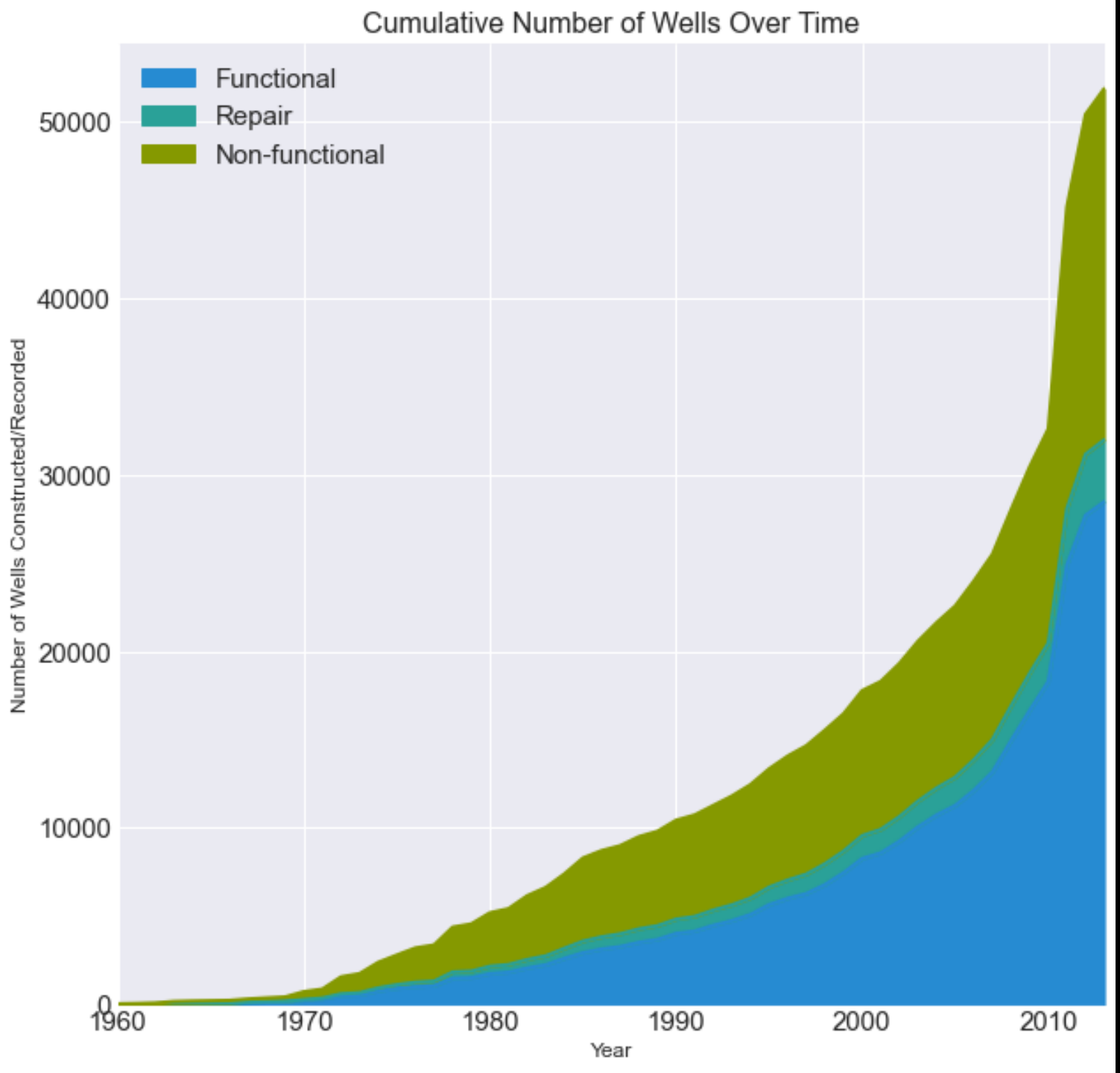
How well are the wells?

Predicting Tanzanian Water Wells Functionality

Sung Bae

Tanzanian Water Crisis Facts

- Importance of water
  - Source of life
  - Improve life quality
  - Increase school attendance
  - Empowers families
- 24 million people without basic access to safe water
- Numerous organizations have been working to provide safe and accessible water



Tanzanian Water Crisis Facts

- More than 50,000 water wells are installed by 2013
  - 54.9 % Functional
  - 38.3 % Non-functional
  - 6.8 % Needs repair
- Imperative to determine which wells are not functioning or need repairing accurately

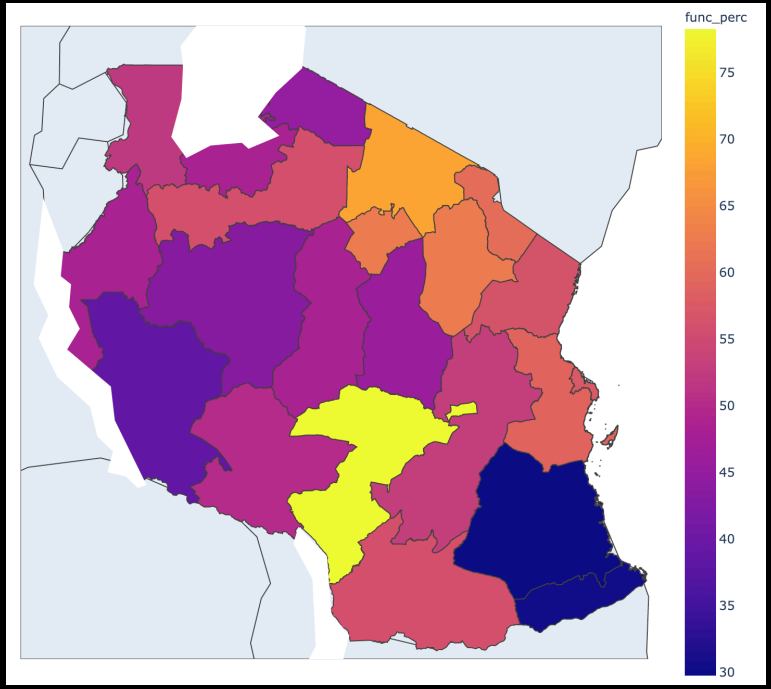
Goals and Objectives

- Construct a model that can classify
  [1] functioning wells,
  [2] functioning but need repairing wells, and
  [3] non-functioning wells.

- Top Priorities
  → Recalls for [2] and [3]
  → Overall accuracy

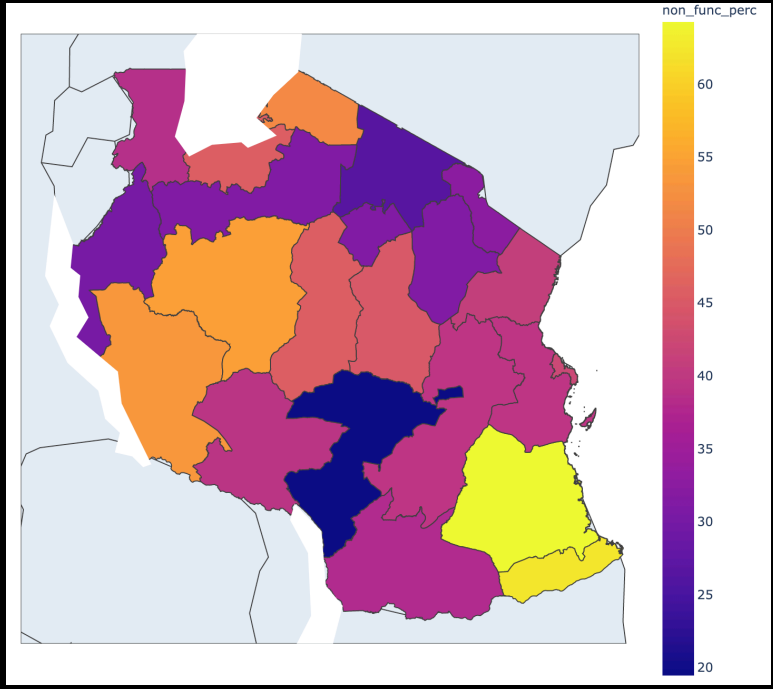
Geographical Factors

Percent Functioning Wells



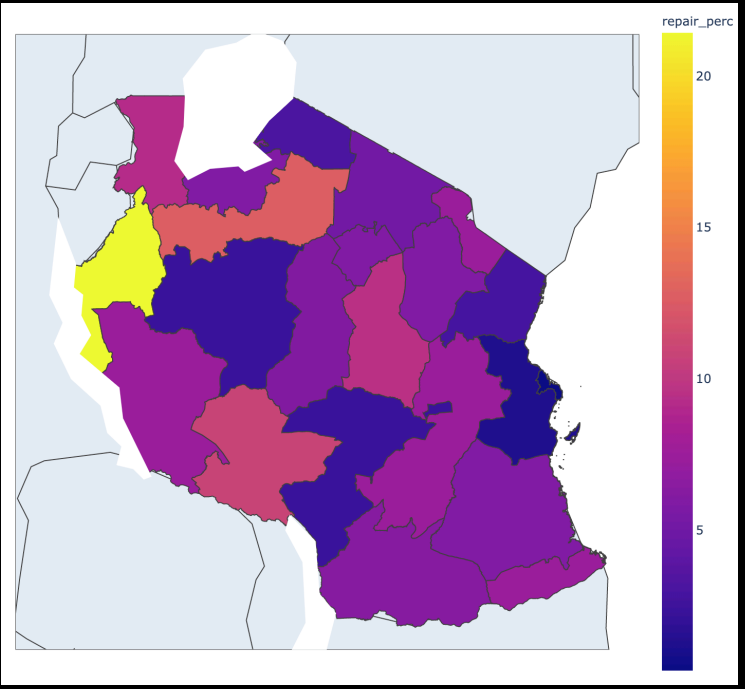
Region: Iringa
N = 5294
Percentage: 78.22%

Percent Non-Functioning Wells



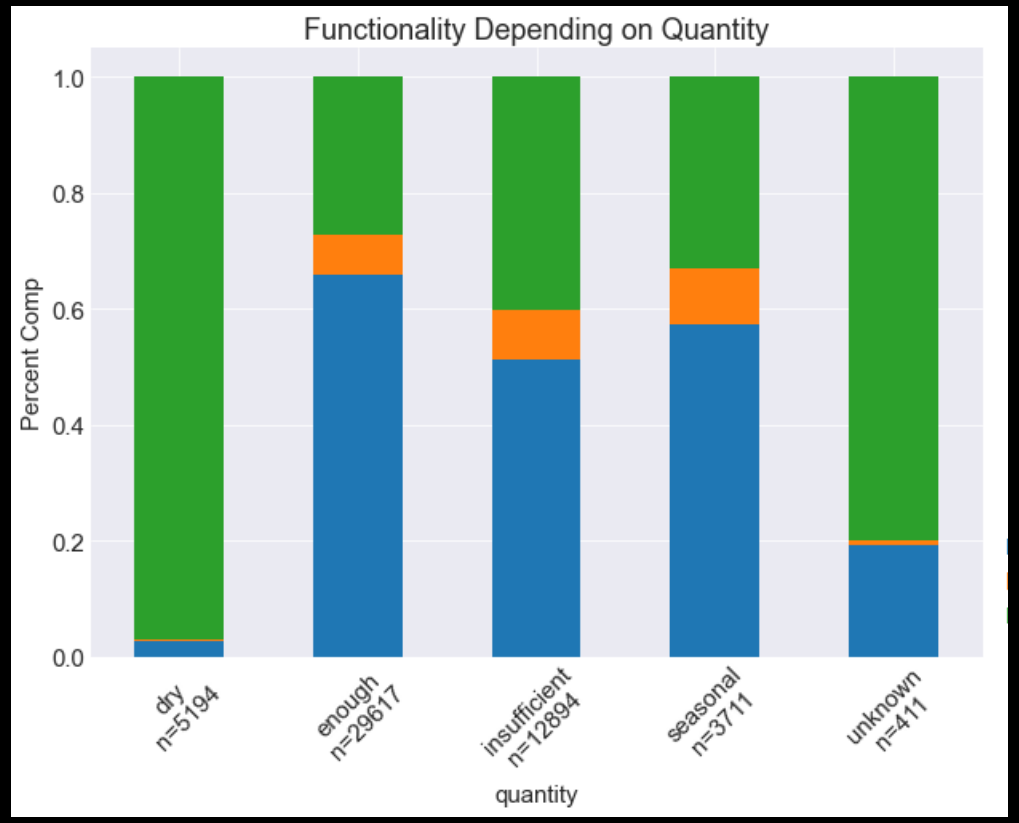
Region: Lindi
N = 1546
Percentage: 64.23%

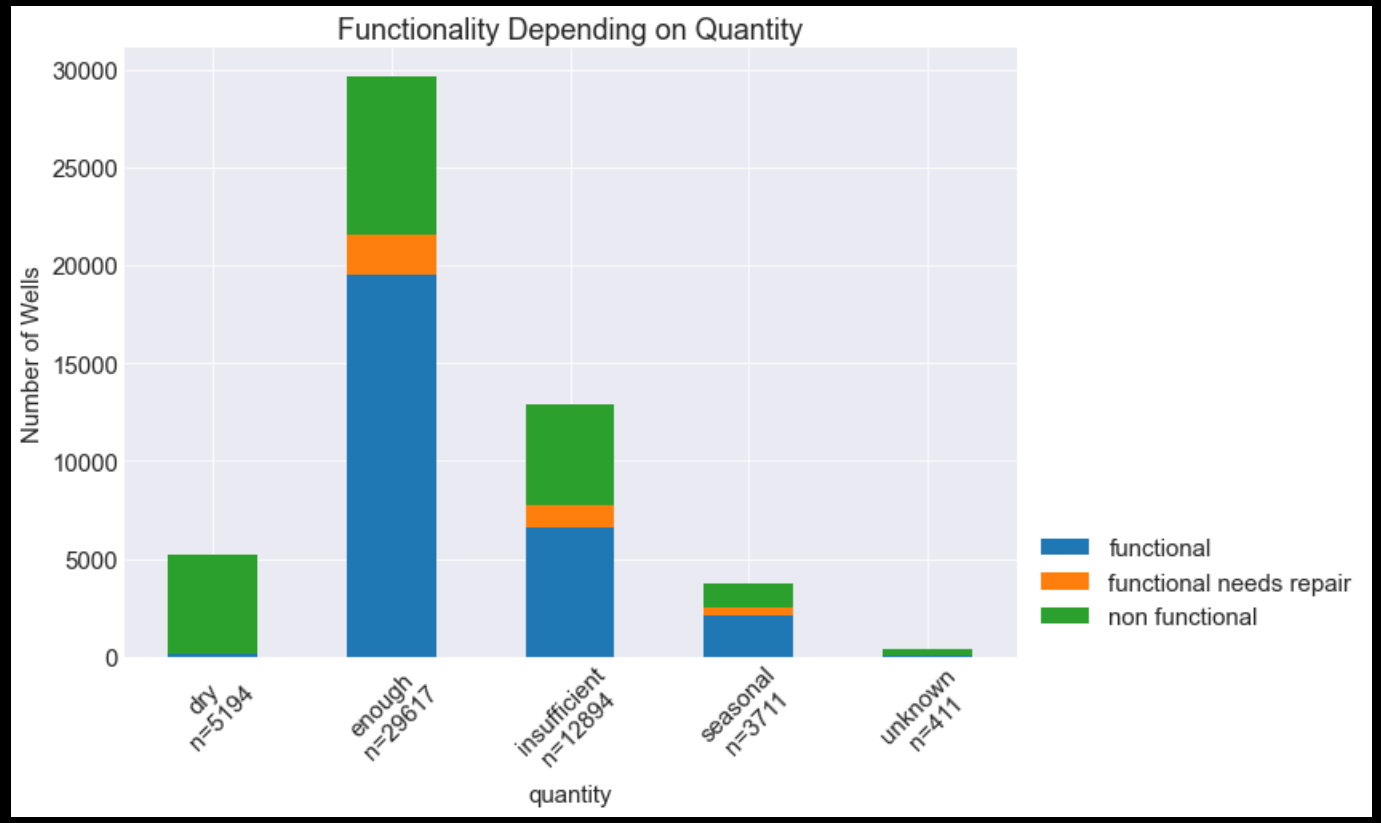
Percent Need-Repairing Wells



Region: Kigoma
N = 2816
Percentage: 21.41 %

Quantity Factors



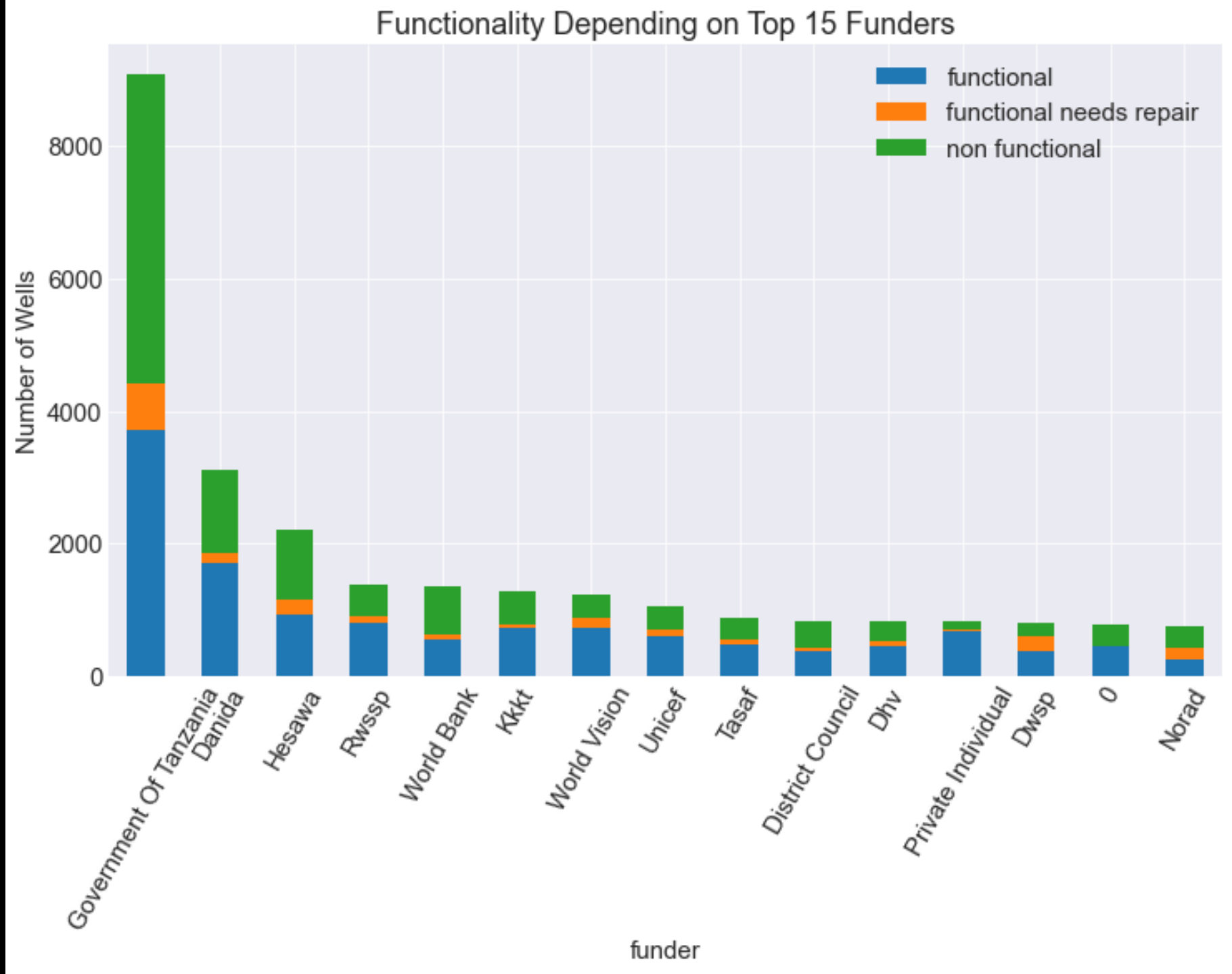


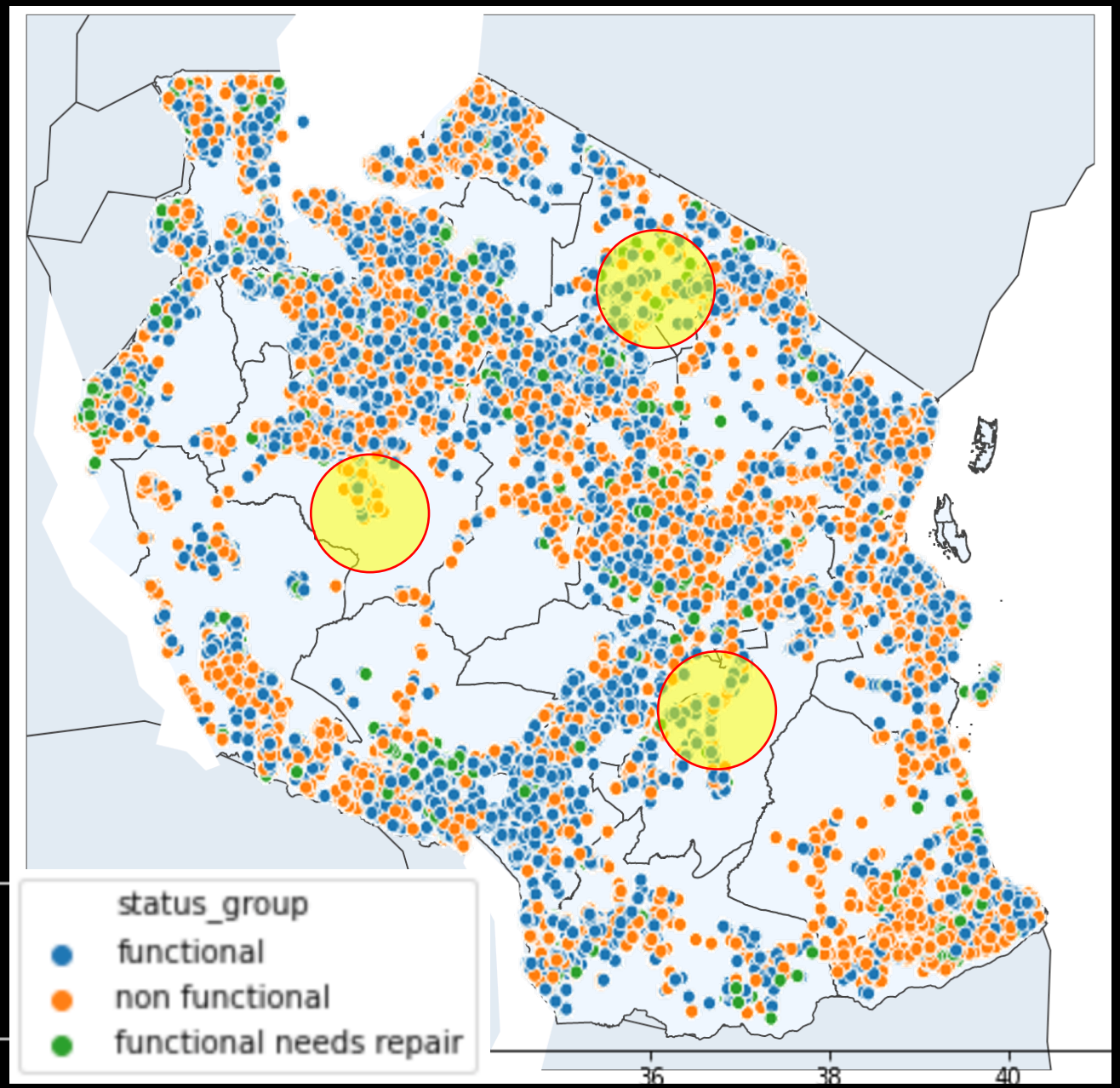
More than 90% non-functional wells for ”dry quantity” wells.

More than 80% non-functional wells for ”unknown quantity” wells.

Funders Factor

- More than 50% of wells funded by the government are either not functioning or need repair





Neighboring

Probability of finding

[1] functioning

[2] need repairing

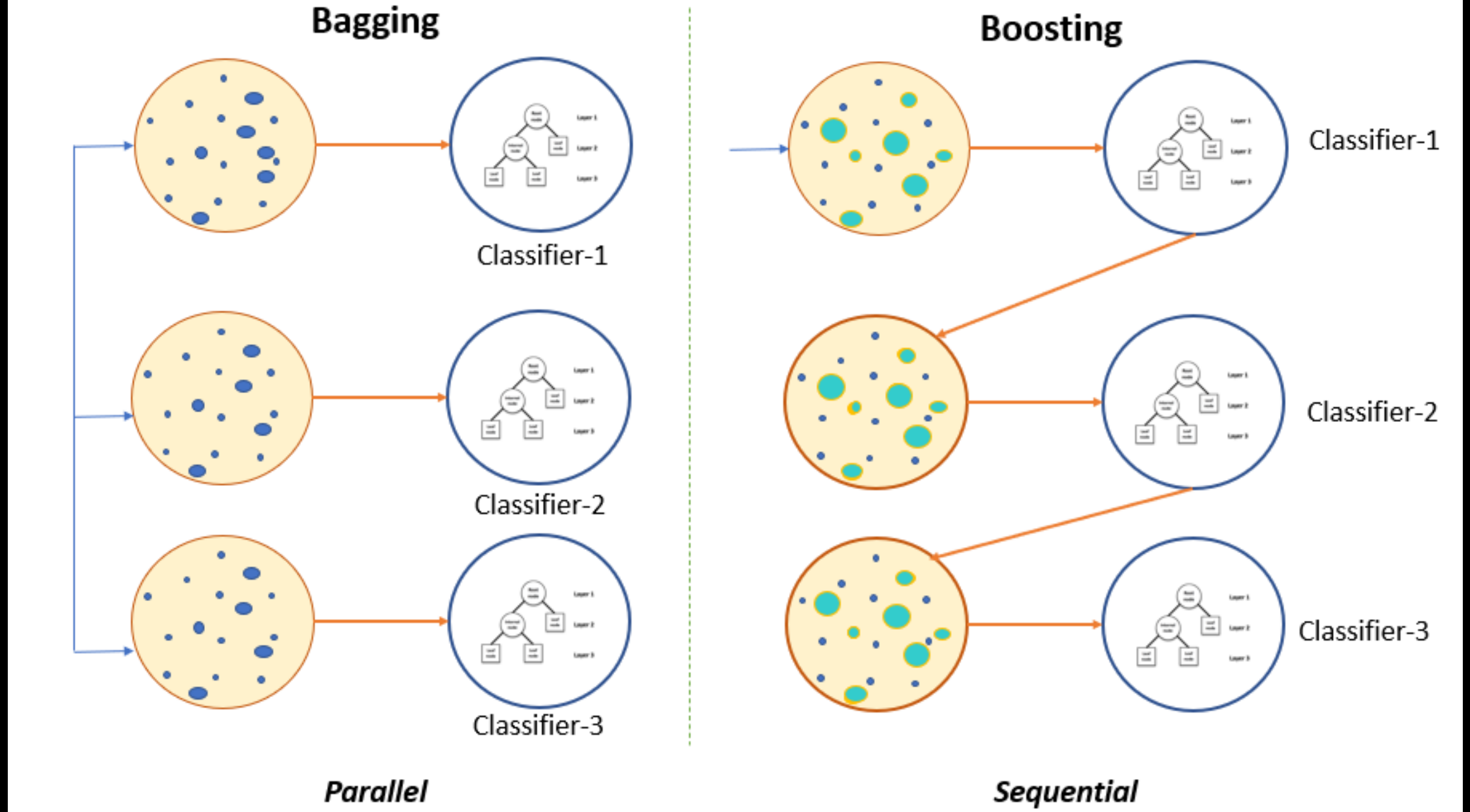
[3] non-functioning

are calculated within 30 km of each water well

Our Plan

Bagging	*Boosting*

- Decreases model's variance
- Examples
  - Random Forest
  - Extra Trees
	- Decreases model's bias
	- Examples
	  - XGBoost
	  - Adaboost
	  - Gradient Boost

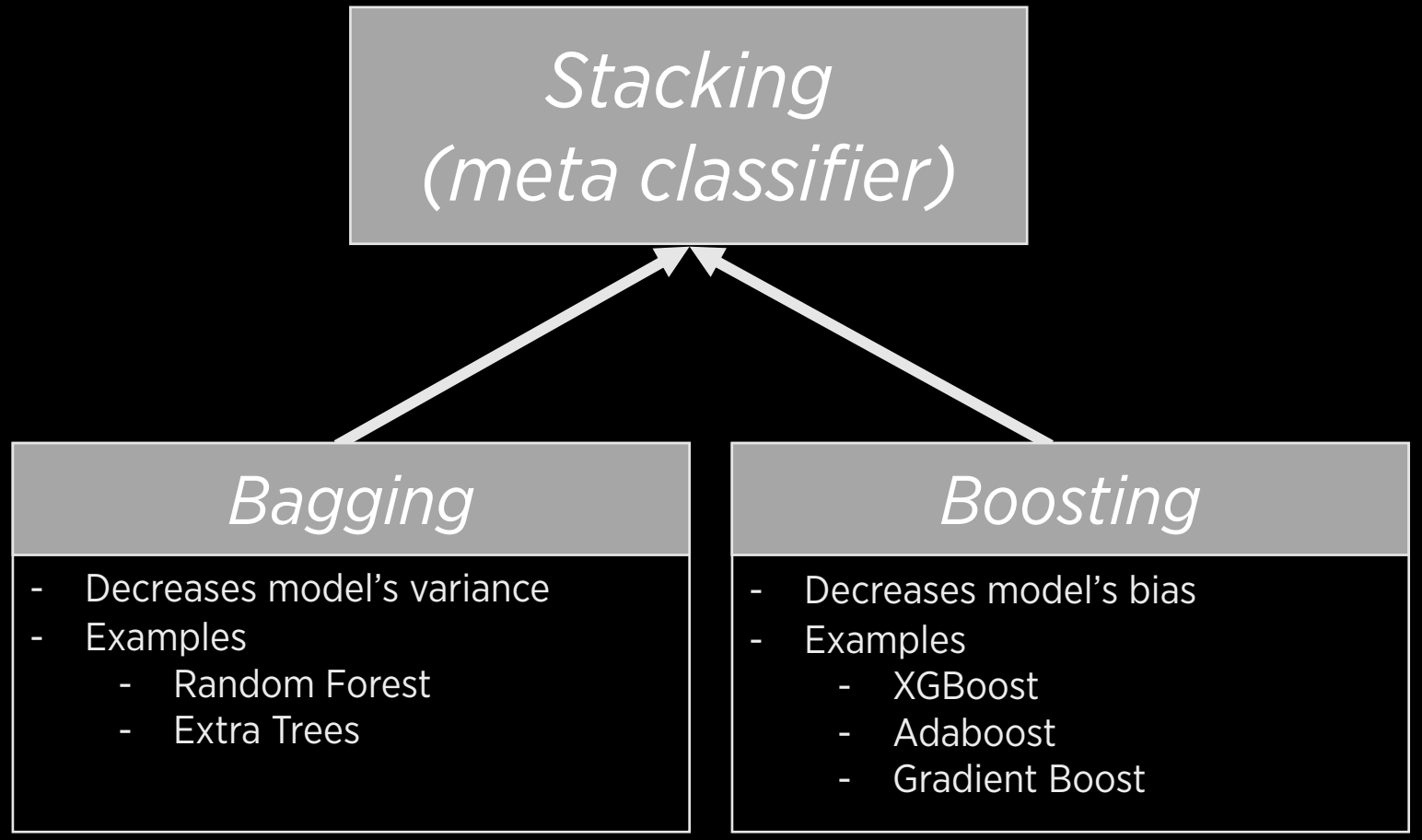

Bagging
Classifier-1
Classifier-2
Classifier-3
Parallel
Boosting
Classifier-1
Classifier-2
Classifier-3
Sequential

Our Plan

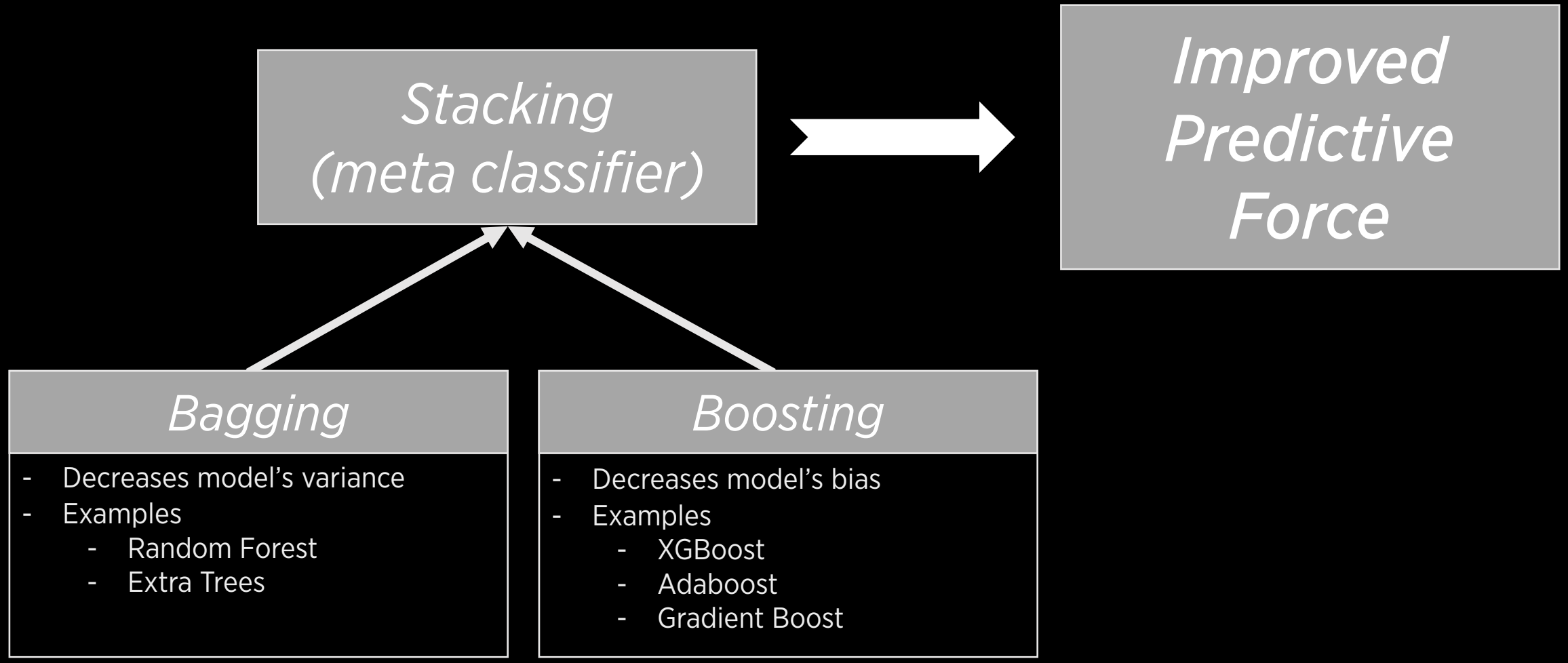
Bagging	*Boosting*

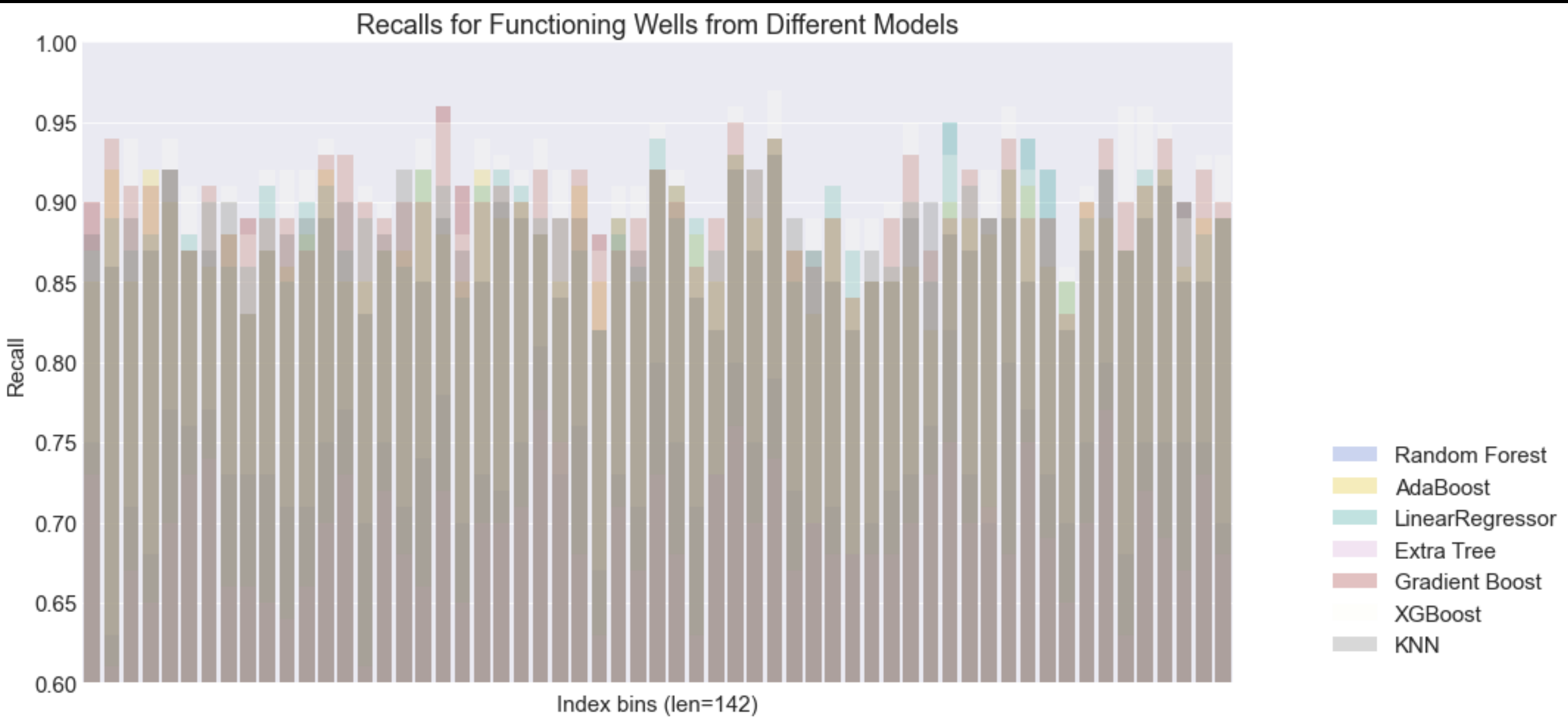
- Decreases model's variance
- Examples
  - Random Forest
  - Extra Trees
	- Decreases model's bias
	- Examples
	  - XGBoost
	  - Adaboost
	  - Gradient Boost

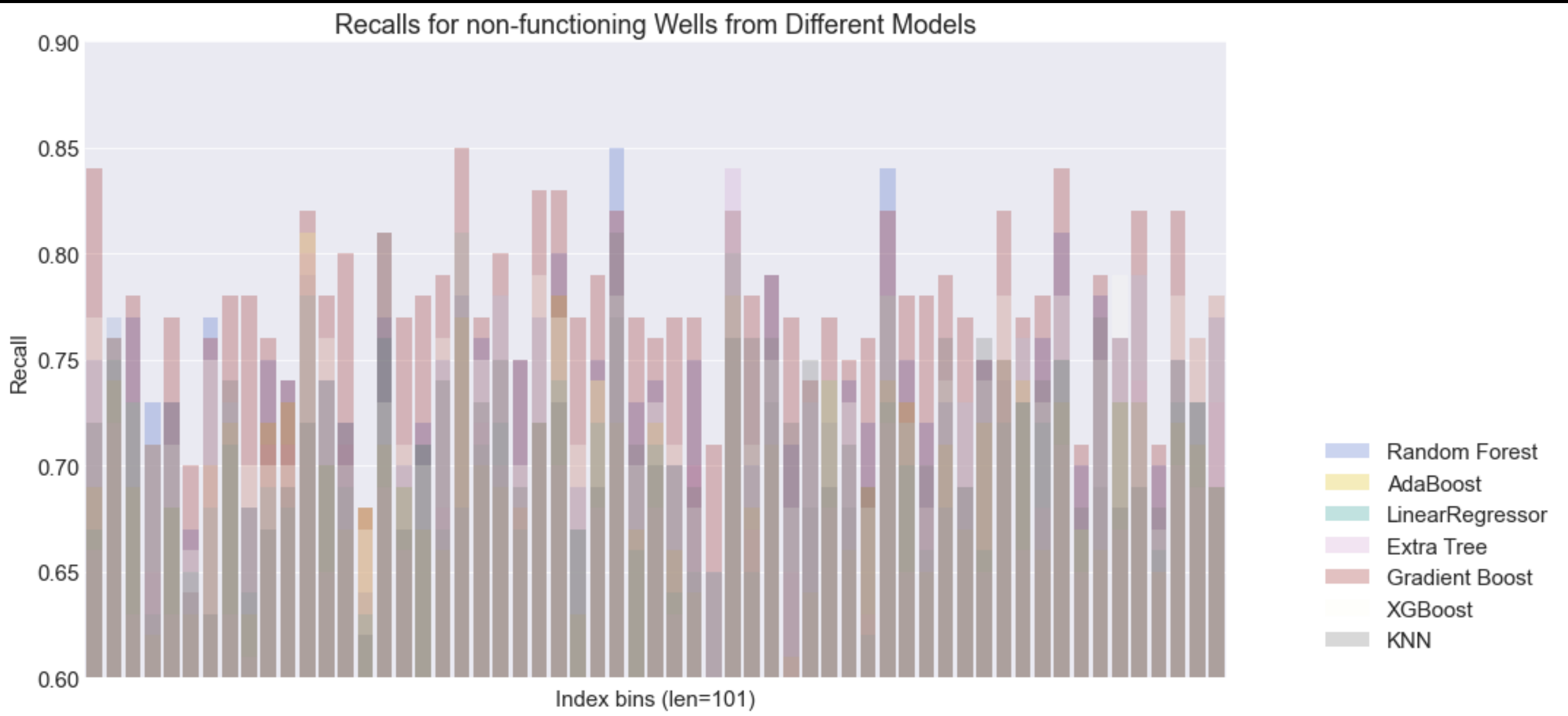
Out Plan

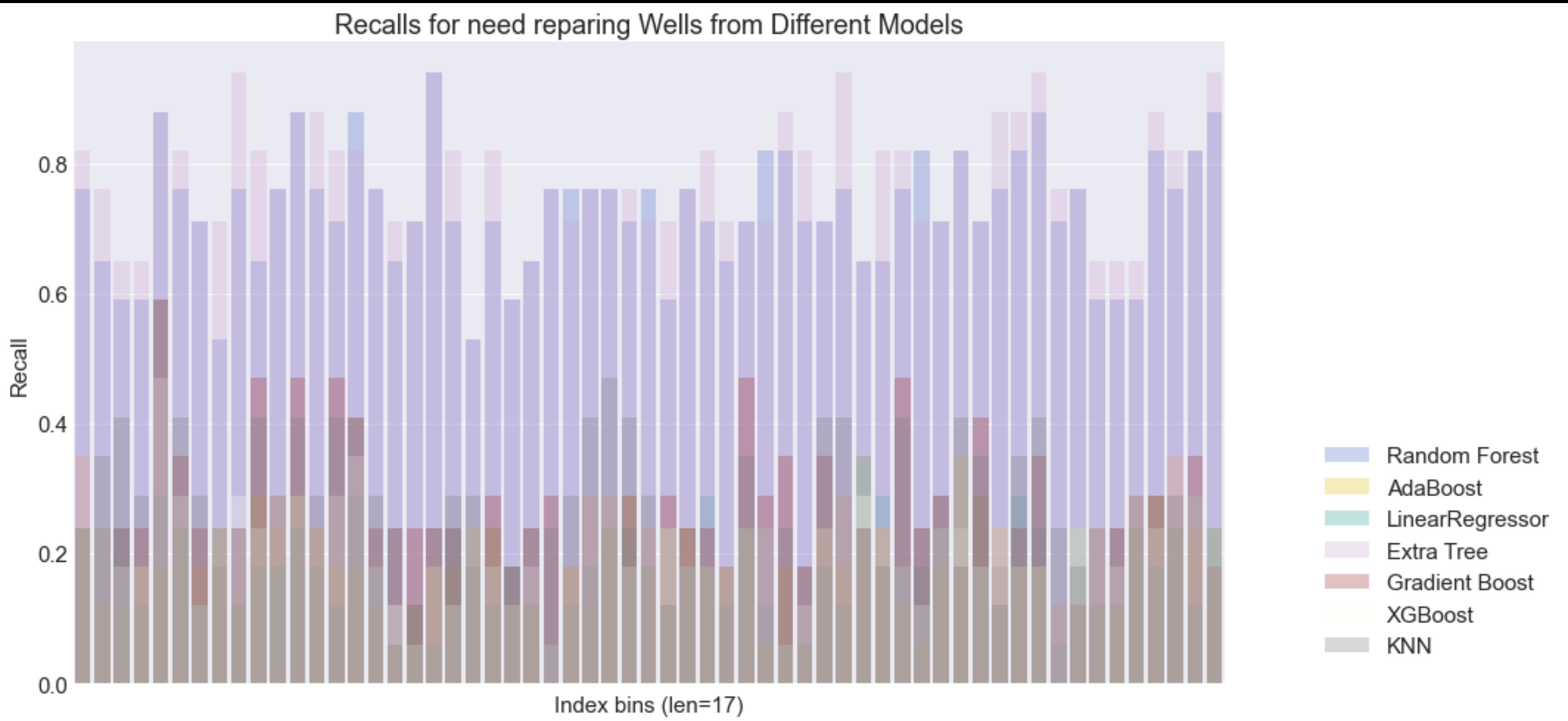

Stacking
(meta classifier)
Bagging
- Decreases model's variance
- Examples
  - Random Forest
  - Extra Trees
Boosting
- Decreases model's bias
- Examples
  - XGBoost
  - Adaboost
  - Gradient Boost

Out Plan


Stacking (meta classifier)
Improved Predictive Force
Bagging
- Decreases model's variance
- Examples
  - Random Forest
  - Extra Trees
Boosting
- Decreases model's bias
- Examples
  - XGBoost
  - Adaboost
  - Gradient Boost


Recalls for Functioning Wells from Different Models
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Recall
Index bins (len=142)
Random Forest
AdaBoost
LinearRegressor
Extra Tree
Gradient Boost
XGBoost
KNN


Recalls for non-functioning Wells from Different Models
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Recall
Index bins (len=101)
Random Forest
AdaBoost
LinearRegressor
Extra Tree
Gradient Boost
XGBoost
KNN


Recalls for need reparing Wells from Different Models
Recall
0.0
0.2
0.4
0.6
0.8
Index bins (len=17)
Random Forest
AdaBoost
LinearRegressor
Extra Tree
Gradient Boost
XGBoost
KNN

Best Accuracy Model: Adaboost

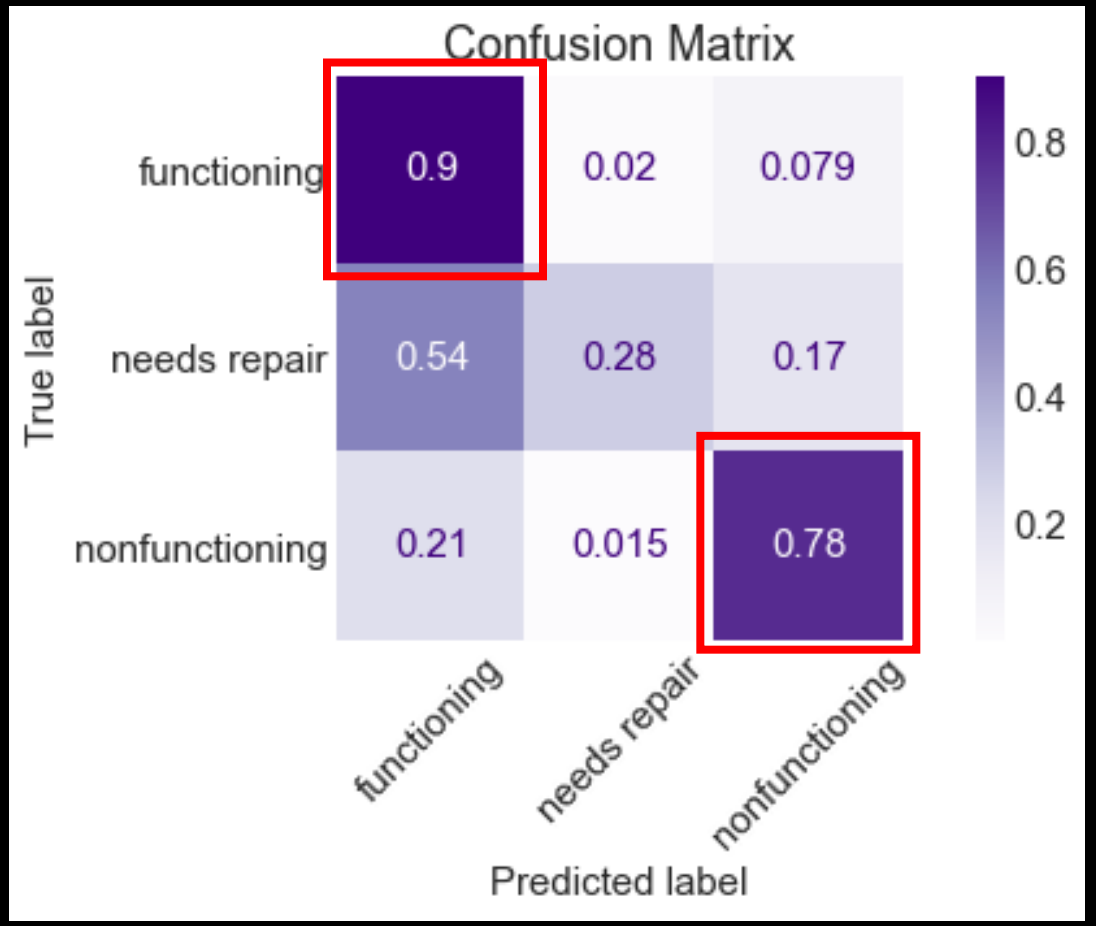
```
----------------------------------------------------------
[i] CLASSIFICATION REPORT
----------------------------------------------------------
Train Accuracy :  0.9348
Test Accuracy :  0.8132
Train AUC :  0.9497
Test AUC :  0.8471
----------------------------------------------------------

                precision    recall  f1-score   support

   functioning       0.81      0.90      0.85      8490
  needs repair       0.53      0.28      0.37      1019
nonfunctioning       0.85      0.78      0.81      6040

      accuracy                           0.81     15549
     macro avg       0.73      0.65      0.68     15549
  weighted avg       0.81      0.81      0.81     15549

----------------------------------------------------------
```



[1] Possibly overfit [2] Accuracy = 81% [3] Test Accuracy = 79% [4] Low Recalls

[5] 1st Layer Models: All the first layer models were used

Best Recall Model: Random Forest

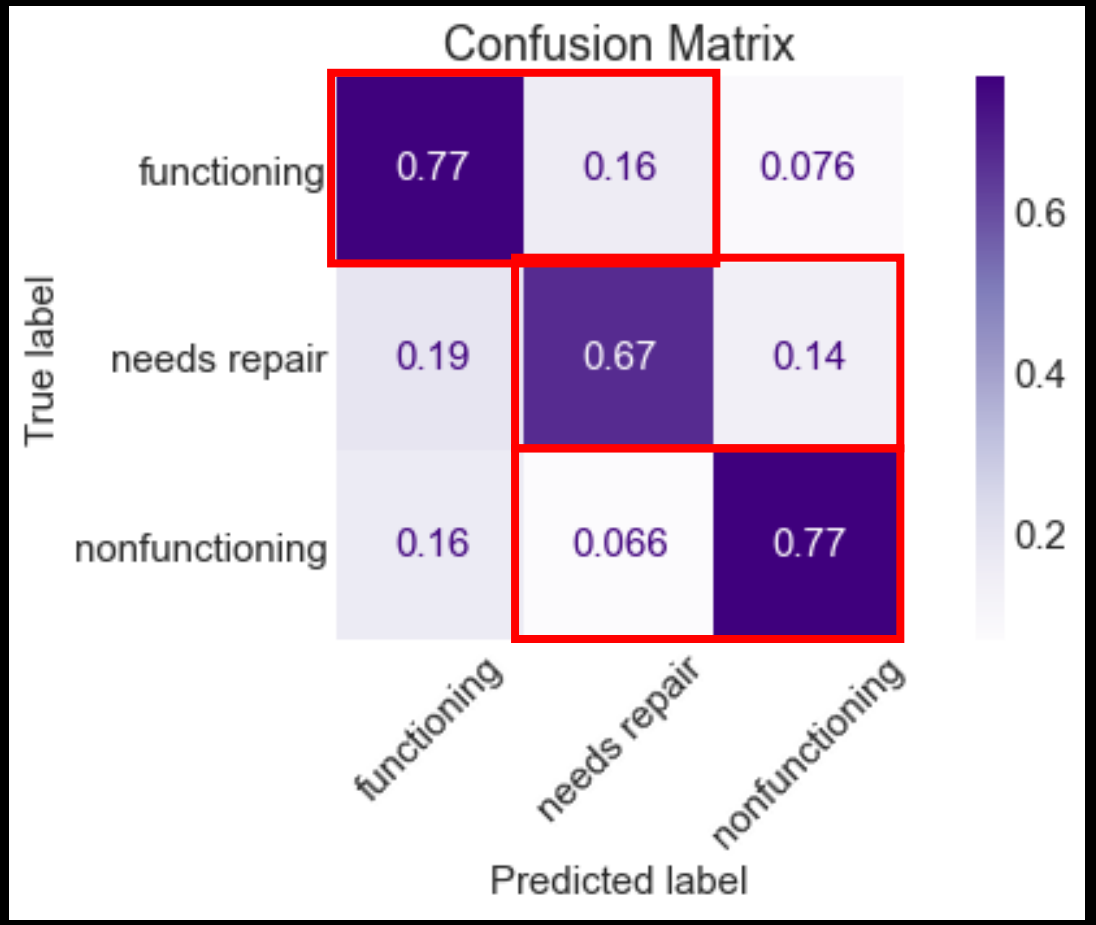
```
-------------------------------------------------------
[i] CLASSIFICATION REPORT
-------------------------------------------------------
Train Accuracy :  0.8479
Test Accuracy :  0.762
Train AUC :  0.9384
Test AUC :  0.8729
-------------------------------------------------------

                precision    recall  f1-score   support

   functioning       0.85      0.77      0.80      8490
  needs repair       0.28      0.67      0.40      1019
nonfunctioning       0.86      0.77      0.81      6040

      accuracy                           0.76     15549
     macro avg       0.66      0.74      0.67     15549
  weighted avg       0.81      0.76      0.78     15549

-------------------------------------------------------
```

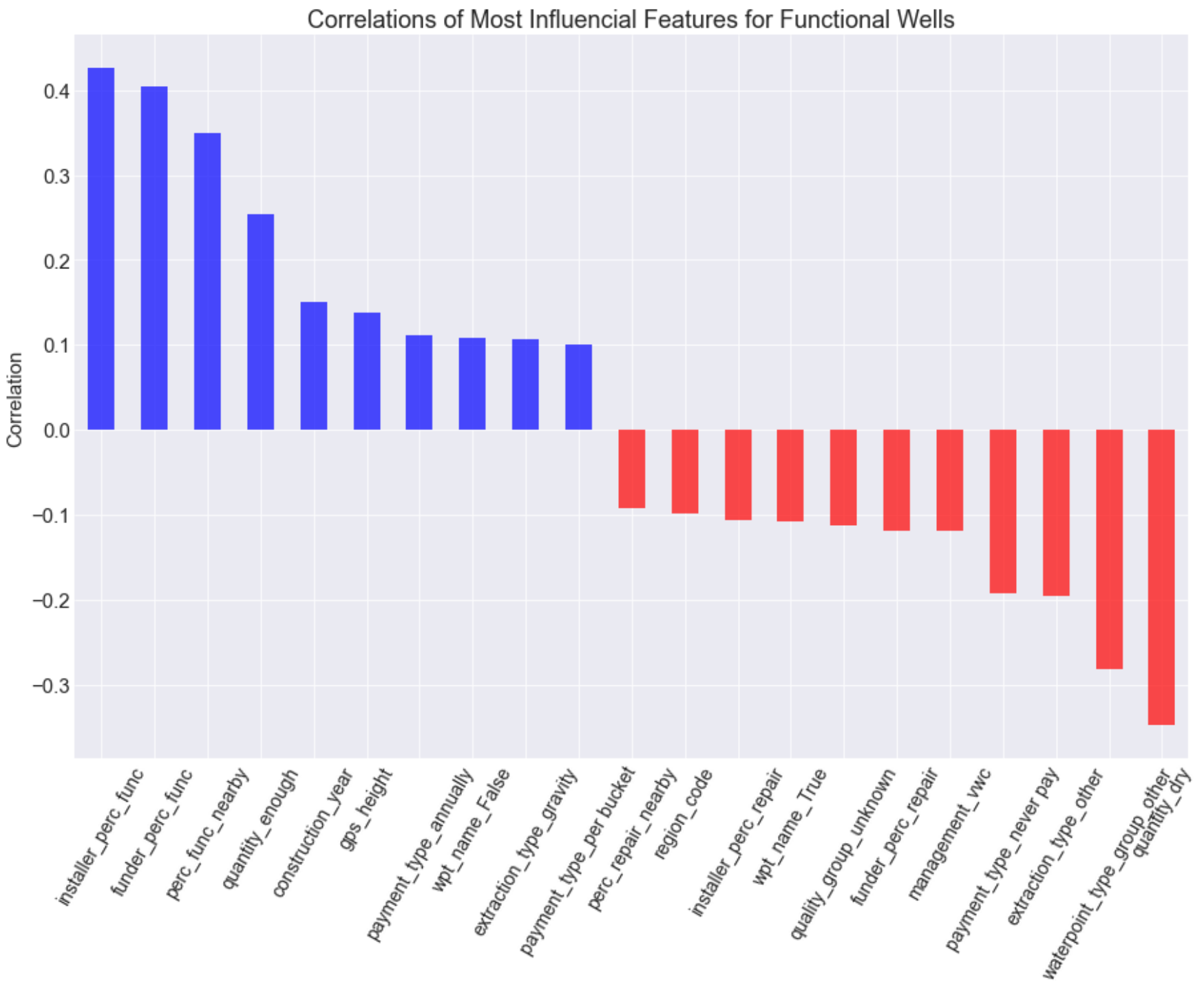


[1] Recall = 74% [2] Accuracy = 76% [3] Test Accuracy = 70% [4] Effective Recalls = 83%, 81%, 84%

[5] 1st Layer Models: Gradient boost, Random Forest, KNN, Adaboost, and Logistic Regression

Interpreting the Model

- Most positive:
  - Installer
  - Funder
  - Neighbor
  - Payment
  - Extractor type
- Most Negative:
  - Quantity
  - Extractor type
  - Management
  - Neighbor



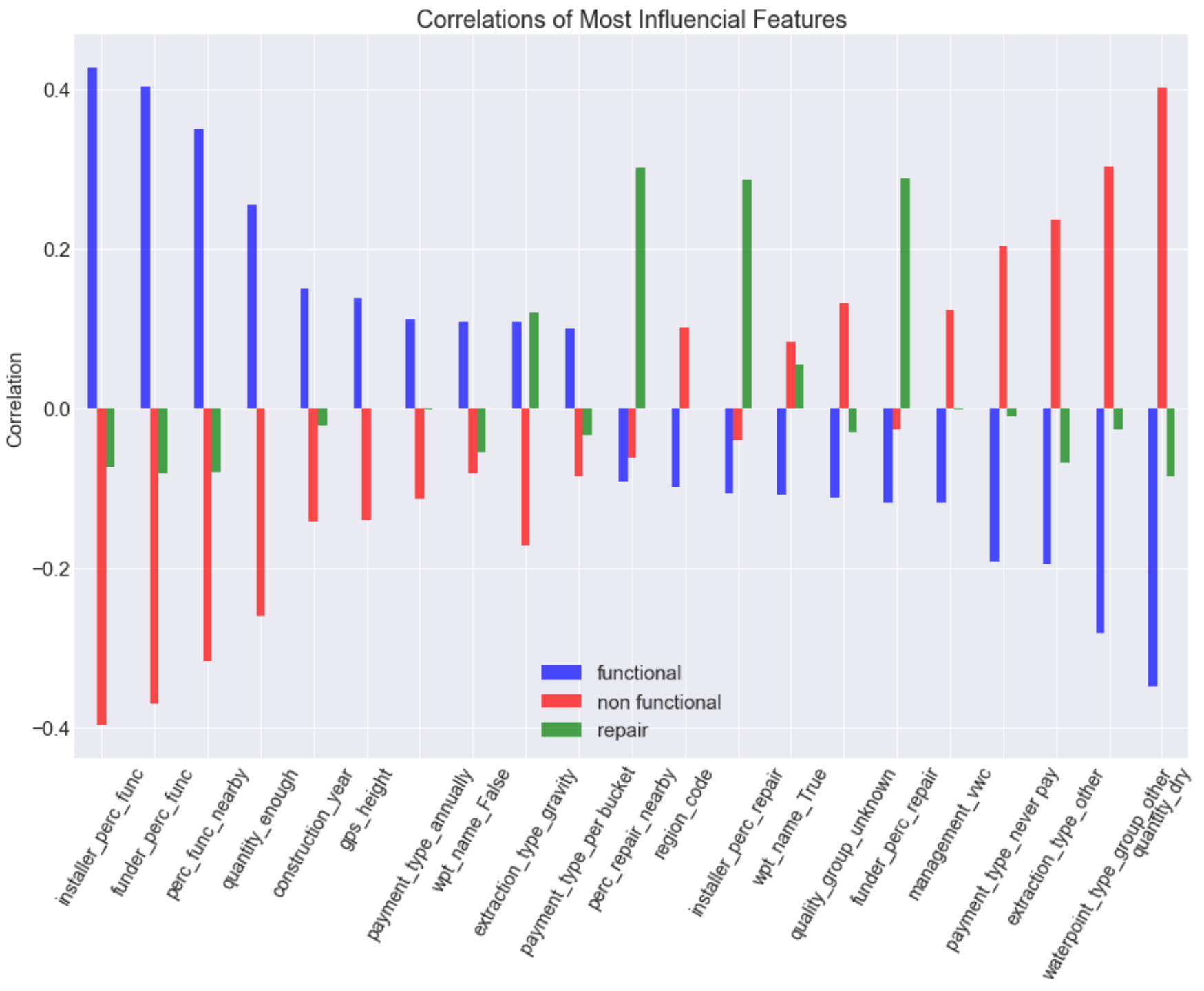
Interpreting the Model

- Expected

Functioning and non-functioning have opposite correlation

- New Finding

Repair is all over the places which makes the prediction extra tough



Conclusion

- Our final models can
  - Predict with 79% accuracy (Adaboost – 7 1st layers)
  - Predict with 82% recall (Random Forest – 5 1st layers)
- Important features
  - Positive
    - Installer, funder, neighbor– high functioning percentage
    - Payment - existence
    - Extractor type - gravity
  - Negative
    - Quantity - dry
    - Extractor type – other than gravity
    - Management – VWC (village water committee)
    - Payment – Lack of payment

Future Studies

- Further hyperparameter tuning can be done for each model used
- Different combinations of ensemble models can be tested
- Different methods of dealing with class imbalance can be used
  - under sampling
  - different class weights

Thank you for listening

Appendix