Results

Training the Model

After reading in the data df.skew()was used to determine the skewness values of the numerical variables. Skewness is the measure of asymmetry of a random variable's probability function about its mean. If a variable's skewness is outside of the range [-1, 1], then the distribution is highly skewed.

```
age              0.530228
fnlwgt           1.459220
education_num   -0.305379
capital_gain    11.902682
capital_loss     4.526380
hr_per_week      0.330869
```

The skewness values for *capital_gain* and *capital_loss* show that their distributions are significantly skewed. Applying a log transformation results in the following:

```
age              0.530228
education_num   -0.305379
capital_gain     3.073208
capital_loss     4.272387
hr_per_week      0.330869
```

The skewness of *capital_gain* is significantly reduced, and the skewness of *capital_loss* is reduced, if minimally. Afterwards, all numerical features were normalized.

	age	education_num	capital_gain	capital_loss	hr_per_week
0	0.383562	0.266667	0.0	0.0	0.397959
1	0.150685	0.800000	0.0	0.0	0.500000
2	0.287671	0.866667	0.0	0.0	0.479592
3	0.561644	0.133333	0.0	0.0	0.397959
4	0.150685	0.600000	0.0	0.0	0.397959

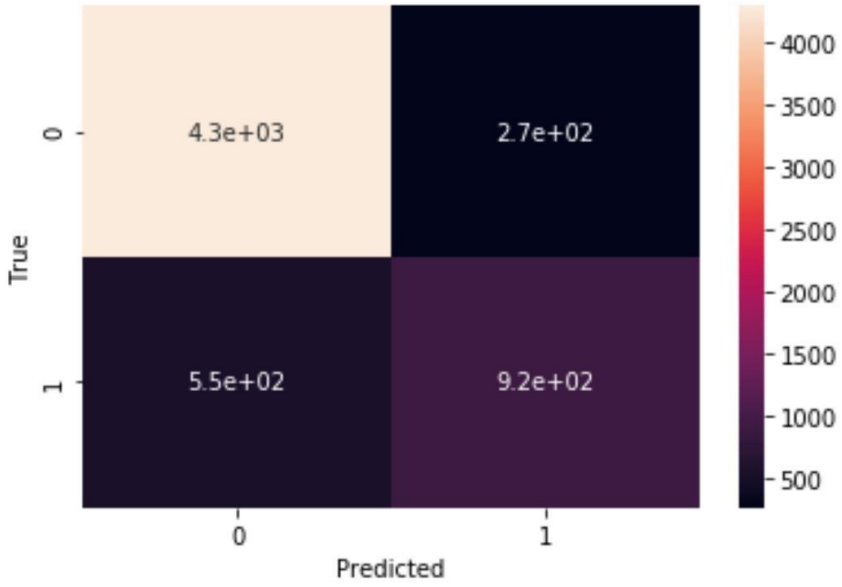
Accuracy The data was set split into 80% training set and 20% test set. Logistic regression with L2 regularization was used. The following are screenshots showing the training, validation, and accuracy scores from various runs:

```
Training Score: 0.9140867835384807
Validation Score: 0.8715398640808885
Accuracy Score: 0.8715398640808885
```

```
Training Score: 0.9214223548427204
Validation Score: 0.8727001491795127
Accuracy Score: 0.8727001491795127
```

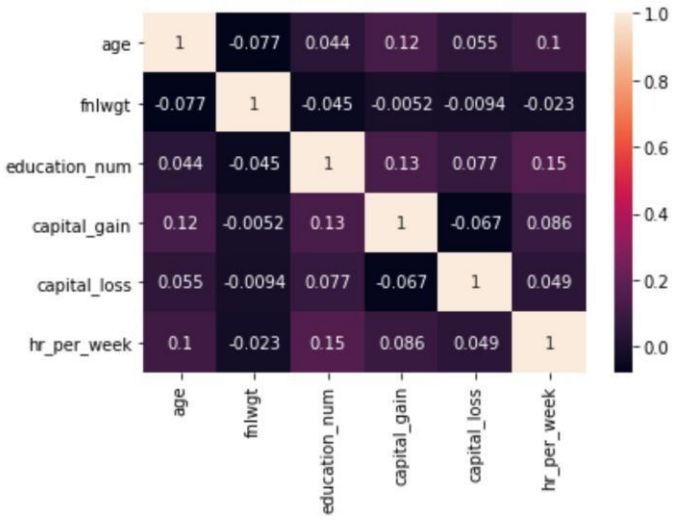
```
Training Score: 0.9147084421235857
Validation Score: 0.8683905188131941
Accuracy Score: 0.8683905188131941
```

The confusion matrix:

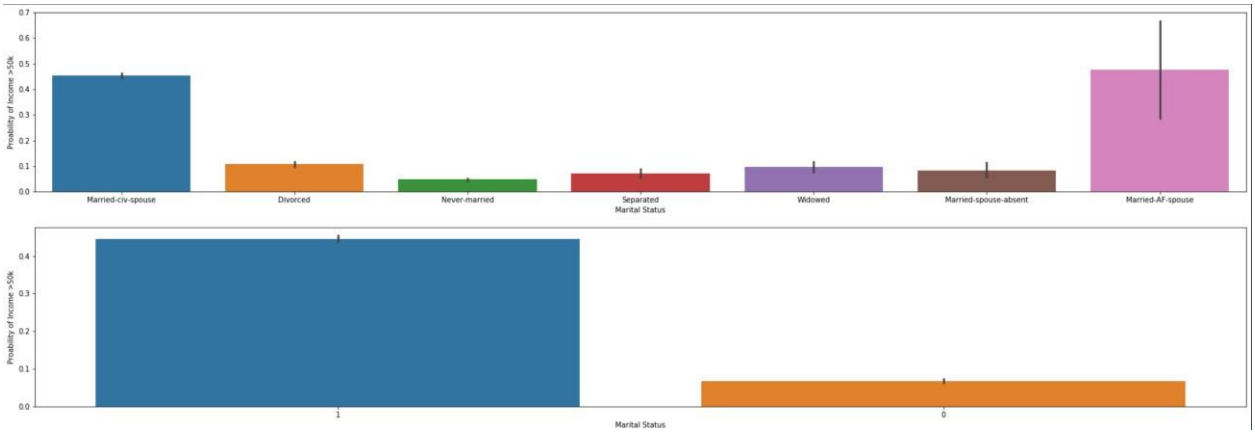


Extracting Better Features

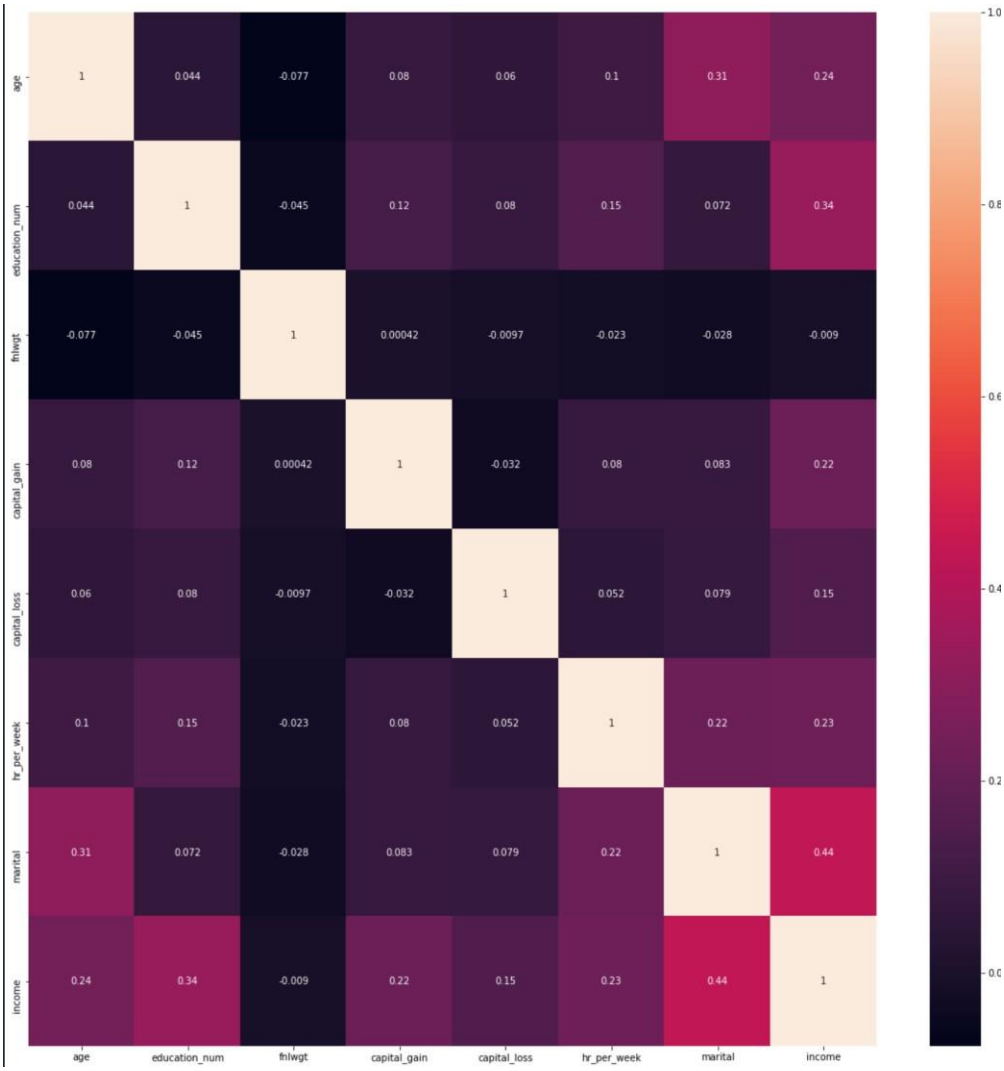
The correlation matrix below shows little to negligible correlation between all of the continuous features. This shows that a multivariate regression model can be attempted because the predictor variables are relatively independent from each other.



Looking at the raw data, there seem to be some redundant features. The features *relationship* and *marital_status* overlap significantly; *relationship,* for instance, contains categories for wife, husband, and unmarried, which are also covered by *marital_status*. Thus, one can discard one of these features and remove redundant information from the model. I will look at *marital_status* first. I would like to see if I can transform this into a numeric value because the attributes *Married-AF-spouse* and *Married-civ-spouse* have similar probabilities.



The barplot above demonstrates that people who are married have a significantly higher change of having an income over 50,000 (over 40%) whereas those who are single have approximately an 8% chance of earing over 50,000. Since *income* and *marital_status* have been transformed into numeric values, let us look at all numeric features and their correlation to income



The five features that contribute most to income, in descending order, are: marital, education_num, age, hr_per_week, and capital_gain. The feature *fnlwgt* shows the least correlation to income of all features by a factor of 16.667. It not will not be considered as a feature in the final model.

For the final model, the features that were removed were *fnlwgt, education, relationship* and *occupation*. *Education* is redundant considering that *education_num* quantifies the amount of years one has spent on their education. *Relationship* was removed because of the redundant data it shared with *marital. Occupation* was removed because *type_employer* was a similar, but more general feature

Tuning

Constant and adaptive learning rates gave the best probabilities.

```
Accuracy of Final Model, optimal learning rate :  0.8577821979114868
Accuracy of Final Model, adaptive learning rate :  0.8606000331510029
Accuracy of Final Model, constant learning rate :  0.8610972981932703
```