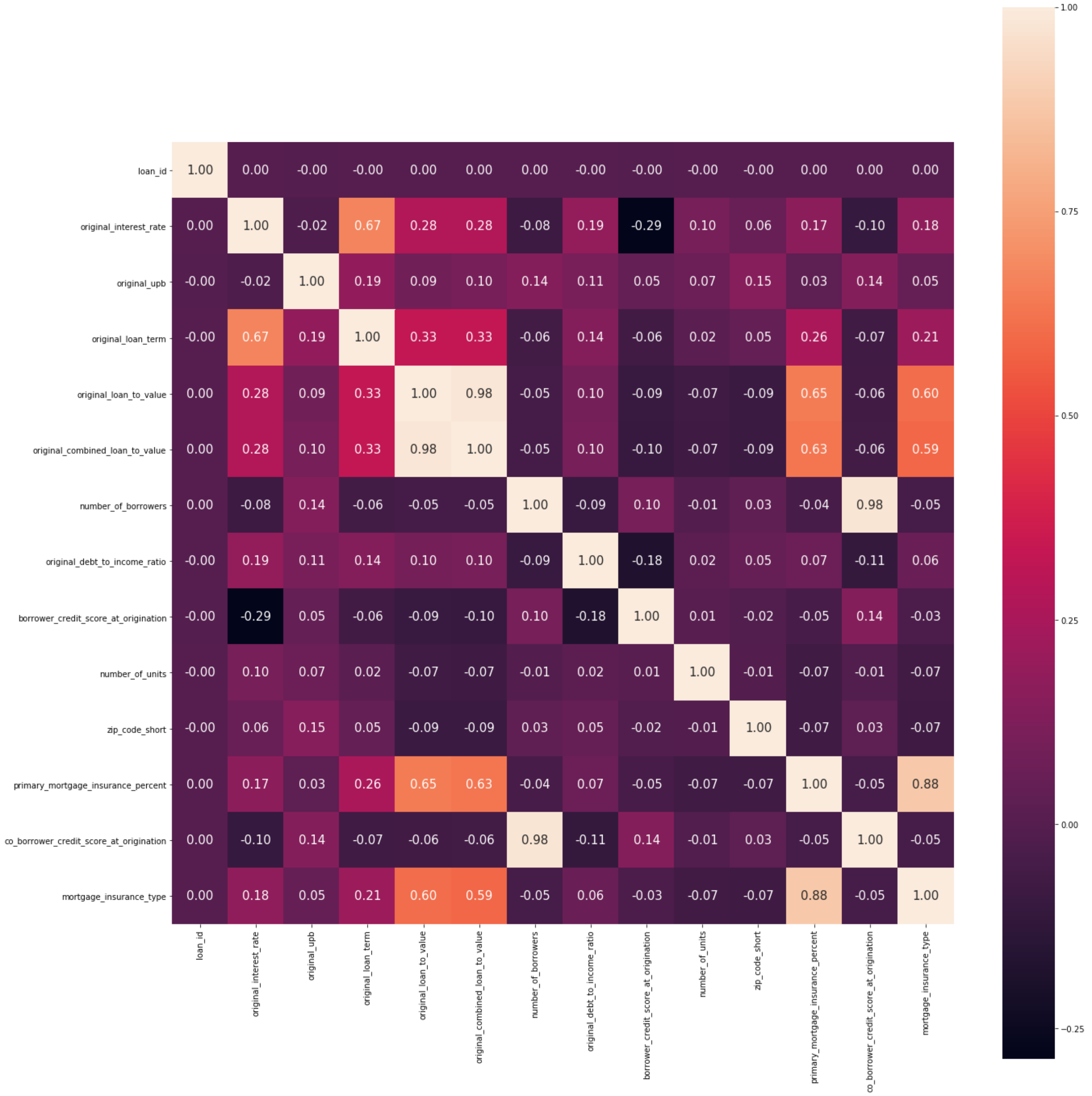
While awaiting the model being created so I could create visualizations, I spent time doing analytics on the dataset. Looking for collinearity and multicollinearity within the data to assist in dealing with the size of the data. Early models were having issues with data size as there were over 1,000 features and at times were causing memory issues during testing. In response to this, I upgraded my graphic processing unit to one that had a 50% larger amount of memory. This overcame some of the issues when dealing with the data; however, it exposed that there were some errors within the model. As those were overcome by the team, I continued data analytics.

These analytics were performed with Python within a Anaconda Jupyter Notebook, the notebook and associated code are located within the Code folder in my individual project submission.

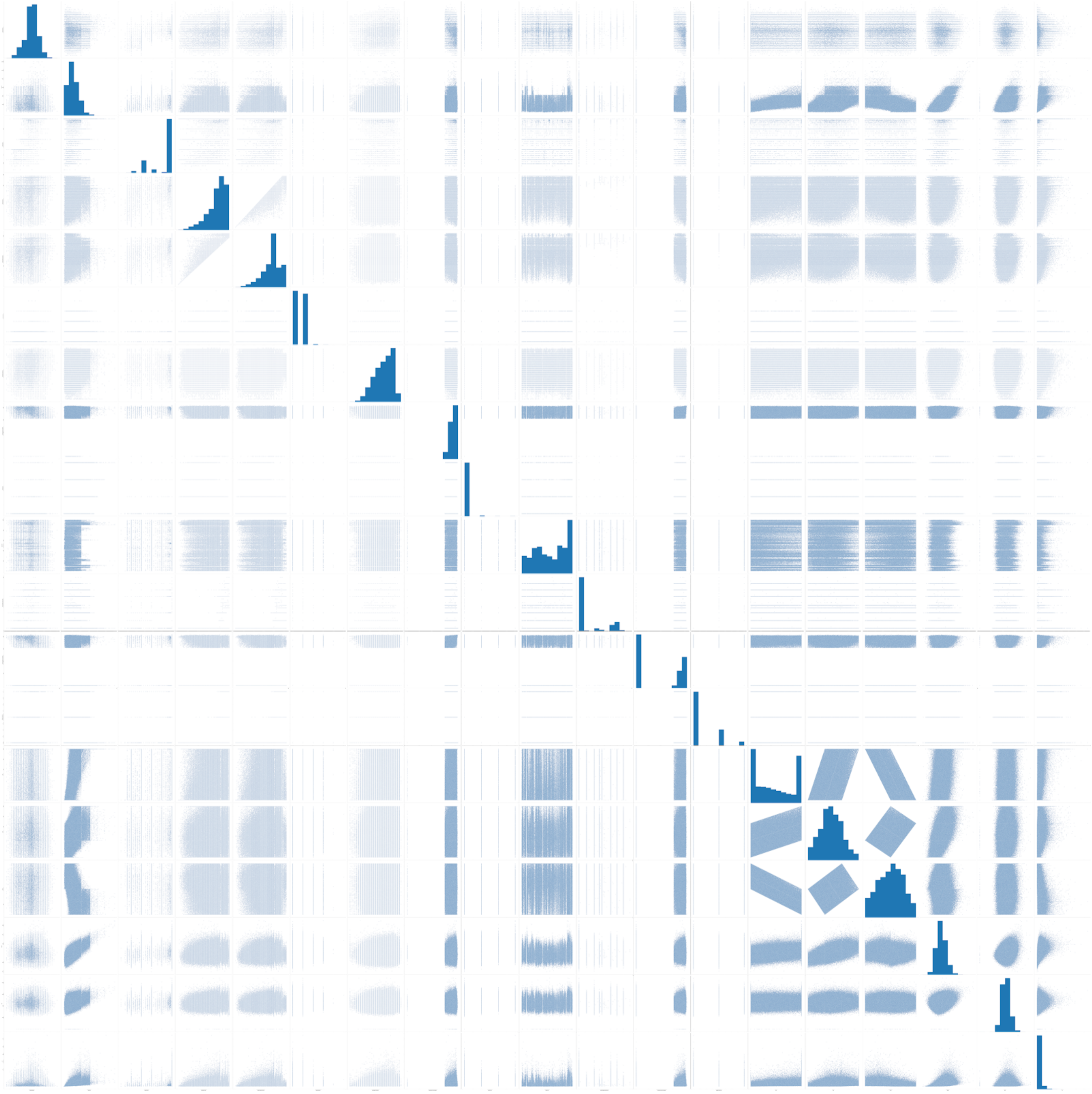




	original_interest_rate	original_upb	original_loan_term	original_loan_to_value	original_combined_loan_to_value	number_of_borrowers	original_debt_to_income_ratio	borrower_credit_score_at_origination	number_of_units	zip_code_short	primary_mortgage_insurance_percent	co_borrower_credit_score_at_origination	mortgage_insurance_type	fastk	fastd	stochrsi	macdhist	dema	roc
original_interest_rate	1.00	-0.02	0.67	0.28	0.28	-0.08	0.19	-0.29	0.10	0.06	0.17	-0.10	0.18	-0.03	-0.02	0.03	-0.02	-0.01	-0.01
original_upb	-0.02	1.00	0.19	0.09	0.10	0.14	0.11	0.05	0.07	0.15	0.03	0.14	0.05	0.80	0.48	-0.63	0.74	0.44	0.56
original_loan_term	0.67	0.19	1.00	0.33	0.33	-0.06	0.14	-0.06	0.02	0.05	0.26	-0.07	0.21	0.16	0.10	-0.12	0.14	0.08	0.10
original_loan_to_value	0.28	0.09	0.33	1.00	0.98	-0.05	0.10	-0.09	-0.07	-0.09	0.65	-0.06	0.60	0.09	0.06	-0.07	0.06	0.04	0.05
original_combined_loan_to_value	0.28	0.10	0.33	0.98	1.00	-0.05	0.10	-0.10	-0.07	-0.09	0.63	-0.06	0.59	0.11	0.06	-0.08	0.08	0.05	0.06
number_of_borrowers	-0.08	0.14	-0.06	-0.05	-0.05	1.00	-0.09	0.10	-0.01	0.03	-0.04	0.98	-0.05	0.12	0.07	-0.10	0.10	0.06	0.08
original_debt_to_income_ratio	0.19	0.11	0.14	0.10	0.10	-0.09	1.00	-0.18	0.02	0.05	0.07	-0.11	0.06	0.09	0.05	-0.07	0.08	0.05	0.06
borrower_credit_score_at_origination	-0.29	0.05	-0.06	-0.09	-0.10	0.10	-0.18	1.00	0.01	-0.02	-0.05	0.14	-0.03	0.04	0.02	-0.03	0.04	0.02	0.03
number_of_units	0.10	0.07	0.02	-0.07	-0.07	-0.01	0.02	0.01	1.00	-0.01	-0.07	-0.01	-0.07	0.03	0.02	-0.02	0.05	0.03	0.04
zip_code_short	0.06	0.15	0.05	-0.09	-0.09	0.03	0.05	-0.02	-0.01	1.00	-0.07	0.03	-0.07	0.12	0.07	-0.09	0.11	0.06	0.08
primary_mortgage_insurance_percent	0.17	0.03	0.26	0.65	0.63	-0.04	0.07	-0.05	-0.07	-0.07	1.00	-0.05	0.88	0.04	0.02	-0.03	0.02	0.01	0.02
co_borrower_credit_score_at_origination	-0.10	0.14	-0.07	-0.06	-0.06	0.98	-0.11	0.14	-0.01	0.03	-0.05	1.00	-0.05	0.12	0.08	-0.10	0.10	0.06	0.08
mortgage_insurance_type	0.18	0.05	0.21	0.60	0.59	-0.05	0.06	-0.03	-0.07	-0.07	0.88	-0.05	1.00	0.06	0.03	-0.05	0.04	0.02	0.03
fastk	-0.03	0.80	0.16	0.09	0.11	0.12	0.09	0.04	0.03	0.12	0.04	0.12	0.06	1.00	0.54	-0.83	0.45	0.08	0.45
fastd	-0.02	0.48	0.10	0.06	0.06	0.07	0.05	0.02	0.02	0.07	0.02	0.08	0.03	0.54	1.00	0.01	0.58	0.13	0.26
stochrsi	0.03	-0.63	-0.12	-0.07	-0.08	-0.10	-0.07	-0.03	-0.02	-0.09	-0.03	-0.10	-0.05	-0.83	0.01	1.00	-0.15	-0.01	-0.36
macdhist	-0.02	0.74	0.14	0.06	0.08	0.10	0.08	0.04	0.05	0.11	0.02	0.10	0.04	0.45	0.58	-0.15	1.00	0.29	0.50
dema	-0.01	0.44	0.08	0.04	0.05	0.06	0.05	0.02	0.03	0.06	0.01	0.06	0.02	0.08	0.13	-0.01	0.29	1.00	0.16
roc	-0.01	0.56	0.10	0.05	0.06	0.08	0.06	0.03	0.04	0.08	0.02	0.08	0.03	0.45	0.26	-0.36	0.50	0.16	1.00



	loan_id	original_interest_rate	original_upb	original_loan_term	original_loan_to_value	original_combined_loan_to_value	number_of_borrowers	original_debt_to_income_ratio	borrower_credit_score_at_origination	number_of_units	zip_code_short	primary_mortgage_insurance_percent	co_borrower_credit_score_at_origination	mortgage_insurance_type
loan_id	1.00	0.00	-0.00	-0.00	0.00	0.00	0.00	-0.00	-0.00	-0.00	-0.00	0.00	0.00	0.00
original_interest_rate	0.00	1.00	-0.02	0.67	0.28	0.28	-0.08	0.19	-0.29	0.10	0.06	0.17	-0.10	0.18
original_upb	-0.00	-0.02	1.00	0.19	0.09	0.10	0.14	0.11	0.05	0.07	0.15	0.03	0.14	0.05
original_loan_term	-0.00	0.67	0.19	1.00	0.33	0.33	-0.06	0.14	-0.06	0.02	0.05	0.26	-0.07	0.21
original_loan_to_value	0.00	0.28	0.09	0.33	1.00	0.98	-0.05	0.10	-0.09	-0.07	-0.09	0.65	-0.06	0.60
original_combined_loan_to_value	0.00	0.28	0.10	0.33	0.98	1.00	-0.05	0.10	-0.10	-0.07	-0.09	0.63	-0.06	0.59
number_of_borrowers	0.00	-0.08	0.14	-0.06	-0.05	-0.05	1.00	-0.09	0.10	-0.01	0.03	-0.04	0.98	-0.05
original_debt_to_income_ratio	-0.00	0.19	0.11	0.14	0.10	0.10	-0.09	1.00	-0.18	0.02	0.05	0.07	-0.11	0.06
borrower_credit_score_at_origination	-0.00	-0.29	0.05	-0.06	-0.09	-0.10	0.10	-0.18	1.00	0.01	-0.02	-0.05	0.14	-0.03
number_of_units	-0.00	0.10	0.07	0.02	-0.07	-0.07	-0.01	0.02	0.01	1.00	-0.01	-0.07	-0.01	-0.07
zip_code_short	-0.00	0.06	0.15	0.05	-0.09	-0.09	0.03	0.05	-0.02	-0.01	1.00	-0.07	0.03	-0.07
primary_mortgage_insurance_percent	0.00	0.17	0.03	0.26	0.65	0.63	-0.04	0.07	-0.05	-0.07	-0.07	1.00	-0.05	0.88
co_borrower_credit_score_at_origination	0.00	-0.10	0.14	-0.07	-0.06	-0.06	0.98	-0.11	0.14	-0.01	0.03	-0.05	1.00	-0.05
mortgage_insurance_type	0.00	0.18	0.05	0.21	0.60	0.59	-0.05	0.06	-0.03	-0.07	-0.07	0.88	-0.05	1.00



Attempts at creating visualizations that operated from a remote server were initially unsuccessful. I initially attempted with HiddenLayer, a package that would create a graph depiction of the model and could assist with the creation of dynamic accuracy and loss. This package is focused to work with Jupyter notebooks. Setting up and operating Jupyter to work on an AWS deep learning instance is initially easy as it is pre-installed along with most frameworks (pytorch, cafe, etc.) but does present its own issues. By having a complex AMI structure, getting packages to install correctly into the correct anaconda environment took experimentation. Eventually the packages were installed correctly, test code was functionable, but I was unable to get it to properly work with our model. The core issues with deploying the visualizations persisted across several packages. Straight displaying also did not properly function.

Summary and Conclusion

The dataset is massive but contains many errors that will persist into the future as Fannie Mae relies on external data providers. Much more rigorous data management and transformations would be required to get the most out of the data. With the vast amount of potential training data, it would be possible to build a system that would best determine missing values. One method could be to identify persistent errors from vendors if they exist, and then examining correct records, to determine if there is a variance between them and a generic correct record. Or to examine if using the originator of the loan plays a larger role in the model by using that as a data import lens. That would reduce the size of the data within the model. A real world case would rely on a complex model that would inference on the corresponding originator model to determine probability of default.

My final conclusion is that going forward, it will be important for me to continue to pursue a more robust computer science skills focusing on programming. Luke Bogacz and Vishal Sinah are more advanced in their programming capabilities and it proved challenging keeping up with them in this group assignment. Both provided great code and my personal skills precluded matching their technical contributions to the success of this project.

Code Calculation

Percentage of non original code: 91.02%

Works Cited/References

Facebook, Pytorch, *https://pytorch.org*

Waleed, HiddenLayer, *https://github.com/waleedka/hiddenlayer*

Sergey Zagoruyko, TNT, *https://github.com/pytorch/tnt*