ckucewicz /
water_well_classification

Code Issues Pull requests Actions Projects Wiki Security Insights Set

1 star 0 forks 1 watching Branches Activity
Tags

Public repository

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

ckucewicz Update README.md f65fe44 · now

deliverables	Updated Presentation Slides	7 minutes ago
trained_models	Updated Presentation Slides	7 minutes ago
visualizations	Updated Presentation Slides	7 minutes ago
.gitattributes	Updated notebook	1 hour ago
.gitignore	Initial Commit	last month
Miscellaneous Notebook.ipy...	Updated notebook	1 hour ago
Notebook.ipynb	Updated Presentation Slides	7 minutes ago
README.md	Update README.md	now

README

Water Well Classification Project

Author: Chris Kucewicz

Business Understanding

Background

The country of Tanzania is struggling to provide safe, clean water to its more than 67 million citizens. A non-governmental organization (NGO) is assisting the Tanzanian government in addressing this critical issue by repairing damaged water wells. With over 70k water wells scattered across an area larger than twice the size of California, testing every single well is both costly and time-consuming.

Water from these wells is not only vital for drinking but also essential for cooking, sanitation, and hygiene practices. Ensuring access to clean water significantly impacts the health and livelihoods of the community. Therefore, the NGO requires an efficient method to identify which water wells need repair, streamlining their efforts to provide safe, clean water for all Tanzanians.

Business Goals

The primary focus of this machine learning project is to assist the NGO in identifying all water wells requiring repair, enabling timely interventions to ensure safe, clean water for Tanzanians. Specifically, the project aims to accurately identify every well in need of repair while minimizing false ***negatives***—cases where a well is incorrectly classified as functioning when, in fact, it requires repair.

Although some functional wells may be incorrectly classified as needing repair, the prevention of potential health hazards is of utmost importance. Therefore, the model prioritizes identifying wells that need repair, erring on the side of caution to ensure the safety of citizens who rely on these essential water sources.

Business Success Criteria

The success of this project will be measured primarily by the model's ability to classify wells in need of repair. The key metric associated with this ability is called recall, which addresses the question:

> *"Out of all the wells that actually need repair, what percentage did our model correctly identify as needing repair?"*

Recall is calculated by dividing the number of true positives (the number of wells that genuinely require repair) by the combined total of true positives and false negatives (the wells that require repair but were incorrectly labeled as functional).

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

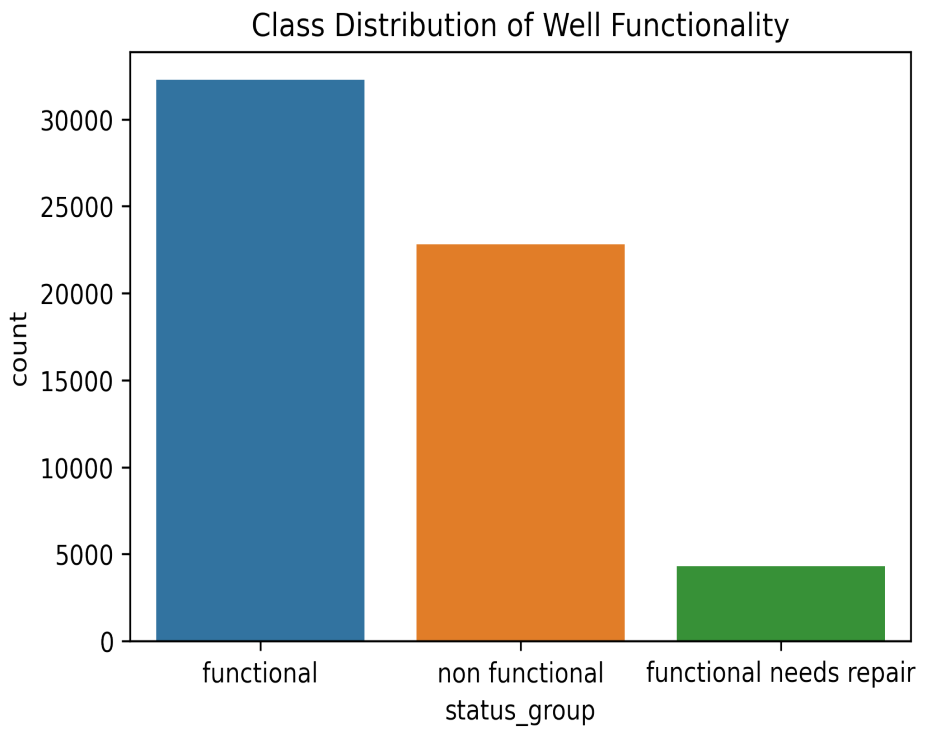
While other metrics, such as accuracy and precision, are often used to evaluate classification models, recall is most suitable here because it addresses the primary goal of identifying all wells needing repair. For instance, high precision would mean that when the model predicts a repair need, it is usually correct. However, this would not be as beneficial here. If the model only predicts repair needs for a few wells, it risks missing many others, potentially leaving large gaps in coverage. Similarly, focusing on accuracy could yield a model that classifies most wells as needing repair if there's a class imbalance, without improving the efficiency or impact of repairs.

By focusing on recall, this project will help the NGO allocate its resources more effectively, reducing costs and time associated with well testing, and ultimately supporting improved public health and quality of life in Tanzania.

Data Understanding

This data was collected by Taarifa and the Tanzanian Ministry of Water. The dataset includes 41 features and 59400 entries.

Each entry in the dataset represents a water well in Tanzania, where `id` is its unique identifier. Additional information is included about each well such as the `latitude` and `longitude`, along with its water source -- `source_type` -- and its total static head -- `amount_tsh` -- (in other words the amount of water available to waterpoint).



The target in this classification problem is stored in the column labeled `status_group`. Prior to any cleaning and preprocessing, this column includes three classes: functional, functional needs repair, and non-functional.

More information about the data can be found here.

Data Preparation

Based on my observations in the Data Understanding phase, I completed the following steps during the data preparation phase to clean and prepare the dataset for modeling:

1. **Dealing with Duplicates**:

- Irrelevant features or those containing duplicate information were dropped. In cases where two features contained the same information, the feature with lower cardinality was retained, as lower cardinality typically leads to smoother machine learning processes.

2. **Processing the `installer` Feature**:

- The high cardinality of the `installer` feature (2,145 unique values) was addressed by grouping similar entries and narrowing the focus to the top 25 installers based on frequency. All other installers were classified as "Other," reducing the cardinality from 2,145 to 25.

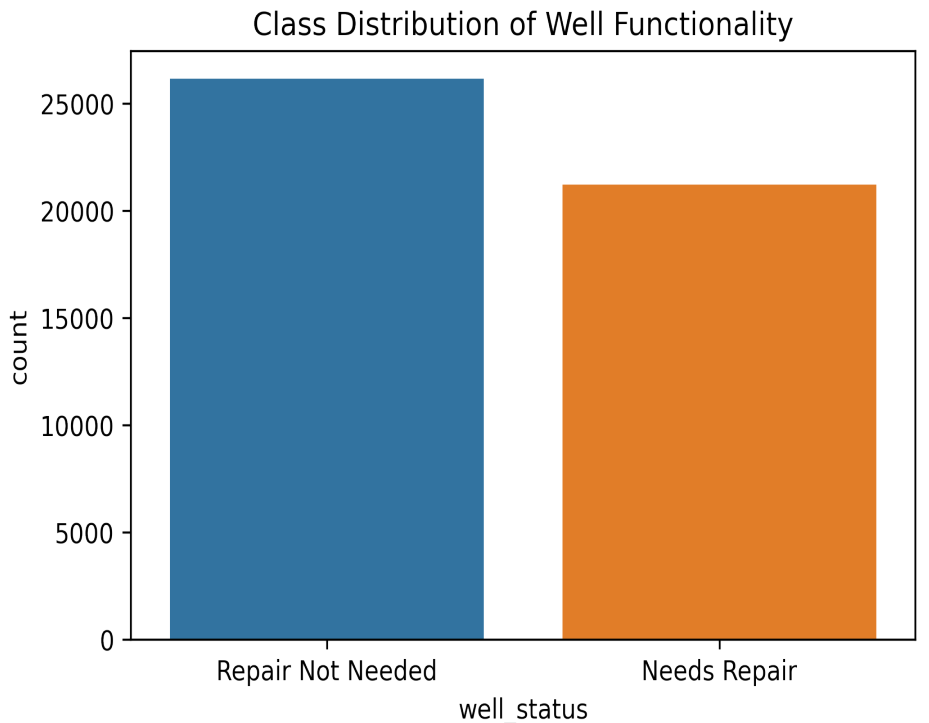
3. **Handling Null Values**:

- At this stage, only three features contained null values, and none had more than 7% of their entries missing. Since dropping these rows would not result in a significant loss of data, I decided to remove them.

4. **Reclassifying the Target Variable**:

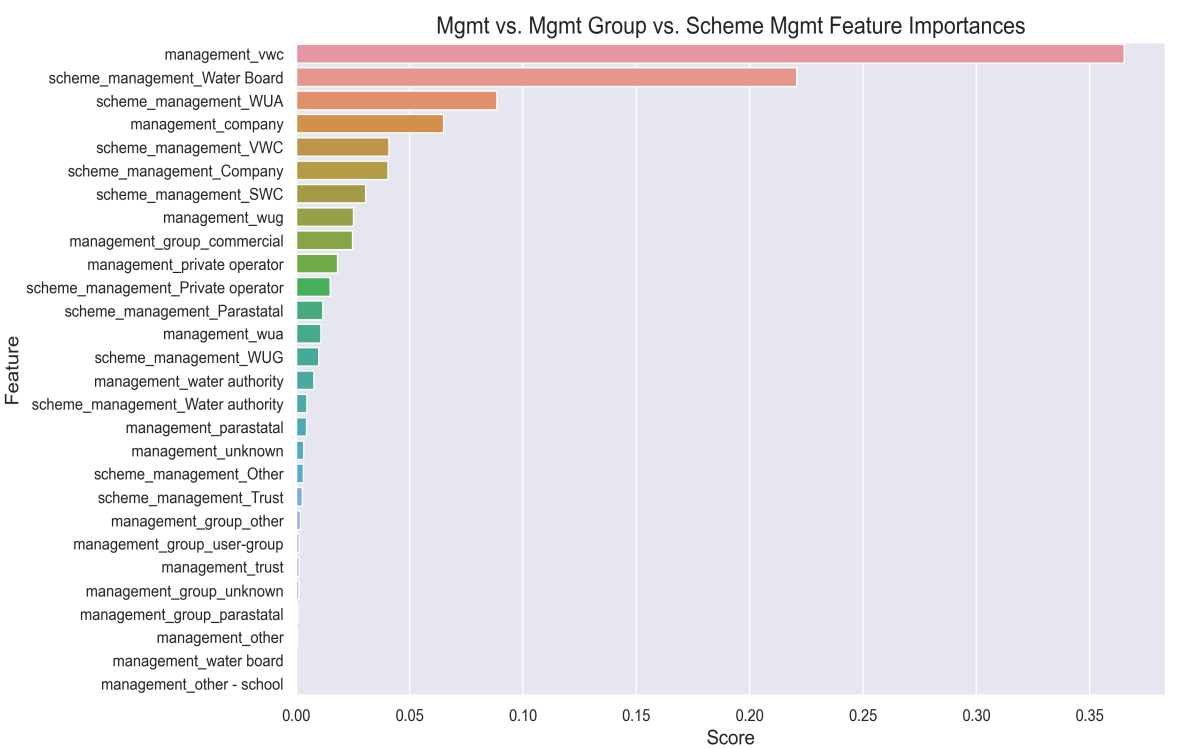
- The original target variable was first reclassified from three classes into two classes, and then converted into a binary Boolean target. This was done to simplify modeling and ensure consistent

interpretation of well functionality, aligning with the project's objectives.



5. **Feature Engineering and Selection**:

- For the `management` vs. `management_group` vs. `scheme_management` and `region` vs. `district_code` features, I trained simple decision trees using each set of features individually. By examining the feature_importances_ attribute, I identified the most significant variable from each group. The top feature from each pair was then selected and retained for further modeling.

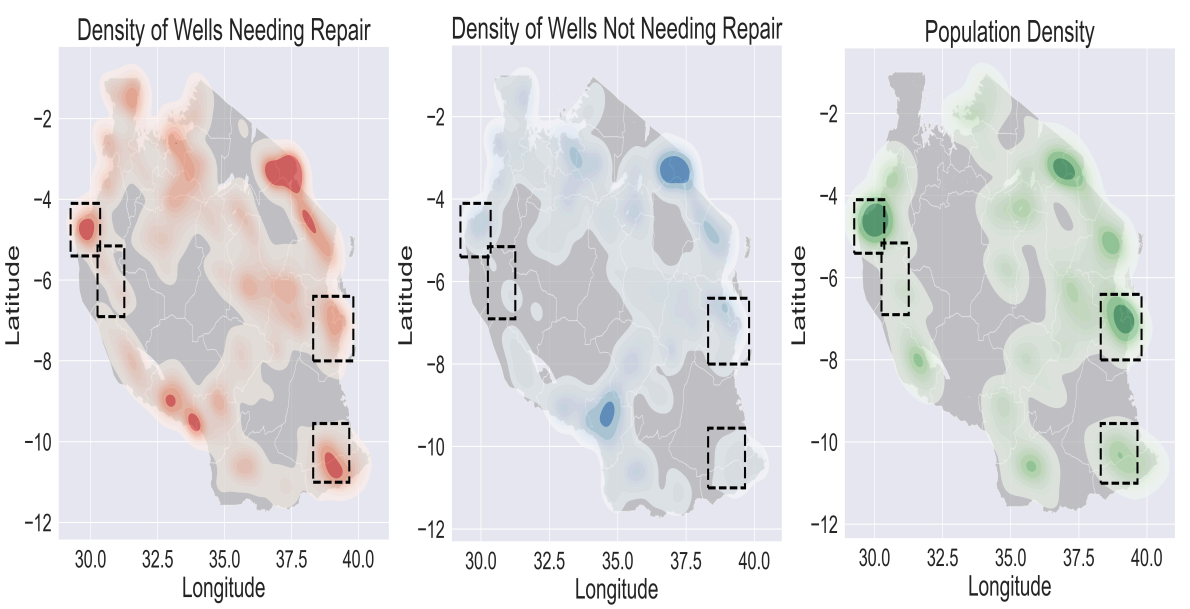


By the end of the data cleaning process, the dataframe was reduced to 19 columns, down from 40, and 47,000 rows, down from nearly 60,000, with all null values removed. Despite the reduction of over 10,000 entries, 47,000 rows still provides plenty of data for training the models, even after performing the train-test split.

Exploratory Data Analysis

The following are findings from the EDA portion of this project:

- **Distribution of Numeric Features**: The numeric features such as `population`, `amount_tsh`, and `gps_height` are not normally distributed and appear to be relatively right-skewed. This skewness can affect model performance by making certain features disproportionately influential. To address this, I applied the `MinMaxScaler()` during preprocessing. The scaler normalizes the data to a range between [0, 1], helping to mitigate the impact of outliers and ensuring that all features contribute equally to the model, which is particularly important for algorithms sensitive to feature magnitudes.
- **Installer Influence**: Wells installed by `government`, `finnish_govt`, and `rwe` had a higher proportion of nonfunctioning wells compared to functional wells, indicating a potential correlation between installer type and well functionality.
- **Water Basins Impact**: Wells located within the `Ruvuma / Southern Coast` and `Lake Rukwa` basins showed a higher proportion of nonfunctioning wells relative to functioning ones.
- **High-Priority Areas**: Four specific areas were identified as high priority for intervention, meeting all three of the following criteria:
  - A high density of wells in need of repair
  - A low density of nearby functional wells
  - Relatively high population density



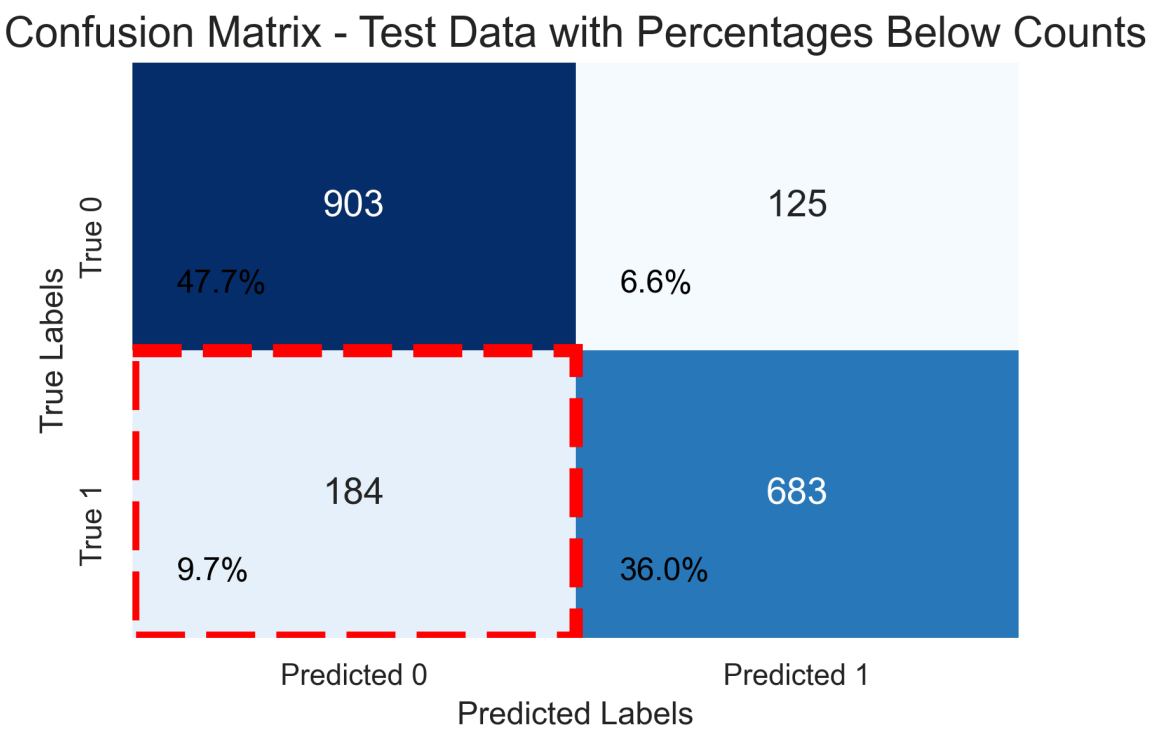
These insights provide a focused starting point for addressing regions with the greatest need for well repairs and functionality improvement.

Modeling

In the Modeling phase, my primary goal was to build and refine predictive models that classified wells needing repair and those that did not. To prevent data leakage and ensure that the model was trained on clean, unbiased data, I started by performing a train-test-split with an 80/20 ratio. This approach ensured that the model was trained on the majority of the data while withholding a portion for validation and testing to evaluate performance on unseen data.

The data was preprocessed through one-hot encoding for categorical features and scaling for numerical features to standardize the input for training. To track model performance, I used the Evaluation Metrics Function, with a focus on recall scores, as minimizing false negatives was crucial for this project.

For the baseline, I began with a logistic regression model, which served as a simple starting point for comparison. From there, I trained more complex models, such as decision trees and random forests, to improve recall scores. To address model overfitting and reduce complexity, I applied feature selection techniques like feature_importances_ and Recursive Feature Elimination (RFE). I also used hyperparameter tuning via GridSearchCV to optimize key model parameters and enhance predictive power. Further explanations for my choices and justifications for each of these steps are provided in more detail throughout the notebook.



Evaluation

This table below presents a comparison of the recall metrics and false negative percentages for different models, with conditional formatting to highlight the best and worst performance. The shading of the cells follows a color gradient where darker shades of blue indicate better values. The values for the “Difference (Training - Test/Val Recall) (%)” column are used to assess overfitting, where higher values indicate greater overfitting, as the model performs well on training data but poorly on test/validation data.

The Decision Tree with RFE and Initial Random Forest models show a large discrepancy between their training and validation/test recall values, indicating overfitting. Although these models exhibit high training recall scores (close to 100%), their test/validation recall values are much lower, suggesting that they may not generalize well to unseen data. In contrast, the Baseline Logistic Regression model, while showing lower recall scores compared to the initial random forest model, demonstrates a more balanced performance with a smaller difference between training and testing recall. This suggests that the model generalizes better, making it more reliable in real-world applications, despite its lower recall.

Final models like the Final Decision Tree and Final Random Forest showed improvements in reducing overfitting but suffered from a significant decrease in recall, particularly on the test data, which aligns with the overfitting mitigation efforts undertaken through feature selection and hyperparameter tuning.

Overall, the Baseline Logistic Regression model appears to be the most balanced, achieving reasonable recall with a low difference between training and test data performance. This model demonstrates a more reliable and generalized performance compared to the more overfit models, despite not achieving the highest recall value.

	Model Name	Training Recall (%)	Test/Val Recall (%)	Difference (Training - Test/Val Recall) (%)	False Negatives (%)
0	Decision Tree with RFE (val data)	100.00	78.60	21.40	9.70
1	Final Decision Tree (test data)	45.03	44.36	0.67	24.80
2	Final Random Forest (test data)	58.86	58.80	0.06	18.40
3	Initial Random Forest (test data)	99.90	77.45	22.45	10.10
4	Baseline Logistic Regression (test data)	66.85	66.00	0.85	15.20

Conclusion

This evaluation highlights the challenges of balancing overfitting with predictive performance. While the initial random forest model achieved the highest recall, its significant overfitting limits its reliability on unseen data. In contrast, the tuned models showed reduced overfitting but at the cost of recall performance, suggesting a loss of signal during feature selection and hyperparameter tuning.

Given the objective of minimizing false negatives in well repair identification, the initial random forest model appears most promising, despite its overfitting. However, if generalizability is a higher priority, the tuned models offer a more balanced alternative.

This project demonstrates the importance of clean data and robust feature engineering to create models that are both accurate and generalizable. Future iterations should focus on refining data preprocessing and exploring additional algorithms that may better capture the complexities of the problem.

Limitations

- **Data Quality**: The dataset required extensive cleaning, including handling missing values, duplicate features, and irrelevant features. Decisions made during preprocessing, such as which features to retain or remove, may have inadvertently affected model performance. Cleaner data could improve the signal-to-noise ratio and lead to more effective models.

- **Computing Constraints**: Complex models like random forests and hyperparameter tuning processes such as GridSearchCV are computationally expensive. Limited resources restricted the breadth of parameter grids explored, potentially impacting the optimality of the models.
- **Time Constraints**: Time limitations restricted the depth of exploratory data analysis and model optimization. With additional time, a wider array of feature selection strategies could have been explored, and larger parameter grids for hyperparameter tuning could have been tested. This might have resulted in better-performing models with improved recall and generalizability.
- **Domain Knowledge**: A lack of extensive domain knowledge about Tanzania and its water well infrastructure posed challenges during feature selection. Collaborating with local experts or incorporating region-specific insights into the analysis could improve the relevance and accuracy of the models, ensuring that they align more closely with the realities of the county.

Recommendations

1. **Prioritize Wells in High-Demand Areas**:
    - Target the Northwest (Kigoma and areas between Kigoma and Sumbawanga) and Southeast (specifically around Dar Es Salaam and Mtwara) for initial focus.
    - These regions have high population densities, significant numbers of wells in need of repair, and a low density of nearby functioning wells, as indicated by the KDE density maps. Prioritizing these areas will maximize the impact of well repairs by addressing both high demand and access issues.
2. **Use the Baseline Logistic Regression Model for Well Status Prediction**:
    - While more complex models (e.g., decision trees and random forests) may show high recall on training data, the baseline logistic regression model demonstrates better generalizability on unseen data.
    - The logistic regression model provides a more reliable and balanced prediction performance, making it a more suitable choice for predicting well status across diverse regions and conditions.
3. **Address Overfitting in Decision Tree and Random Forest Models**:
    - Both the decision tree and random forest models demonstrated high recall on unseen data but suffered from overfitting, meaning their performance dropped significantly when tested on new data.
    - If the NGO wishes to leverage these models for well status prediction, it is crucial to mitigate overfitting by improving feature selection, adjusting model hyperparameters, or employing cross-validation techniques. Reducing overfitting will ensure that the models generalize better and provide more reliable predictions in real-world settings.

Next Steps

1. **Enhance Data Collection and Sources**:
    - Invest in better data quality by ensuring more accurate, complete, and up-to-date information. This may involve collaborating with local agencies or using additional data sources to fill in gaps or reduce biases.
    - Improving data collection processes, such as regular updates and standardized reporting methods, can increase the reliability of models and lead to more effective decision-making.
2. **Address Overfitting in Random Forest and Decision Tree Models**:

- Allocate more time to refining the hyperparameters and performing more extensive cross-validation to reduce overfitting in the decision tree and random forest classifiers.
- Techniques such as pruning, ensemble methods, or adjusting the depth of trees could be explored to improve the models' ability to generalize to new data.

3. **Explore Advanced Modeling Techniques**:
   - Investigate other algorithms such as Gradient Boosting or Support Vector Machines (SVM), which may offer improved performance, especially for balancing recall and generalization.

Additional Information

View the full project in the Jupyter Notebook.

View the presentation

Contact Chris Kucewicz at cfkucewicz@gmail.com with additional questions.

Repository Structure

```
├── deliverables
    ├── water_well_classification_presentation.pdf       # non-technical presentation
slideshow
```

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%