Water Pump Project

Meredith Newhouse

Purpose

The purpose of this project is to create a model that can predict the functionality of water pumps using data from Taarifa and the Tanzania Ministry of Water. This project will approach the prediction using binary classifications and ternary classifications. One model will be made to predict whether a pump is functional or not functional. Another model will predict on three classes, whether the pump is functional, needs repairs, or doesn't work at all. Feature importance will be drawn from the final models.

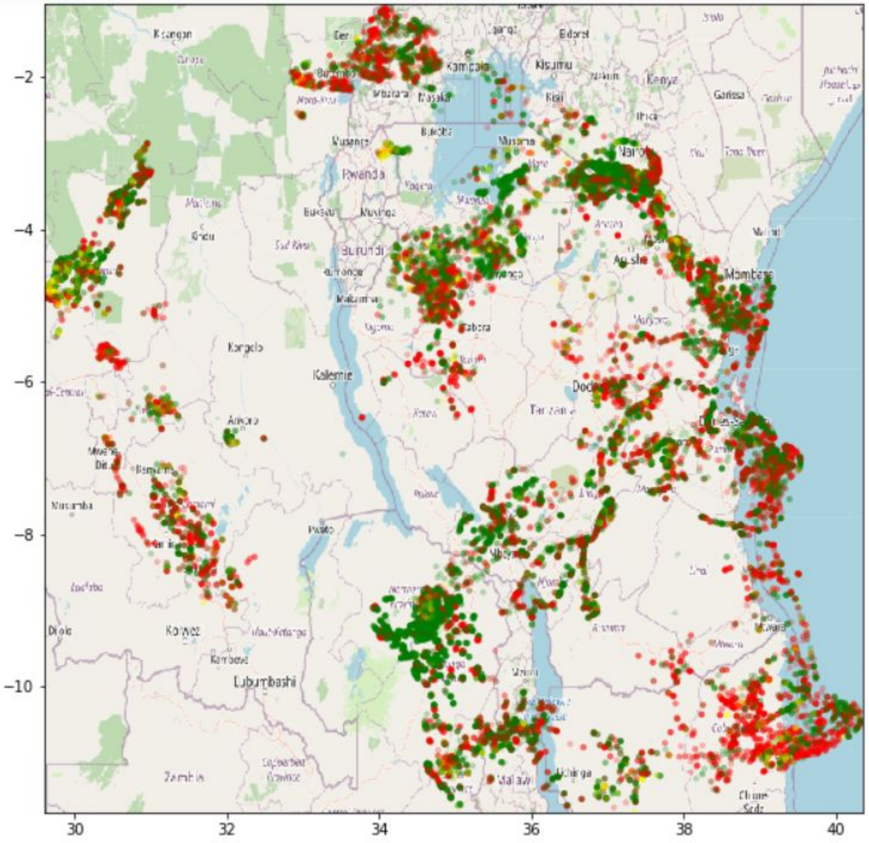
Data Used

- The data used in this project was provided by Taarifa and the Tanzania Ministry of Water.
- The data was found on drivendata.org;

https://www.drivendata.org/competitions/7/pump-it-up-data-mining-the-water-table/page/23/

- The original target class is ternary with about 54% of the data representing functional pumps, 38% representing non functional pumps, and 7% representing pumps that are functional but need repair.
- For the binary model the non functional pumps and pumps needing repair were combined.

EDA



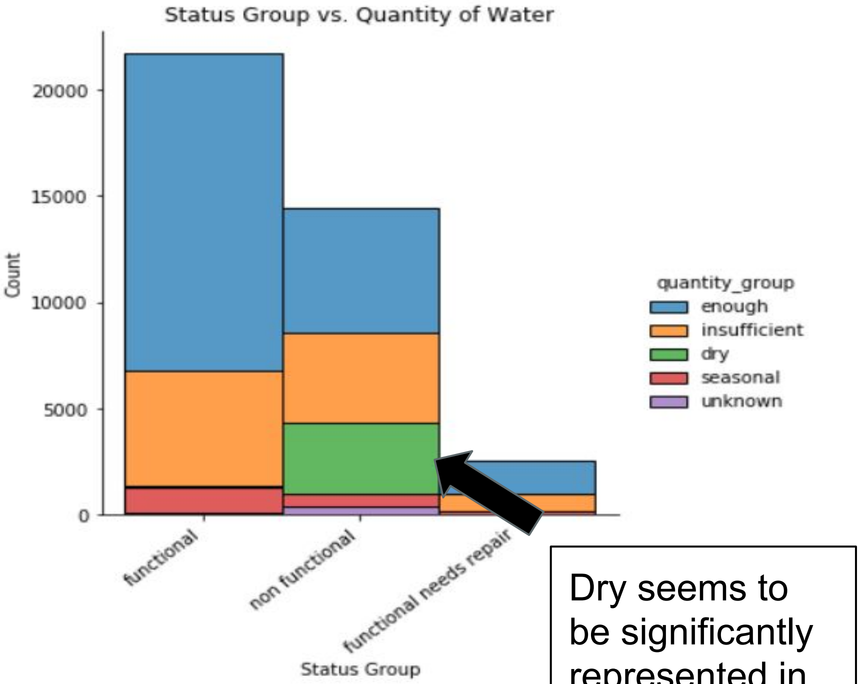
🔴 Non Functional Water Pumps

🟢 Functional Water Pumps

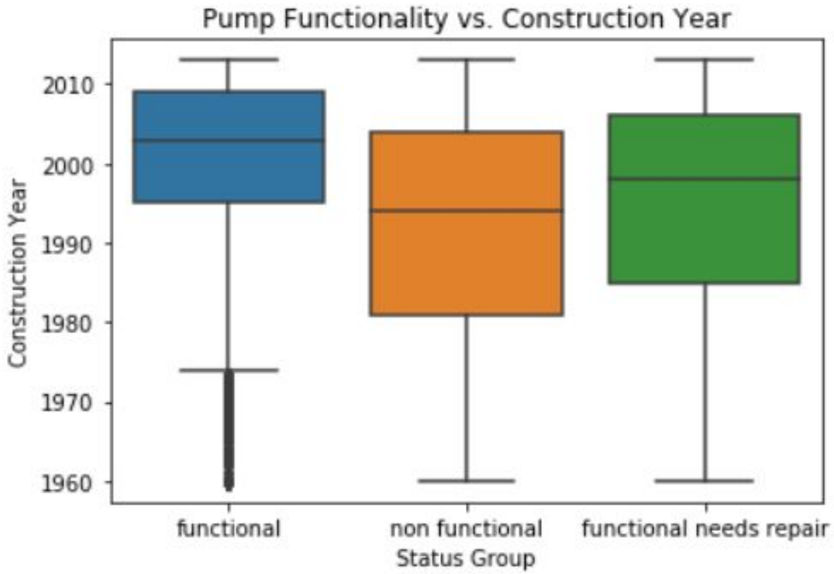
🟡 Needs Repair

Location may have an impact on functionality.

EDA Cont.



Dry seems to be significantly represented in the non functional group



Construction year looks to have some correlation to the functionality of the pump

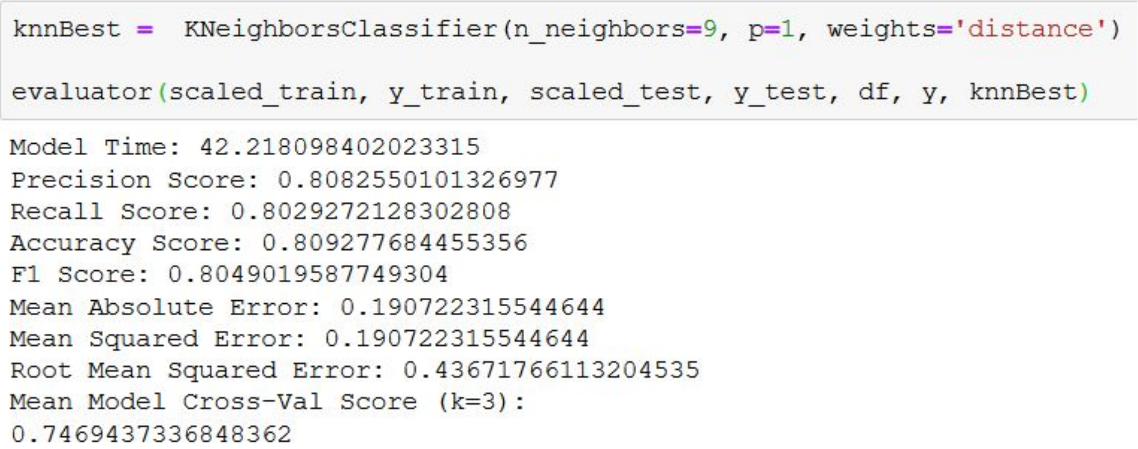
Final Models

- K nearest neighbors model works by finding the distances between a query and the examples in the data, using the specified number of examples closest to the query, and predicts the classification by voting for the most frequent label.

- A decision tree model breaks down a data set into smaller and smaller subsets while an associated decision tree is incrementally developed.

Final Models

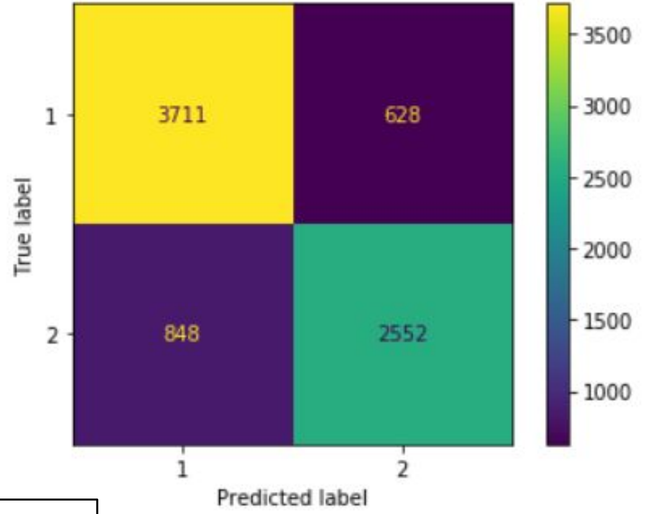
Binary KNN



```
knnBest =  KNeighborsClassifier(n_neighbors=9, p=1, weights='distance')

evaluator(scaled_train, y_train, scaled_test, y_test, df, y, knnBest)
```

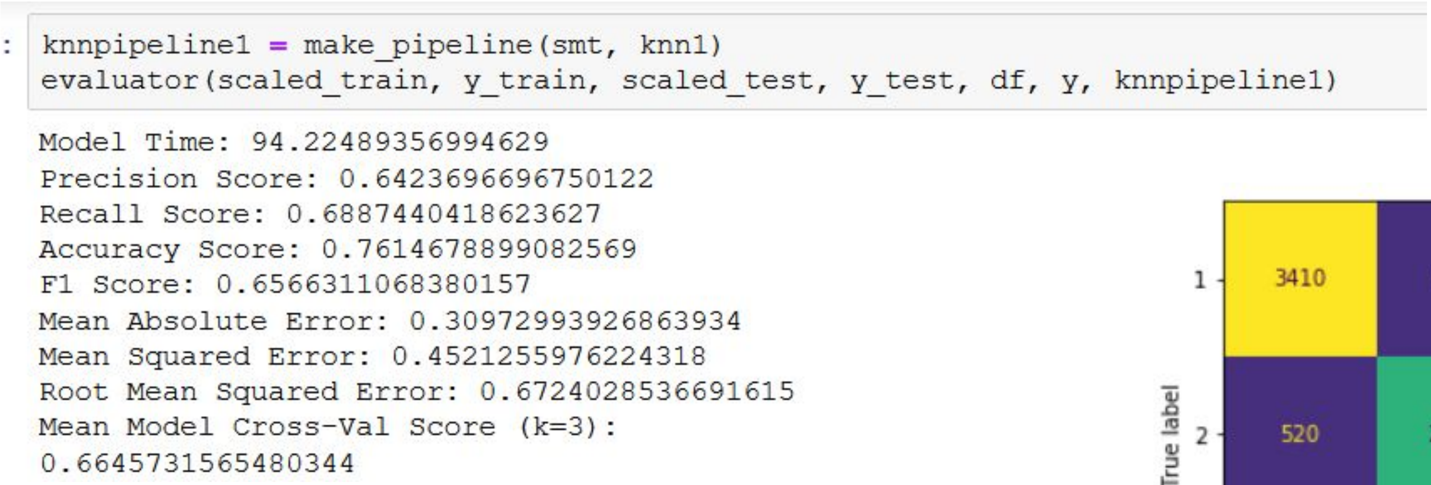
```
Model Time: 42.218098402023315
Precision Score: 0.8082550101326977
Recall Score: 0.8029272128302808
Accuracy Score: 0.809277684455356
F1 Score: 0.8049019587749304
Mean Absolute Error: 0.190722315544644
Mean Squared Error: 0.190722315544644
Root Mean Squared Error: 0.43671766113204535
Mean Model Cross-Val Score (k=3):
0.7469437336848362
```



Compared to Final Binary Decision Tree Model:
Precision Score: 0.7902741459154872
Recall Score: 0.7861569147133387
Accuracy Score: 0.792221217211526
F1 Score: 0.7877357023008846

Final Models Cont.

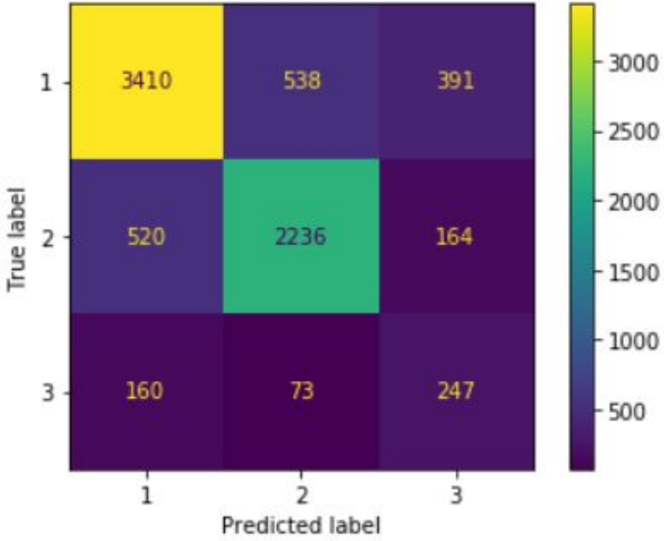
Ternary KNN Model



```
: knnpipeline1 = make_pipeline(smt, knn1)
  evaluator(scaled_train, y_train, scaled_test, y_test, df, y, knnpipeline1)
```

```
Model Time: 94.22489356994629
Precision Score: 0.6423696696750122
Recall Score: 0.6887440418623627
Accuracy Score: 0.7614678899082569
F1 Score: 0.6566311068380157
Mean Absolute Error: 0.30972993926863934
Mean Squared Error: 0.452125597622431 8
Root Mean Squared Error: 0.6724028536691615
Mean Model Cross-Val Score (k=3):
0.664573156548034 4
```

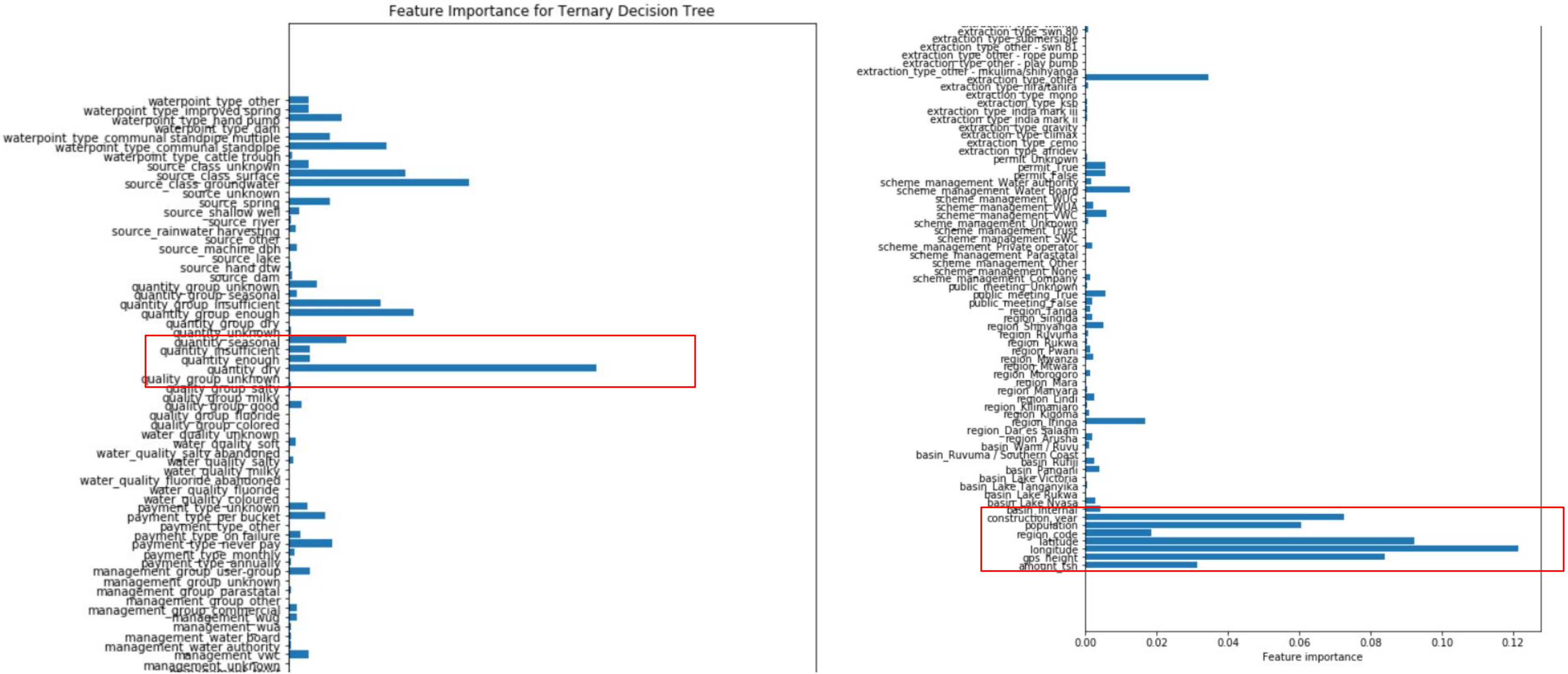
Compared to Final Ternary Decision Tree Model:
Precision Score: 0.6539190665359614
Recall Score: 0.6430726740198889
Accuracy Score: 0.7719343584442434
F1 Score: 0.6480633873631786



Interpretation of Models Cont.

- Both models had specific strengths and weaknesses.
    - Binary KNN model had the highest scores overall.
    - Ternary KNN model had higher recall and F1 while Ternary DT model had higher precision and accuracy.
        - This means that the KNN model was better at finding all the positive samples, a lower false negative rate, whereas the DT model had a lower false positive rate. However the KNN model had the best ratio of recall and precision represented in the F1 score.
        - As explained on scikit-learn.org "*A system with high recall but low precision returns many results, but most of its predicted labels are incorrect when compared to the training labels. A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct when compared to the training labels.*"
        - The KNN model was better at predicting pumps that were functional but needed repair, which was a more difficult target to classify due to the class imbalance.

Feature Importance


Feature Importance for Ternary Decision Tree

Conclusions

- The KNN model performed the best on the binary classification data in all scores
- The ternary KNN model had the best F1 score, the ratio between precision and recall, and predicted pumps that were *functional but need repair* the best. F1 score represents the balance between returning many results and all the results labeled correctly.
- Some of the features with the most importance for the decision tree model included;
  - Dry water quantity
  - Location
  - Year constructed

Next Steps

- Further feature analysis could be done such as grouping similar features to create a smaller set of data
- This project used SMOTE to address class imbalance, however I would also look into trying other methods to address class imbalance such as under sampling.

Thank You

Check out my github repo here:
https://github.com/newhousem/WaterWellProject

Contact me: meredithnewhouse@gmail.com

Thank you to DrivenData for providing the data sets used in this analysis and Yish for helping to answer all of my questions.