Analysis of Predicting Poverty Around the World

Microsoft Professional Capstone: Data Science
James Chen, July 2019, Microsoft DAT102x

Executive Summary:

This executive summary is in support of the learning performed as the Capstone project for the certification of Microsoft Professional Program of Data Science. By using the 20 variables from the data source in the real world, I could predict the poverty probability from the test_values with certain accuracy. Through the data science methodology which I learned in this program, it turned out the Coefficient of Determination for 0.4213 in Azure Machine learning studio and 0.3978 in the Competition Challenge.

KEY ELEMENTS TO USE:

- DEMOGRAPHICS 6/7 :

  `country` - `is_urban` –`age` - `female` - `married` -`relationship_to_hh_head`

- EDUCATION 3/6:

  `education_level` - `literacy` - `can_add` -

- EMPLOYMENT 4/10:

  `employment_type_last_year` -`share_hh_income_provided` – `income_friends_family_last_year`

  `income_government_last_year`

- ECONOMIC 2/16:

  `avg_shock_strength_last_year - borrowed_for_daily_expenses_last_year`

- PHONE 4/7:

  `phone_technology - can_use_internet - can_make_transaction – phone_ownership`

- FINANCIAL INCLUSION 1/12:

  `num_financial_activities_last_year`

Data Explore:

At the first glance of the dataset: train_values_wJZrCmI, there are 58 variables with two features, Numeric and String, are categorized into six dimensions: Demographics, Education, Employment, Economic, Phone, and Financial inclusion. According to my personal opinion, I felt some dimensions influence more when predicting poverty probability such as the country. The people from a certain country may have a high percentage to either rich or poor. This kind of concept somehow is proved in the model scoring. And base on the training result I could better select the variables when doing the model training. Moreover, when looking deeper into the variables, I can find that some of them should be the category label types but Numeric or Binary in terms of the data preprocessing.

Data Preprocessing:

Before I start to train by the algorithm and build the machine learning model, I did the data preprocessing process to make the data resource to be more accountable.

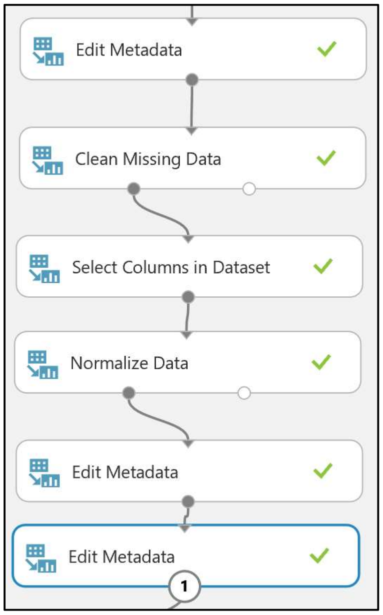
- Edit Megadata:
  To clear the feature for *Row_id* since we believe it should not influence the model accuracy.

- Cleaning missing data:
  To replace the missing data by probabilistic PCA. As we can find there are lots of missing data in the original dataset ,and will be the determination when impact the accuracy of the model

- Select columns in Dataset:
  Select the elements which may be the potential candidates for us to explore our model.

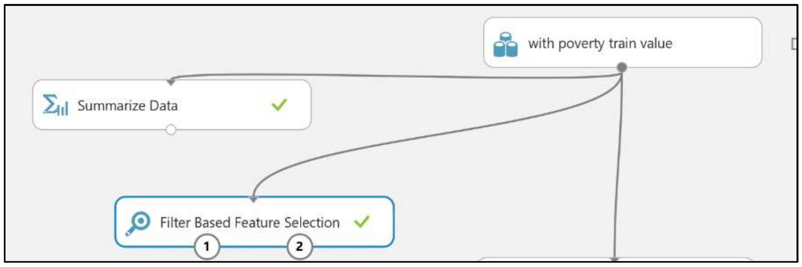
- Normalize data:
  To normalize the numeric variables which are not normal bell-shaped distribution by using Zscore

- Edit Megadata:
  To define all category variables accordingly

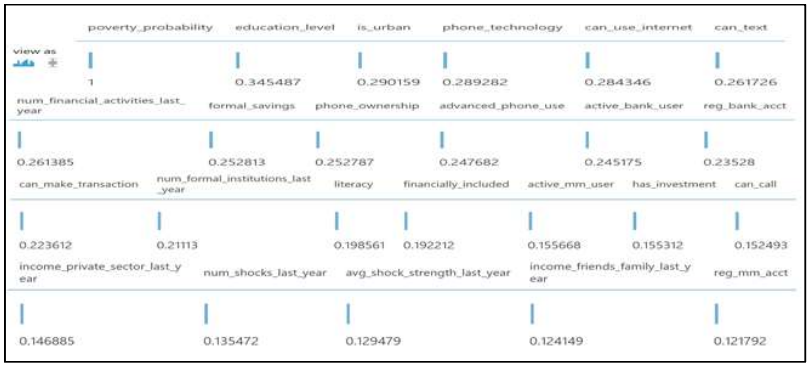


Feature selection:

In order to narrow down the features which could better affect the result, I use the Summarize Data function to review the descriptive statistic. Then , I use Filter Based Feature Selection to see the relative weight which related to *Poverty_Probability*.



By these two measures, I determined half variables (24) to be selected for modeling. However the prediction result was not quit good (accuracy between 0.32 to 0.37 in AZ ML Studio). This experiment showed even though the variable has high coefficient to the others does not mean it could highly influence the prediction.



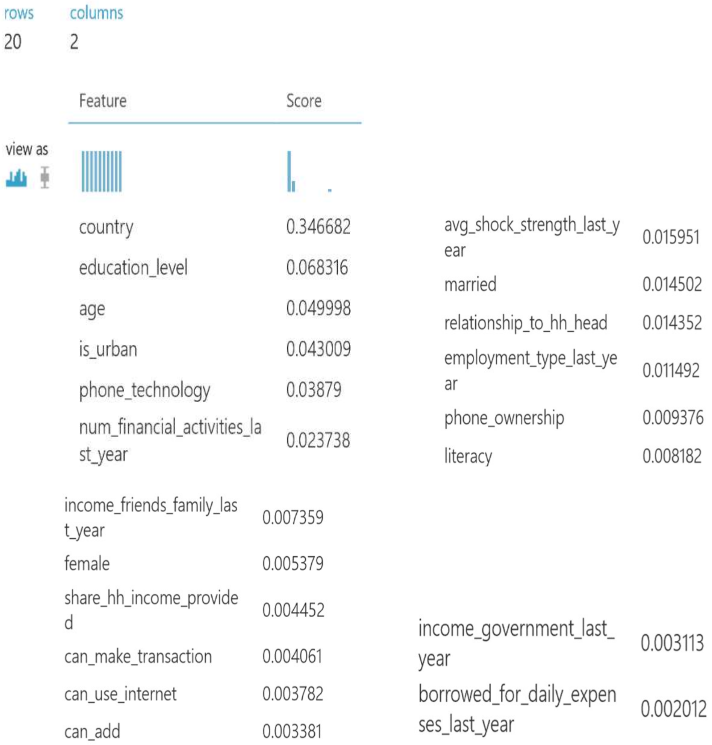
poverty_probability	education_level	is_urban	phone_technology	can_use_internet	can_text
1	0.345487	0.290159	0.289282	0.284346	0.261726

num_financial_activities_last_year	formal_savings	phone_ownership	advanced_phone_use	active_bank_user	reg_bank_acct
0.261385	0.252813	0.252787	0.247682	0.245175	0.23528

can_make_transaction	num_formal_institutions_last_year	literacy	financially_included	active_mm_user	has_investment	can_call
0.223612	0.21113	0.198561	0.192212	0.155668	0.155312	0.152493

income_private_sector_last_year	num_shocks_last_year	avg_shock_strength_last_year	income_friends_family_last_year	reg_mm_acct
0.146885	0.135472	0.129479	0.124149	0.121792

Therefore, I replaced three variables once to re-run the model, and tried to find the best combination. Finally, I revisited the variables which already been included into my model, then reviewed the rest and get some which could be potential to have the impact to my model base on my study of Sub-Saharan Africa, and chose five of them to re-run to make the final model, and get the final twenty variables.



rows 20 columns 2

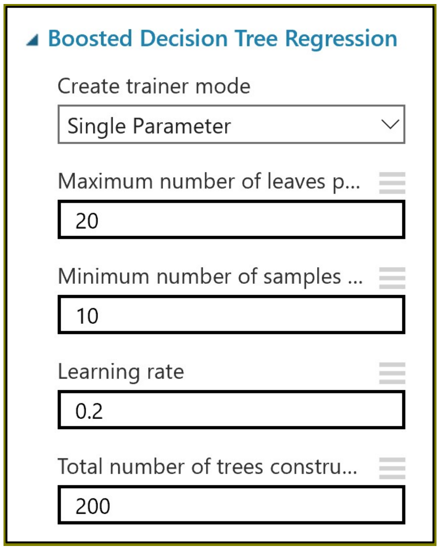
view as

Feature	Score
country	0.346682
education_level	0.068316
age	0.049998
is_urban	0.043009
phone_technology	0.03879
num_financial_activities_last_year	0.023738
avg_shock_strength_last_year	0.015951
married	0.014502
relationship_to_hh_head	0.014352
employment_type_last_year	0.011492
phone_ownership	0.009376
literacy	0.008182
income_friends_family_last_year	0.007359
female	0.005379
share_hh_income_provided	0.004452
can_make_transaction	0.004061
can_use_internet	0.003782
can_add	0.003381
income_government_last_year	0.003113
borrowed_for_daily_expenses_last_year	0.002012

Model selection:

To determine the best one to use, I did several exercises by using different regression models. Testing Linear regression, Boosted Decision Tree Regression, Decision Forest Regression, and Neural Network regression. For this exercise, my model performed the best with Boosted Decision Tree regression.

Moreover, by using the Tune Model Hyperparameters, it runs the Boosted Decision Tree Regression model against the data exploring settings as below:



◢ Boosted Decision Tree Regression

Create trainer mode

Single Parameter

Maximum number of leaves p...

20

Minimum number of samples ...

10

Learning rate

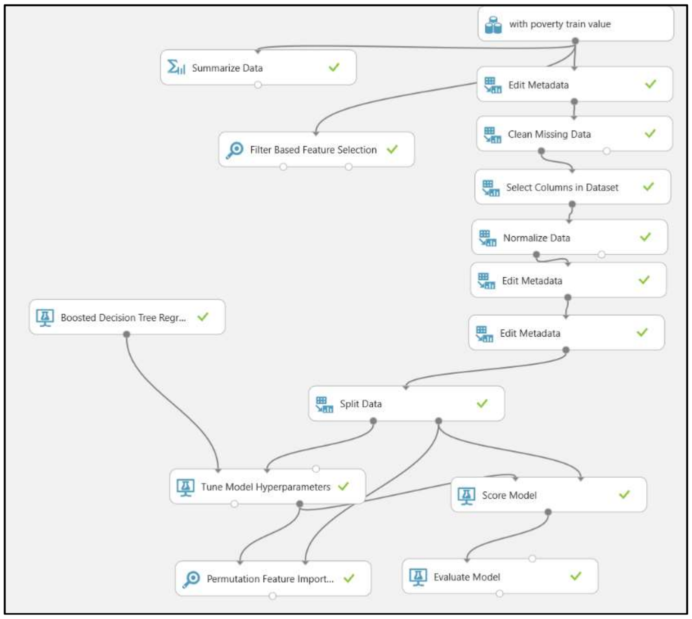
0.2

Total number of trees constru...

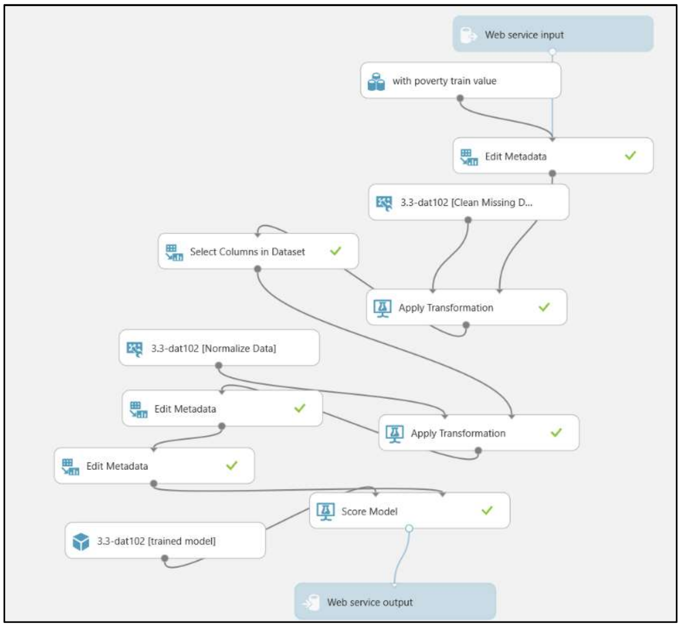
200

As the result, it turns into the best Accuracy of R^2 , and the highest Coefficient of Determination.

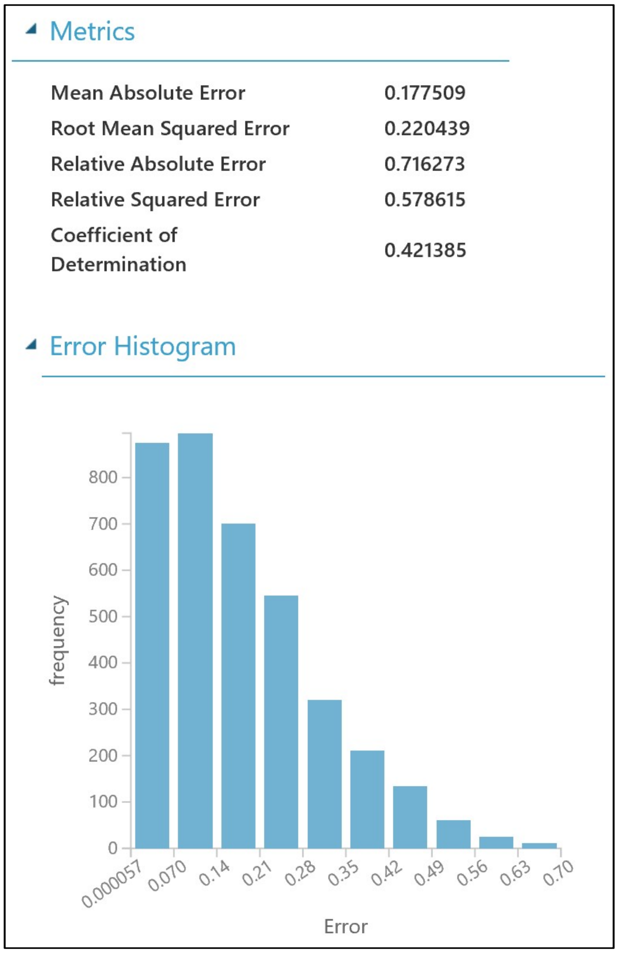
Final Model



Final Predictive Model:



Model evaluation :



Metrics

Mean Absolute Error	0.177509
Root Mean Squared Error	0.220439
Relative Absolute Error	0.716273
Relative Squared Error	0.578615
Coefficient of Determination	0.421385

Error Histogram

Conclusion:

The final model in this capstone achieves a coefficient of determination R^2 of around 0.421 in Azure ML Studio, and 0.3978 in Competition which turns out to be 81/100 in edx grade.

Through this analysis, I summarize the finding as follows:

1. When doing the prediction exercise of the real world, the data quality is very important. Even though we clean and modify the missing data, then we remove the data which has few impacts for this prediction. The final accuracy turns out only close to 0.4 or slightly above (in the competition website as well) , which means all variables of this project may not be the best elements to predict poverty.

2. The mathematically coefficient does not mean to have a high potential for the poverty prediction. For example, when running Filter Based Feature Selection, the Country shows a very low influence on the prediction, but actually, it gives a very high score for our modeling.

3. When selecting variables, though the more variables generate the higher score, however, they also give the deviation to affect the accuracy of the prediction model. Therefore, there is a trade-off with how many variables to be used.

4. The algorithm is critical for the accuracy of the prediction model. When executing the ML studio, the accuracy varies when using different algorithms. The reason might be due to data nature. Most variables in this project are binary type (True/False), if the types are more different, the algorithm should give the weight differently.