Ryan Aclan

Principles of Data Science

April 24th, 2022

Final Report

Predicting Income for Filipino Households

**Introduction**

The motivation behind this project is to try different machine learning models and find which would performs best when trying to predict household incomes in the Philippines. Being able to predict household incomes can help in identifying areas and statuses of wealth and also assist in describing and classifying the quality of life based on the attributes of a house and its members. It can also reveal how different levels of living, spending patterns, and family structure reflect the overall household income.

The data is taken from Family Income and Expenditure Survey (FIES) issued by the Philippine Statistics Authority (PSA). It consists of many features ranging from various types of expenditures to the number of washing machines a household has. Because the goal is to predict a continuous variable, this is a predictive regression problem.

In this paper, I go over my comparison of different machine learning models and exercise in principal component analysis.

**Approach**

The entire project was done using Python across two Jupyter Notebooks to allow for investigating and training multiple learning models at once.

In this dataset, there are 60 columns with 41,545 records. This means there are 59 features. 15 of these features are categorical variables with 14 of them being nominal and 1 of them being ordinal. I handled the nominal variables by converting them into numerical values via label encoding and then used one-hot encoding, which resulted in a new dataframe with 511 columns. There were 2 categorical variables that contained missing values. For this, I labeled those values as “NULL” and let the label encoder handle them as separate values.

I ended up dropping the ordinal variable, which was “Household Head Highest Grade Completed”. This was because the progression of the order was unclear and therefore could be misrepresented when trying to use ordinal encoding or one-hot encoding.

Once the data was preprocessed, I moved on to making models to predict the household income. For all models, I used a train-test split for 90/10 and normalized the data with the StandardScaler from sklearn.

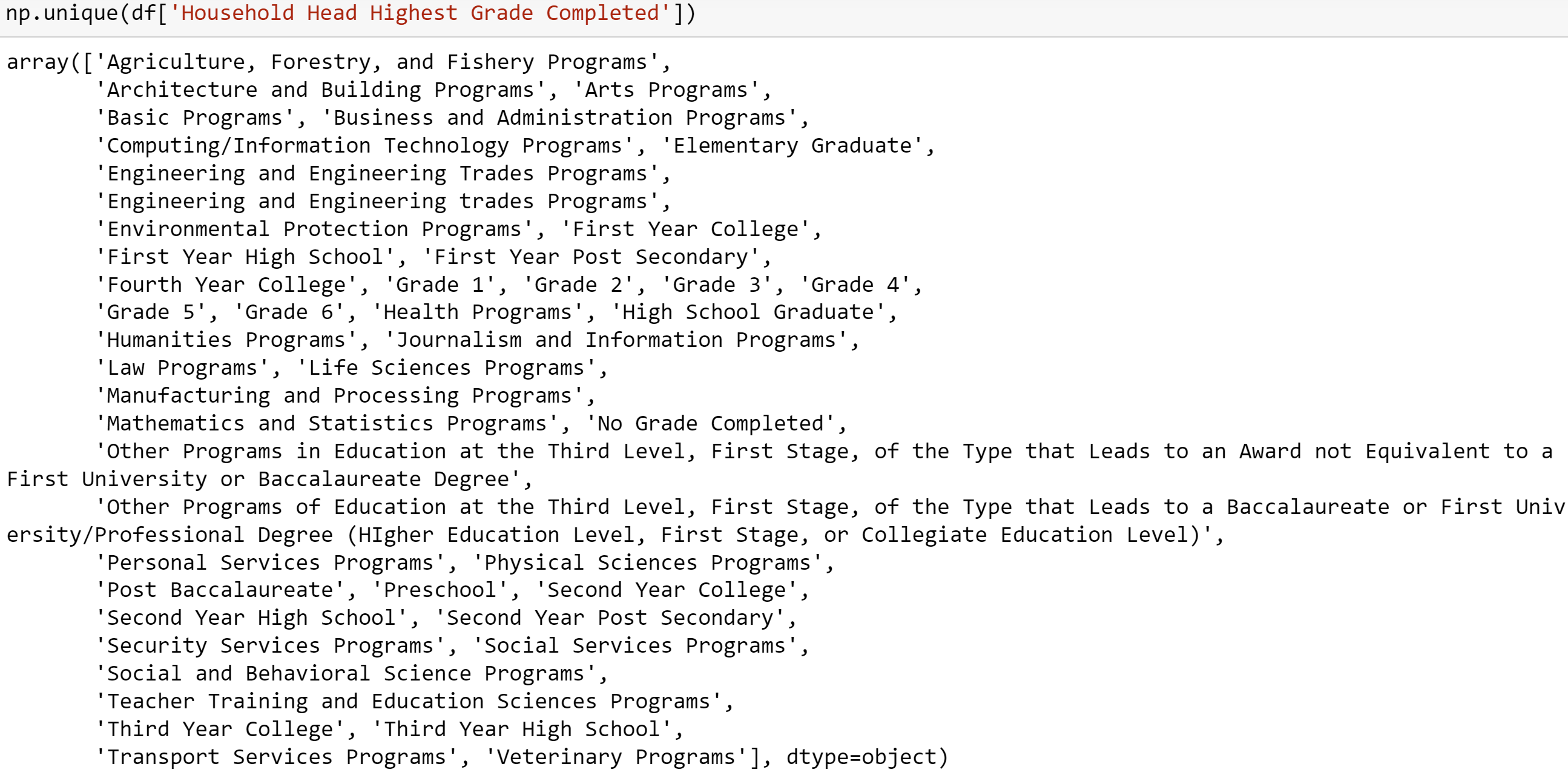
First, I tried a linear regression model without the use of principal components and recorded its performance (MAE, MSE, RMSE, and R^2 score). Then, I ran the model multiple times through every possible number of principal components being used (ranging from 2 principal components all the way to 510) in an attempt to find what number of principal components the model performed best with. Once found, the model’s performance with a training set and testing set was recorded.

Once I had tried a linear regression model, I did a preliminary test of four other models: ridge regression, lasso regression, linear support vector regression, and decision tree. Out of these four, ridge regression and lasso regression performed the best, though I also did more investigating with the decision tree since it performed fair as well. Linear support vector regression was the only one that was dropped due to its poor performance.

For the ridge and lasso regression models, I needed to determine the best alpha value to use, so I tried every power of 10 from 0.01 to 1000000. I found that the best alpha value was 1000, so I used that alpha value when conducting the rest of the project.

I repeated the process I did with the linear regression model with the three other models and recorded their performances.

**Results**

The image below shows all the different classes for “Household Head Highest Grade Completed”.

The table below shows the R^2 scores for the five learned models I thought to investigate.

|  |  |
| --- | --- |
| **Model** | **R^2 Score** |
| Linear Regression | -13711366053578848256.0000 |
| Ridge Regression (alpha=1.0) | 0.6863 |
| Lasso Regression (alpha=1.0) | 0.6863 |
| Linear Support Vector Regression | -0.3926 |
| Decision Tree | 0.5746 |

The table below shows the R^2 scores for the ridge and lasso regression models given different values for alpha.

|  |  |  |
| --- | --- | --- |
| **Alpha Value** | **Ridge** | **Lasso** |
| 0.01 | 0.68627 | 0.68627 |
| 0.1 | 0.68627 | 0.68627 |
| 1.0 | 0.68627 | 0.68627 |
| 10.0 | 0.68629 | 0.68631 |
| 100.0 | 0.68647 | 0.68661 |
| 1000.0 | 0.68781 | 0.68750 |
| 10000.0 | 0.68751 | 0.67584 |
| 100000.0 | 0.61063 | 0.44554 |
| 1000000.0 | 0.29796 | -0.00112 |

The table below shows the performance of the four learning models without principal component analysis.

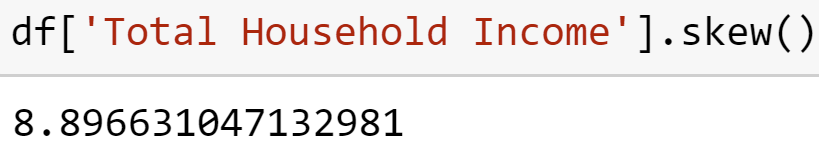
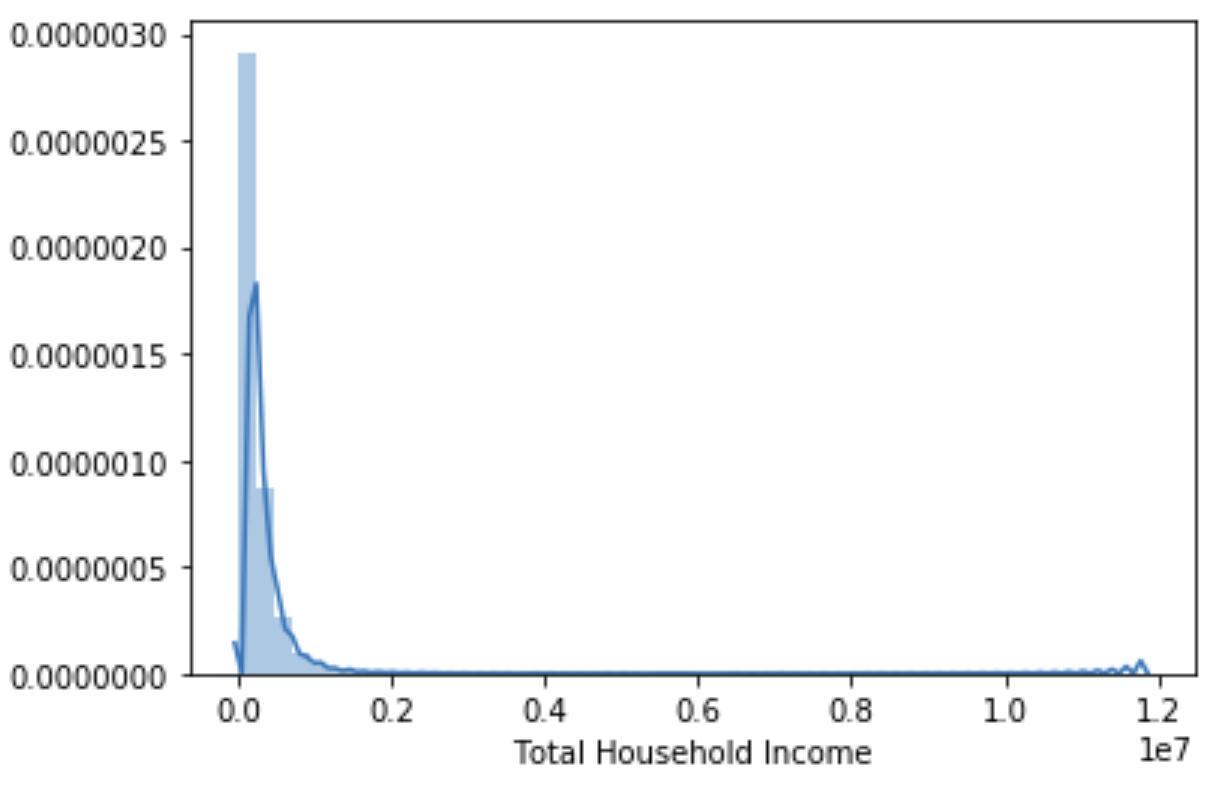
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **MAE** | | **MSE** | | **RMSE** | |  |
| **Model** | **Train** | **Test** | **Train** | **Test** | **Train** | **Test** | **R^2** |
| Linear | 49383.3048 | 27065045342562.77 | 10607974493.6372 | 1.614e+30 | 102995.0217 | 1270571182375245.8 | -1.3711e+19 |
| Ridge | 48945.2965 | 54936.0956 | 10629214293.2621 | 36756948219.9654 | 103098.0809 | 191721.0166 | 0.6878 |
| Lasso | 48962.1312 | 54373.9914 | 10797434932.3310 | 36793081654.4593 | 103910.7065 | 191815.2279 | 0.6875 |
| Decision Tree | 0.0 | 77416.6156 | 0.0 | 50728031861.9610 | 0.0 | 225228.8433 | 0.5746 |

The table below shows the performance of the four learning models with principal component analysis.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **MAE** | | **MSE** | | **RMSE** | |  |
| **Model** | **# of PCs** | **Train** | **Test** | **Train** | **Test** | **Train** | **Test** | **R^2** |
| Linear | 220 | 49864.4688 | 55341.2991 | 10818551894.6422 | 36743522579.7945 | 104012.2680 | 191686.0000 | 0.6879 |
| Ridge | 221 | 49468.4397 | 55108.6317 | 10859247210.7758 | 36987854818.5800 | 104207.7119 | 192322.2681 | 0.6858 |
| Lasso | 197 | 49051.0555 | 54532.9335 | 10992497824.3558 | 36957611814.8788 | 104845.1135 | 192243.6262 | 0.6861 |
| Decision Tree | 234 | 0.0 | 81156.4833 | 0.0 | 55639906801.0700 | 0.0 | 235881.1285 | 0.5274 |

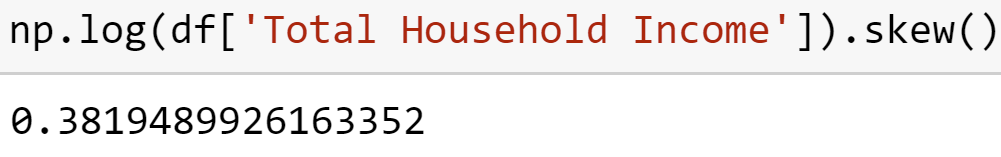
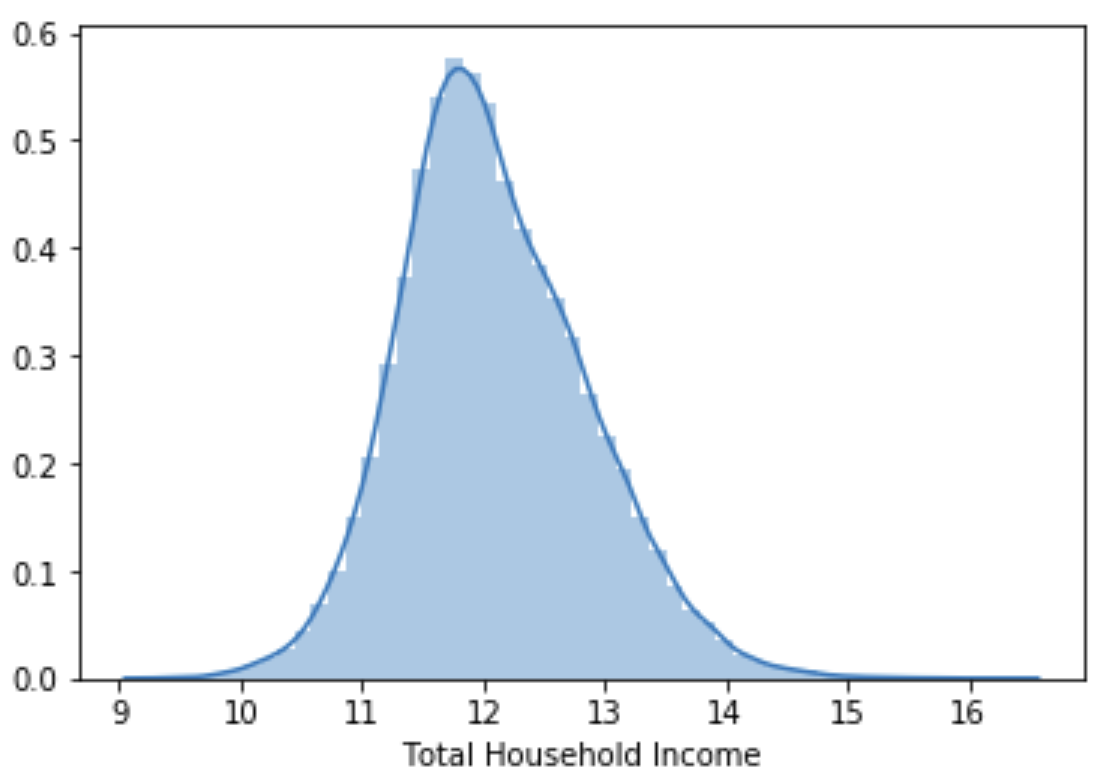
In the end, it appears that linear regression with 220 principal components performed the best when trying to predict the incomes of Filipino households.

Some results worth noting include the fact the ridge regression model, the lasso regression model, and the decision tree all performed slightly worse with principal components than without.

It’s also worth noting that the household incomes were incredibly right-skewed, as shown below.

A few considerations include investigating the linear support vector regressor despite its poor initial R^2 score. I also considered trying to make the data less right-skewed.

One of these methods consisted of taking the log of the household incomes, yielding in the distribution shown below.



Another one of these methods consisted of dropping all records with incomes that were considered outliers; however, because the data was so right-skewed, this resulted in a significant number of records being dropped.

**Conclusion**

Despite having a worse performance without principal components, the linear regression learning model with 220 principal components performed the best. Some limitations of this result are that the data was incredibly right-skewed and that the linear support vector regressor was not investigated.

**Acknowledgements**

* <https://www.kaggle.com/datasets/grosvenpaul/family-income-and-expenditure>
* <https://towardsdatascience.com/top-3-methods-for-handling-skewed-data-1334e0debf45>
* <https://www.youtube.com/watch?v=2JwDkvWEVlM>