deepssharma / **project_phase3_tanzania** Public

Code Issues Pull requests Actions Projects Wiki Security

main 1 branch 0 tags Go to file Add file Code

deep... 151b0f9 now		17 commits
.ipyn...	Near final Vers...	17 hours ago
ima...	Added final i...	10 minutes ago
Data...	First Commit	2 days ago
Data...	First Commit	2 days ago
EDA....	Near final Vers...	17 hours ago
Not...	Pdf of the Not...	3 minutes ago
Pum...	First Commit	2 days ago
REA...	Update READ...	now
mod...	Refixed the er...	5 minutes ago
pres...	Added final i...	10 minutes ago
proj...	First Commit	2 days ago
trial.i...	Near final Vers...	17 hours ago

README.md

Pump it Up: Data Mining the Water

About

No description, website, or topics provided.

Readme
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

Jupyter Notebook 100.0%

Table

Data Exploration and Business Problem

The data was obtained from the Pump it Up: Data Mining the Water Table provided at DrivenData. The data is collected from Taarifa and the Tanzanian Ministry of Water, and is used to predict which pumps are functional, which need some repairs, and which don't work at all! The Taarifa Platform is an open source API designed to use citizen feedback on local problems.The major goal of this project is to provide clean water access to the people of Tanzania. Currently, the people of Tanzania have poor access to clean drinking water throughout the entire country. Approximately 47% of all Tanzanian citizens do not have access to clean drinking water. Over 1.4 billion dollars in foreign aid has been giving to Tanzania in an attempt to help fix the freshwater crisis. However, the Tanzanian government has been struggling to fix this issue.

The main focus of this study is to predict the functionality of water pumps using machine learning models. If models are accurate, this could help save the Tanzanian government a lot of time and money. Predicting correctly the faulty water pumps would help to cut the cost needed to send workers to each and every water pump for inspection. The government can use this study to find the water pumps that are working, need repair and the ones aren't working at all.

A complete list of variables in the dataset is given below:

Target Feature:

- status_group - If the water pump is functional, non-functional or need repairs

Predictive Features:

- amount_tsh - Total static head (amount water available to waterpoint)
- date_recorded - The date the row was entered
- funder - Who funded the well
- gps_height - Altitude of the well
- installer - Organization that installed the well
- longitude - GPS coordinate

- latitude - GPS coordinate
- wpt_name - Name of the waterpoint if there is one
- num_private -
- basin - Geographic water basin
- subvillage - Geographic location
- region - Geographic location
- region_code - Geographic location (coded)
- district_code - Geographic location (coded)
- lga - Geographic location
- ward - Geographic location
- population - Population around the well
- public_meeting - True/False
- recorded_by - Group entering this row of data
- scheme_management - Who operates the waterpoint
- scheme_name - Who operates the waterpoint
- permit - If the waterpoint is permitted
- construction_year - Year the waterpoint was constructed
- extraction_type - The kind of extraction the waterpoint uses
- extraction_type_group - The kind of extraction the waterpoint uses
- extraction_type_class - The kind

of extraction the waterpoint uses

- management - How the waterpoint is managed
- management_group - How the waterpoint is managed
- payment - What the water costs
- payment_type - What the water costs
- water_quality - The quality of the water
- quality_group - The quality of the water
- quantity - The quantity of water
- quantity_group - The quantity of water
- source - The source of the water
- source_type - The source of the water
- source_class - The source of the water
- waterpoint_type - The kind of waterpoint
- waterpoint_type_group - The kind of waterpoint

Modeling

The data was split into training and test sets.

The data was pre-processed. This is a classification problem with three classes! A detailed data exploration was done to understand different variables provided in the dataset. See Notebook project_v3.ipynb in the same github repository

Several types of classifiers were built, tuned (using GridSearchCV to test combinations of hyperparameters) and validated:

- Logistic Regression
- Random Forest
- XGradient Boosted
- Stacking Classifier (using above models)

Evaluation

I used Roc_Auc mostly and also looked at f-scores as the scoring metric for tuning hyperparameters and evaluating model performance.

- The Roc_Auc metric utilizes "probabilities" of class prediction. Based on that, we’re able to more precisely evaluate and compare the models. We also We also care equally about positive and negative classes, and the roc curve gives a desirable balance between sensitivity/recall (maximizing True positive Rate), Precision and Accuracy score .

- To bulid a good model one needs to ca refully evaluate the predictions and understand the role of different features that drive the model predictions. A careful comparison between test and train data helps to understand to a great extent the model characteristics

Major Issues

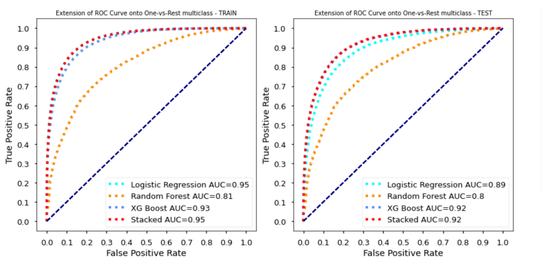
- It was a challenging dataset given its length ~(60K entries) and number of categorical variables (which cause issue in one-hot encoding that generates too many columns). This was a major issue when I had to run GridSearchCV for hyperparameter tunings. I wasnt able to run even one model even after reducing the number of columns from 41 to 23. I killed the process after waiting for 1.5 days. This is when I found out about **HalvingGridSearchCV**. This reduces the running time by factors anywhere ranging from 2-5. Sklearn says its still in experimentation and examples show that the parameters found by two methods are pretty much same. Using this I was able to run GridSearch in a few hours for each model scenario. However this feature is only available in recent version of sklearn and so I had to update it

- The second issue was that this is a ternary classification problem (not the usual Yes/No binary), so I had to use **ovr (One vs Rest)** option and to plot ROC curves for this multi-label problem required update sklearn as well.

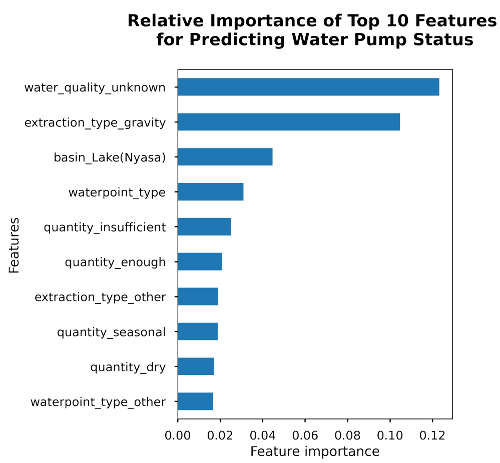
Results

XGB Classifier is the best model found in this study with an roc_auc_score of about 91% for the training set and 89% for the test data. A summary of the comparison of the models is shown below:

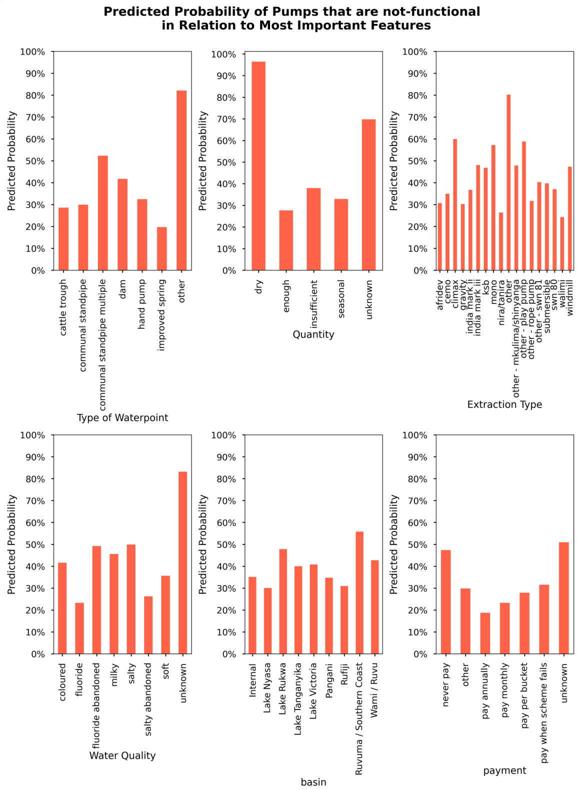
	cv_train	train	test
Logistic Regression	0.827096	0.931100	0.827683
Random_Forest	0.780599	0.795829	0.782454
XG Boost	0.877268	0.905872	0.877508
Stacked	0.872568	0.921331	0.875255



- The most features from best model are shown below:



Predicted Probablities for Non-Functional Pumps



Recommendations

- The Water ministry should reallocate funds to replacing pumps at salty locations with the correct types.
- Allocate some budgets for R&D to find out the right extraction type for a given location.
- Investigate for the reasons why for some pumps payments are never paid or unknown.
- Also look into the water pumps that are in dry areas and or whom the quantity of water is not known.