T-hoveen /
Predicting-Waterpoint-Functionality-in-Tanzania

Code Issues Pull requests Actions Projects Wiki Security Insights Setting

0 stars 0 forks 1 watching Branches Activity
Tags

Public repository

1 Branch 0 Tags Go to file t Go to file + Add file Code ···

T-hoveen project db33130 · 1 hour ago

Data	project	1 hour ago
visualizations	project	1 hour ago
.gitignore	modelling	15 hours ago
README.md	project	1 hour ago
functions.py	modelling	15 hours ago
index.ipynb	project	1 hour ago

README

Table of Contents

Business Overview

After gaining independence, the Tanzanian government introduced a policy to provide free potable water to all rural residents by 1991. Formalized in 1971, this policy made the government responsible for developing, operating, and maintaining water supply systems without implementing cost recovery measures. During the 1970s, many projects were funded by donors, particularly from Sweden, leading to the construction of numerous waterpoints. By the time the dataset was collected, some of these waterpoints remained fully operational, others required repairs, and some had ceased functioning altogether.

Objectives

The main goal of this project is to develop a model capable of predicting whether a waterpoint is functional or non-functional based on a set of independent variables. This insight can assist the Tanzanian Government and other stakeholders in identifying waterpoints that may require repairs based on their specific characteristics.

Stakeholders

- Government Entities: The Tanzanian Ministry of Water, responsible for infrastructure management and policymaking.
- Non-Governmental Organizations (NGOs): Organizations working to improve access to safe water in underserved areas.
- Local Communities: Direct users who benefit from operational waterpoints.
- Funders and Donors: Investors focused on the impact and sustainability of water infrastructure projects.

Success criteria

- **Accuracy:** The model should have a high accuracy in predicting the status of waterpoints.

Data Understanding

To achieve my objective I sourced my data from

- DrivenData. (2015). Pump it Up: Data Mining the Water Table. Retrieved [December 3 2024] from https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table.

Constraints

- **Data Quality:** The accuracy of the model depends on the quality and completeness of thedata.
- **Resource Limitations:** Limited resources for maintenance and repairs may affec tth eimplementation of the model’s recommendations.

- **Multinormial Classification:** The dataset is a multinormial classification problem but I will treat it as a binary classification problem

Data Visualisations

Water type group vs status group

Water type group vs status group

The above figure shows that simple pumps are the most common pumps found in Tanzania

Payment Type vs Status group

Water type group vs status group

The figure shows that most wells users dont have to pay to use the water point but coincidentally most non functional wells were used by users who did not have to pay. This means that those who managed the water points may have lacked funds to repair their wells leading to the wells they managed to cease from functioning

Extraction group vs Status group

Extraction group vs status group

The figure shows that water points that used gravity as a way to extract water had the most functioning wells. This indicates that pumps that used gravity

Model Performance

1. **LogReg with Standard Scaler and One Hot Encoding**

- **F1 Score:** 0.8195
- **ROC AUC Mean:** 0.8474
- **ROC AUC std:** 0.0029
- **Summary:** This model showed good performance with a stable ROC AUC score showing consistent results across different folds

2. **LogReg with Min Max Scaler and One Hot Encoding**

- **F1 Score:** 0.8195
- **ROC AUC Mean:** 0.8478
- **ROC AUC std:** 0.0027
- **Summary:** This model showed a similar performance with the previous with almost identical ROC AUC scores but was more consistent across different folds

3. **LogReg with Robust Scaler and Target Encoding**

- **F1 Score:** 0.8077
- **ROC AUC Mean:** 0.8250
- **ROC AUC std:** 0.0043
- **Summary:** This model performed worse than the previous models with a wore F1 score and ROC mean. It showed that target encoding made the model performed worse than One Hot encoding. It also was more inconsistent in performance with a worse ROC AUC std

4. **LogReg with Robust Scaler and One Hot Encoding**

- **F1 Score:** 0.8196
- **ROC AUC Mean:** 0.8479
- **ROC AUC std:** 0.0028
- **Summary:** This model showed similar performance with other linear regression models with one hot encoding further showing the strength of using one hot encoding for our categorical performance

5. **Decision Tree with Robust Scaler and One Hot Encoding**

- **F1 Score:** 0.8218
- **ROC AUC Mean:** 0.8179
- **ROC AUC std:** 0.0051
- **Summary:** This model showed the worst performance among our other models with the worse ROC AUC scores compared to other. The model also showed the least consistency on each fold

model choice

From the above analysis of model performance we have witnessed that logistic regression have performed better than the decision tree model. One Hot Encoding has proved to be the preferred encoding to deal with categorical columns. It improved the model scoring of model that used them. Choice of scaling showed no effect on our model metric scoring though robust scaling improved the runtime of our model.

Therefore the choice of model from my analysis will be the logistic regression with Robust Scaler and One Hot Encoding

Recommendations

1. **Payment** Based on the findings it is recommended to priotise building wells that have a payment transaction. This ensures that scheme managers have money they can use to maintain wells in Tanzania
2. **Improved data collection.** The dataset had a lot errors that were noticed in data cleaning. This can affect the model accuracy so improvement in data collection might improve the accuracy of the model
3. **Integration of external factors** : Further information such as climate of the area would have been useful in our analysis

THANK YOU

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

● Jupyter Notebook 99.7% ● Python 0.3%

Suggested workflows

Based on your tech stack

Python application

Configure

Create and test a Python application.

Pylint

Configure

Lint a Python application with pylint.

Django

Configure

Build and Test a Django Project

More workflows

Dismiss suggestions