CS273P: MACHINE LEARNING AND DATA MINING
FINAL PROJECT: ADULT DATASET

Harry Pham (#16168220), Keith Tachibana (#69125572) → *TEAM TEST*

OVERVIEW

1. Introduction

To begin, we started by using the data loader that was provided to us in order to get a "feel" for the distribution of the data that eventually led to our decision to exclude certain features. What this report will attempt to do is explain correlations between some of the data features by first going over the feature analysis, then our implementation of the base model, and the various classification models we used, including logistic regression, k-nearest neighbor (kNN), decision trees, random forest, and gradient boosting. We also did neural networks as an experiment. This necessitated tuning of certain hyperparameters, which ended up giving us an accuracy of 87.3% from our best model.

2. Libraries Used

We used matplotlib, numpy, scipy, keras, seaborn, scikit-learn, and pandas. We used scipy for the chi-squared contingency (see Appendix A), keras for neural networks (see Appendix B), seaborn to make feature visualization easier, scikit-learn for all our models (besides neural networks), ROC AUC, the confusion matrix, and simple imputer (see Appendix D), and pandas for manipulating the data frames, finding missing values, and column statistics.

ANALYSIS

3. Feature Analysis (please refer to Appendix A):

Upon loading the data, we were presented with a training data set containing approximately 29,315 records with binomial labels indicating a salary less than 50k or greater than 50k; the validation data set contained 3,246 records and the test data set contained 16,281 records (all data was loaded using the data_loader script provided to us).
The following is a description of the various features examined:

age: *continuous*
workclass: *Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.*
fnlwgt: *continuous*.
education: *Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.*
education-num: *continuous.*
marital-status: *Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.*
occupation: *Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.*
relationship*: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.*
race: *White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.*
sex: *Female, Male.*
capital-gain: *continuous.*
capital-loss: *continuous.*
hours-per-week: *continuous.*

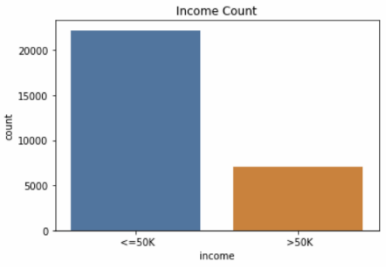
native-country: *United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.*

Total number of numeric features = 6
Total number of categorical features = 8

	age	workclass	fnlwgt	education	education.num	martial.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week	native.country	income
count	29315.000000	27653	2.931500e+04	29315	29315.000000	29315	27646	29315	29315	29315	29315.000000	29315.000000	29315.000000	28817	29315
unique	NaN	8	NaN	16	NaN	7	14	6	5	2	NaN	NaN	NaN	41	2
top	NaN	Private	NaN	HS-grad	NaN	Married-civ-spouse	Prof-specialty	Husband	White	Male	NaN	NaN	NaN	United-States	<=50K
freq	NaN	20424	NaN	9516	NaN	13469	3714	11853	25039	19604	NaN	NaN	NaN	26277	22229
mean	38.590756	NaN	1.894121e+05	NaN	10.071295	NaN	NaN	NaN	NaN	NaN	1094.195941	87.954801	40.443084	NaN	NaN
std	13.634218	NaN	1.057128e+05	NaN	2.576820	NaN	NaN	NaN	NaN	NaN	7458.664789	404.127059	12.311993	NaN	NaN
min	17.000000	NaN	1.376900e+04	NaN	1.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	1.000000	NaN	NaN
25%	28.000000	NaN	1.175685e+05	NaN	9.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	40.000000	NaN	NaN
50%	37.000000	NaN	1.779550e+05	NaN	10.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	40.000000	NaN	NaN
75%	48.000000	NaN	2.368395e+05	NaN	12.000000	NaN	NaN	NaN	NaN	NaN	0.000000	0.000000	45.000000	NaN	NaN
max	90.000000	NaN	1.484705e+06	NaN	16.000000	NaN	NaN	NaN	NaN	NaN	99999.000000	4356.000000	99.000000	NaN	NaN

Figure 1.1: Here we can see that the data is skewed toward making under 50k - 22,229 out of the 29,315 records were those that made under 50k, which represented 76% of the data

To further support our claim that the data was skewed toward those making under 50k versus those making greater than 50k, we were able to produce the following count plot to the right:



Other columns that were dominated by other features were work class (the majority being private), race (the majority being white), native country (the majority being United States), and capital gain and capital loss (the majority being zero). Due to the fact the majority of the values for capital gain and loss were zero, we combined both features into a feature called “Capital Gain Loss” which has the equation “capital gain – capital loss”. We felt justified in this decision because it still preserved the nature of the value, whether it was positive or negative, such that if the individual made capital gains, then his or her value was still positive based on our equation, and if they endured capital losses, then his or her value for that feature was in the negatives.

```
age                0
workclass       1662
fnlwgt             0
education          0
education.num      0
martial.status     0
occupation      1669
relationship       0
race               0
sex                0
capital.gain       0
capital.loss       0
hours.per.week     0
native.country   498
income             0
dtype: int64
```

We searched through each column to see which features were missing from the data set, and discovered that the missing features were represented by a question mark. The features that were missing included 1,662 work class labels, 1,669 occupation labels, 198 native country labels as evidenced in the following screenshot to the left. In order to deal with the missing data, we did a simple technique called imputation which just replaces the question marks with the most frequent value for that column.

Additionally, we found that education.num, age, hours per week, capital gain, and capital loss were positively correlated with income, as you can see in the heat map to the left. Ultimately, we decided to remove fnlwgt from the training set because it had little correlation with income, which is also revealed by the heat map. Other relationships that were of significance were education.num and education with a 1.000 correlation due to the fact that one was the numerical representation of the other (see Appendix E).

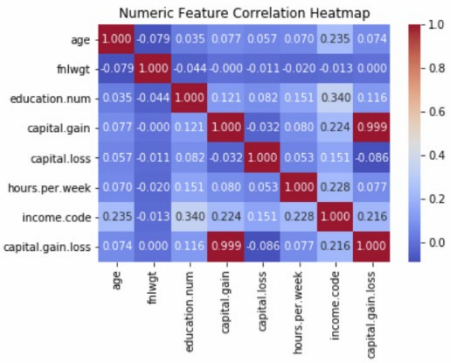
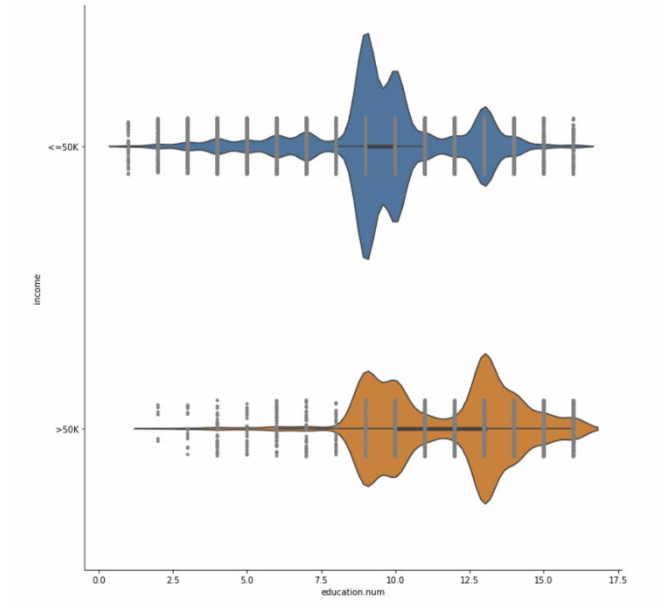
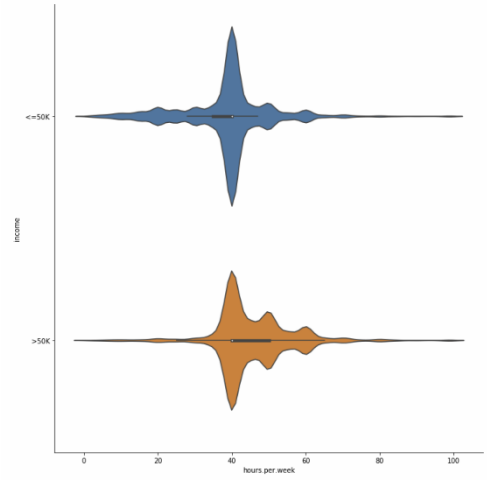


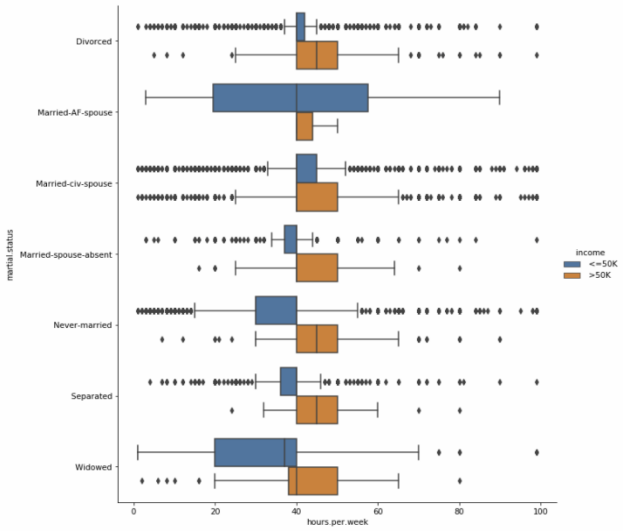
Figure 1.2: As this heat map shows, capital gain and capital loss have a tightly knit relationship, with a 0.999 correlation which led to our decision to combine the two features into one

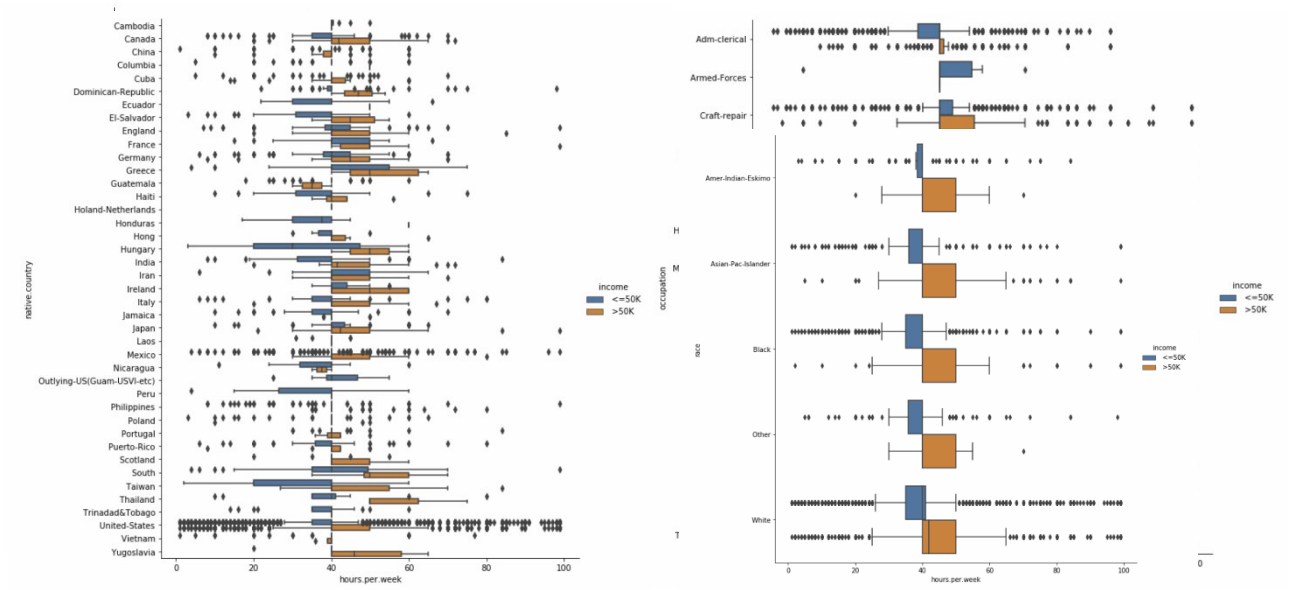


Another important relationship was that between education and income, with those that made over 50k having a higher education compared to those making under 50k overall. As you can see in the screenshot on the left, the tail end of the orange curve has a maximum value that is higher than that of the blue curve, indicating higher education, as well as the maximum height of the orange curve being greater overall than that of its blue counterpart, which again indicates greater rates of higher education overall.

We noticed that hours per week ended up being a good separator amongst the majority of the features, which we based on many of the cat plots that we produced, including (but not limited to) hours per week versus work class, hours per week versus marital status, hours per week versus relationship, hours per week versus occupation, hours per week versus native country, and hours per week versus race, all of which you can see in the following six screen shots by focusing on the divide between the orange and blue blocks:







We discovered that marital status and relationship were strongly correlated because the p-value was less than 0.001 (represented as 0.0 using scipy), which means that these features are not independent from one another, which led us to remove one of them, as you can see in the following table:

martial.status relationship	Divorced	Married-AF-spouse	Married-civ-spouse	\
Husband	0	9	11844	
Not-in-family	2179	0	17	
Other-relative	95	1	113	
Own-child	304	0	91	
Unmarried	1436	0	0	
Wife	0	10	1404	

martial.status relationship	Married-spouse-absent	Never-married	Separated	Widowed
Husband	0	0	0	0
Not-in-family	189	4221	377	497
Other-relative	31	542	50	45
Own-child	40	4038	85	13
Unmarried	115	800	414	355
Wife	0	0	0	0

Chi2: 34867.16845517329 P: 0.0 DOF: 30

We did one hot encoding for the categorical features to convert our categories to numerical values because our models could only work with numerical values (see Appendix C). We also converted our class labels from ≤ 50k and > 50k to 0 and 1, respectively.

So in summary, the features we dropped were fnlwgt, education, and relationship based on the justifications that we elaborated above.

4. Base Model (please refer to Appendix B):

We used three base models: one was guessing all 0's (≤ 50k), another was logistic regression, and the last one was kNN.

Metric	All Under 50K Baseline	Logistic Regression	kNN
AUC	0.5	0.6250004652510627	0.6097237472118169
Accuracy	0.7674060382008626	0.8012939001848429	0.7686383240911892

This was done with all default parameters with no feature selection. The reason we used AUC as our loss function instead of accuracy was because our data was so skewed toward those making under 50k. As you can see, our AUC for our baseline model for all under 50k was 0.5 and our accuracy was pretty high at 76.7%, even though we were just guessing for the entire thing using all zero's. The reason we chose all under 50k baseline as our base model is because we could get a

pretty high accuracy just by guessing one of the class labels, which we could use to compare against the models tested in the next section.

5. Models Tested

- Logistic Regression (LR): This was included because it was one of our baseline models and we wanted to see how it would fare against our feature selection.
- k-Nearest Neighbors (kNN): Likewise, this was also included due to it being one of our baseline models and we similarly wanted to see how it would do against our feature selection.
- Decision Tree (DT): We chose this as one of our classifiers to test because it usually has good accuracy but is at risk at overfitting the data.
- Random Forest (RF): This was chosen to counteract decision trees due to the latter overfitting the data because it uses the ensemble technique "bagging" to reduce the variance and help against overfitting the data.
- Gradient Boosting (GB): We did this one because the class labels were skewed toward one side such that it would probably need help to learn from the errors from the data bias and the sequential models from boosting will learn on getting the >50k labels correct.
- Neural Networks (NN): We learned in class that this would be a good classifier and wanted to experiment around with it.

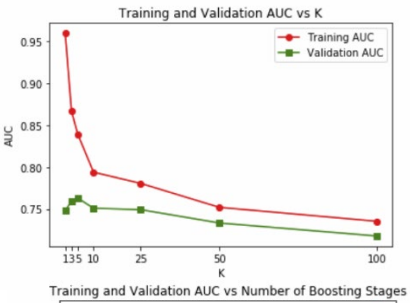
6. Hyperparameters (please refer to appendices C, D, E, and F):

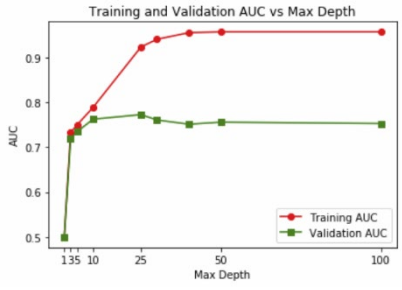
The area under curve (AUC) before hyperparameter tuning is given by the following table:

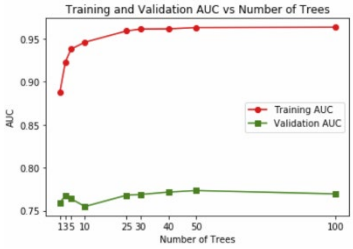
Metric	LR	kNN	DT	RF	GB	NN
AUC	0.7562438021911996	0.7629646861150472	0.7479104910126788	0.7616811248973124	0.7683267710778671	0.6318590634894

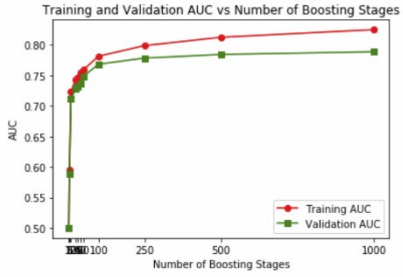
As you can see, our feature selection has improved our AUC even without hyperparameter tuning, which shows that we have indeed removed irrelevant or duplicate features.

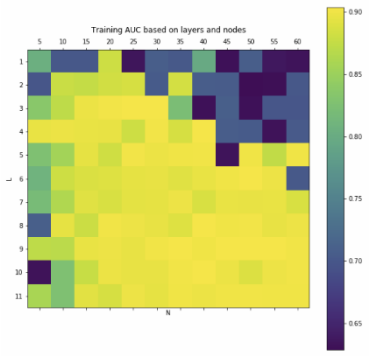
We ended up varying the values of K for kNN, max depth for decision trees, number of trees for random forest, and number of boosting stages for gradient boosting, versus AUC for both the training and validation data sets, and the following four plots were a result of that:











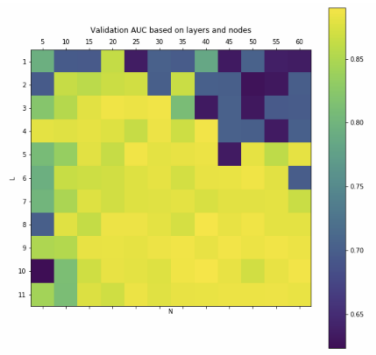
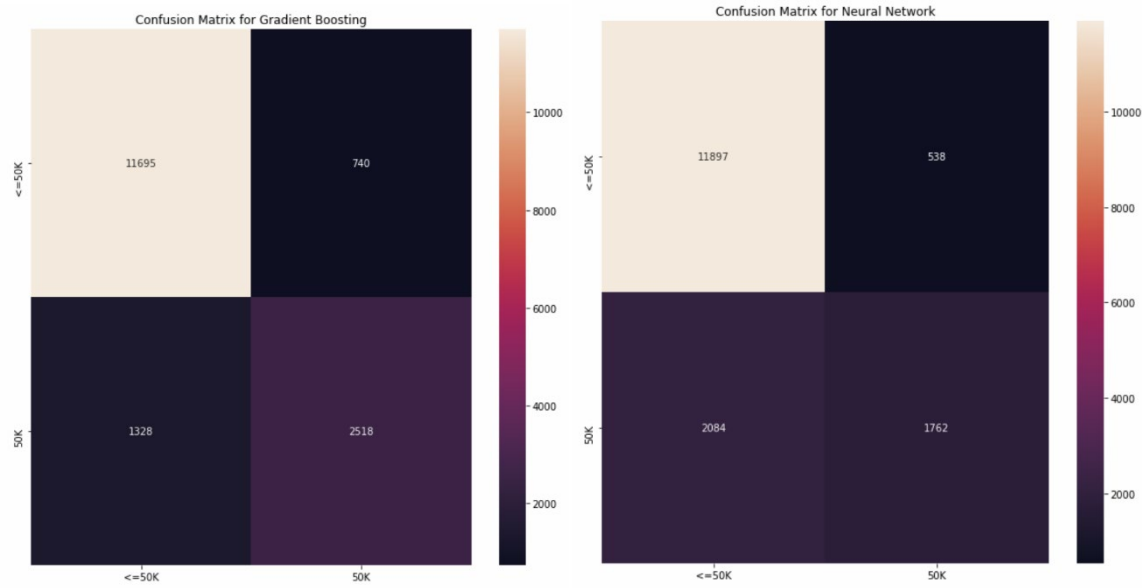


Figure 6.2: Hyperparameter tuning for (starting from top left to right going down) kNN, decision trees, random forest, gradient boosting, and neural networks (training and validation data sets)

The final hyperparameters we settled on were K = 5 for kNN, max depth = 25 for decision trees, number of trees = 50 for random forest, number of boosting stages = 1,000 for gradient boosting, and number of layers = 10 and number of nodes = 40 for neural networks.

The final AUC and accuracy (ACC) values are represented in the table below:

Metric	LR	kNN	DT	RF	GB	NN
Training AUC	0.7641522542652434	0.8390930055507156	0.9232337079612396	0.9627315118931064	0.8249969269533672	0.9042306110184234
Training ACC	0.8510318949334334	0.8914207743476036	0.9494797885041788	0.9753368582636875	0.8900903974074705	0.840457103871738
Validation AUC	0.7562438021911996	0.7629646861150472	0.7711903249047564	0.7693072544604285	0.7886749915590164	0.7022231025067728
Validation ACC	0.8468884781269255	0.8394947627849662	0.8308687615526802	0.8478126925446704	0.866913123844732	0.8376463339494763
Test AUC	0.7575874735833824	0.7737198173089771	0.7510543437419042	0.7645351773057654	0.7975983695560127	0.7074366738240097
Test ACC	0.8501320557705301	0.8458325655672256	0.8239665868189915	0.8467538848965052	0.8729807751366624	0.8389533812419384



7. Conclusion

Our AUC has increased after we did our hyperparameter tuning. We can see that gradient boosting has the best performance out of all our models, which is because it improved on getting the greater than 50k class correct. We believe our results for decision tree and random forests are still overfitting because of high training AUC and accuracy values. The reason why our neural network seems to not have done as well as we expected it to is because it still overvalues the less than 50k label, which increases the error for the greater than 50k label. This can be seen in its confusion matrix in comparison with the confusion matrix for gradient boosting. The gradient boosting confusion matrix shows that it has reduced its errors on the less than 50k label, which proves it has focused on getting those labels more correct.

8. Who Did What

We both worked together on the feature selection and model analysis. Harry worked on the neural networks experimentation and Keith worked on the report with input from Harry.

APPENDIX

Appendix A: FeatureAnalysis.ipynb
Appendix B: BaseModels.ipynb
Appendix C: ModelHyperparameterTesting.ipynb
Appendix D: NNHyperparameterTesting.ipynb
Appendix E: ModelFinal.ipynb
Appendix F: NNFinal.ipynb

REFERENCES

Chi Square: https://codereview.stackexchange.com/a/108842
Keras: https://towardsdatascience.com/how-to-build-a-neural-network-with-keras-e8faa33d0ae4
One Hot Encoding: http://fastml.com/converting-categorical-data-into-numbers-with-pandas-and-scikit-learn/
Simple Imputation: https://scikit-learn.org/stable/modules/impute.html
Correlation Matrix: https://stackoverflow.com/questions/29432629/plot-correlation-matrix-using-pandas