Prediction Model for Female-headed households in South Africa

Shreyas Iyer, Tanush Sharanarthi

Executive Summary:

Business Problem: The business objective is to build a predictive model that accurately estimates the percentage of households per ward in South Africa that are female-headed and living below a particular income threshold, by using data points that can be collected through other means without an intensive household survey. This solution can potentially reduce the cost and improve the accuracy of monitoring key population indicators such as female household headship and income level in between census years.

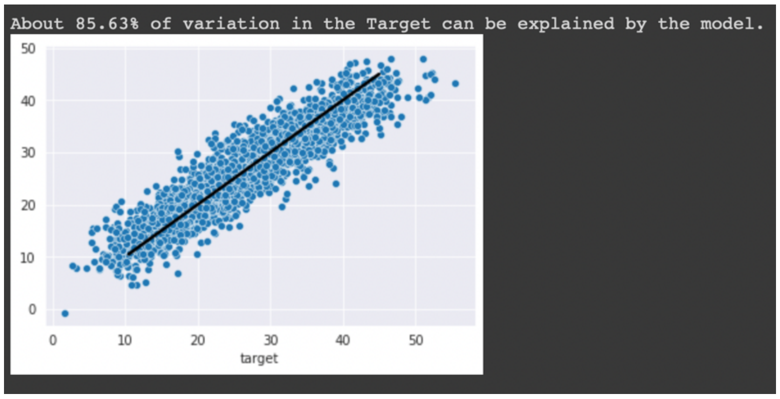
MSE will be the primary aggregate performance metric and RMSE will be the secondary performance metric, since female-headed households tend to face greater social and economic challenges and, in general, are more vulnerable to lower household incomes and higher rates of poverty. Therefore, large errors are more important than smaller prediction errors.

The model is trained on 2822 wards and tested on 1013 wards. The target variable is the percentage of households per ward that are both female-headed and earn an annual income that is below R19,600 (approximately $2,300 USD in 2011).

The models we use are Linear Regression, Random Forest Regressor, Stacked Regressor and Keras Regressor. Since RandomForest Regressor performed the best out of the models, we use it to make our predictions on the test data and we obtain a MSE value of 3.7402.

Analysis:

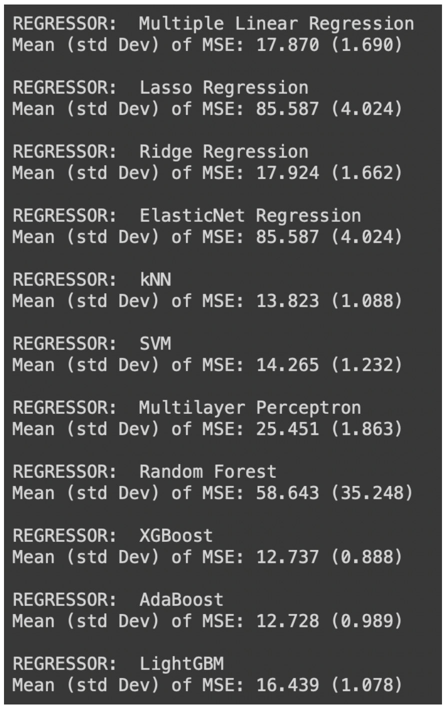
In order to prepare the data for these models we checked the shape of the data and description as well. We found that there were no duplicates or null values in the data. There was however skewness in the features of the data which we had to perform skewness correction on. This was followed by feature scaling using the RobustScaler library. This is because our data was quite skewed and so it would tend to have more outliers than a normal dataset. We also did feature selection and dropped 5 features that had a low correlation with the target. We then created polynomial and interaction features to derive new combination of features that might be useful to our model and selected the top 20 of them. To address the outliers in the data we used the IsolationForest library.



```
About 85.63% of variation in the Target can be explained by the model.
```

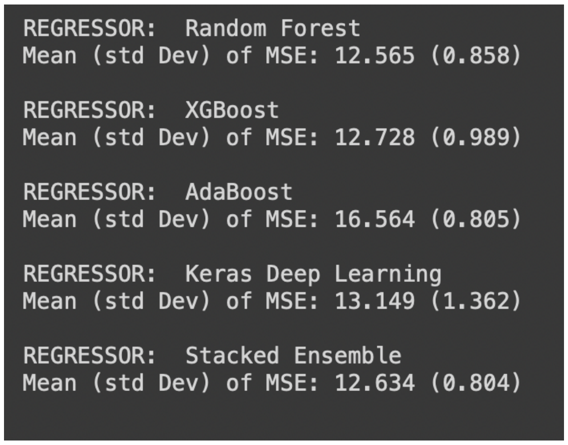
For modelling the data, we performed Recursive Feature Elimination (RFE) to obtain the 10 most important features using a RandomForestRegressor. We obtained an accuracy of 97.96% after running RFE. To visualize the important features, we plotted a pairplot and used Sweetviz to display automated visualization. We validated the LinearRegression and RandomForestRegressor models and obtained an RMSE value of 4.18 and 1.29 respectively. We then built the stacked ensemble model and the keras deep learning model.

We ran a regressor assessment and obtained the following values



```
REGRESSOR:  Multiple Linear Regression
Mean (std Dev) of MSE: 17.870 (1.690)

REGRESSOR:  Lasso Regression
Mean (std Dev) of MSE: 85.587 (4.024)

REGRESSOR:  Ridge Regression
Mean (std Dev) of MSE: 17.924 (1.662)

REGRESSOR:  ElasticNet Regression
Mean (std Dev) of MSE: 85.587 (4.024)

REGRESSOR:  kNN
Mean (std Dev) of MSE: 13.823 (1.088)

REGRESSOR:  SVM
Mean (std Dev) of MSE: 14.265 (1.232)

REGRESSOR:  Multilayer Perceptron
Mean (std Dev) of MSE: 25.451 (1.863)

REGRESSOR:  Random Forest
Mean (std Dev) of MSE: 58.643 (35.248)

REGRESSOR:  XGBoost
Mean (std Dev) of MSE: 12.737 (0.888)

REGRESSOR:  AdaBoost
Mean (std Dev) of MSE: 12.728 (0.989)

REGRESSOR:  LightGBM
Mean (std Dev) of MSE: 16.439 (1.078)
```

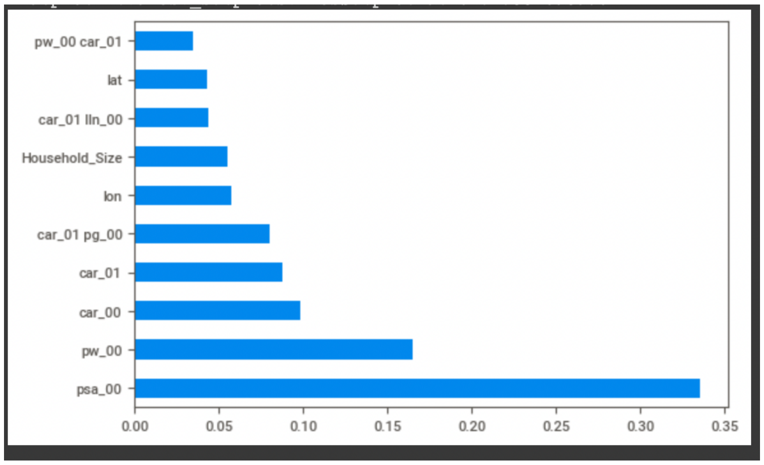
We then ran an ensemble assessment and obtained the following results



```
REGRESSOR:  Random Forest
Mean (std Dev) of MSE: 12.565 (0.858)

REGRESSOR:  XGBoost
Mean (std Dev) of MSE: 12.728 (0.989)

REGRESSOR:  AdaBoost
Mean (std Dev) of MSE: 16.564 (0.805)

REGRESSOR:  Keras Deep Learning
Mean (std Dev) of MSE: 13.149 (1.362)

REGRESSOR:  Stacked Ensemble
Mean (std Dev) of MSE: 12.634 (0.804)
```

It is clearly seen that the ensemble methods perform better than the regressors with RandomForestRegressor being the best model out of them.

Since RandomForest Regressor was the winner, we did hyperparameter tuning for it. We then performed a randomized search on the hyperparameters to get the best set of hyperparameters for our model, followed by a grid search on a range of values around the hyperparameters we got from the randomized search. We also plotted the feature importance.



Finally, we make a prediction on the test data using Random Forest.

Conclusion:

Based on our model, we recommend using a RandomForest Regressor to predict whether the households are in fact female headed. The MSE value we obtained with our regressor is **3.7402**.
Further, we plan to evaluate the stacked ensemble model with tuned hyperparameters for each of the components. Due to limited computational power, we are yet to confirm whether the stacked ensemble will in fact perform better than the Random Forest. By further evaluating our model, and tuning various hyperparameters, we are confident in reaching the benchmark MSE which is currently marginally lower than the MSE value we obtained by 0.14, as determined by the Zindi Trailblazers program.

Our main objective is to promote gender equality and through this model we can gain important information about the percentage of female headed households in South Africa with income less than R19.6k at the ward level between census years. By only gaining important information of the features used in this model we can accurately determine the level of equality without the need of a large amount of census data in the future.

References:

Dataset source : https://zindi.africa/competitions/be-a-trailblazer/data

Brownlee, J. (2017, February 9). How to train a final machine learning model. Machine Learning Mastery. Retrieved March 18, 2022, from https://machinelearningmastery.com/train-final-machine-learning-model/

Brownlee, Jason. “Stacking Ensemble Machine Learning with Python.” *Machine Learning Mastery*, 26 Apr. 2021, https://machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/

Brownlee, Jason. “How to Scale Data with Outliers for Machine Learning.” *Machine Learning Mastery*, 27 Aug. 2020, https://machinelearningmastery.com/robust-scaler-transforms-for-machine-learning/

Erik Marsja, &amp; *, N. (2020, November 20). How to use square root, log, &amp; box-cox transformation in python. Erik Marsja. Retrieved March 18, 2022, from https://www.marsja.se/transform-skewed-data-using-square-root-log-box-cox-methods-in-python/

Brownlee, Jason. “How to Choose a Feature Selection Method for Machine Learning.” *Machine Learning Mastery*, 20 Aug. 2020, https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

Brownlee, Jason. “How to Use Polynomial Feature Transforms for Machine Learning.” *Machine Learning Mastery*, 27 Aug. 2020, https://machinelearningmastery.com/polynomial-features-transforms-for-machine-learning/

Top 7 cross validation techniques with Python Code. Analytics Vidhya. (2021, November 19). Retrieved March 18, 2022, from https://www.analyticsvidhya.com/blog/2021/11/top-7-cross-validation-techniques-with-python-code/