

# DataDrip

Water Pumps functionality prediction (Sprint 2)

Shakeel Ahmed  
Amey Chitnis

# Overview

- Features processing (Implemented using Pipeline structure)
- Baseline model
- Model results
- Performance metrics

# Features processing

## Built in Transformers:

- **MinMaxScaler()**: For feature normalization between 0 and 1
- **OneHotEncoder()**: For categorical variables
- 

## Customized Transformers:

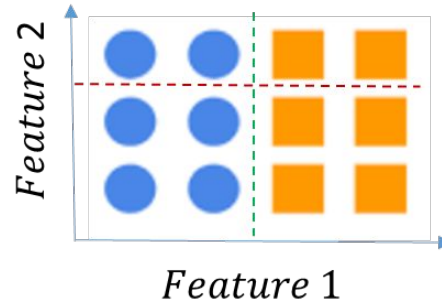
- **IQRCappper()**: Interquartile range is used to remove outliers and replace them with mean median, clipping
- **YearExtractor()**: For the date column
- **StringConverter()**: To treat each category in categorical feature as label even if it's a number. Gives consistent ohe behavior
- **ConstructionYearTransformer()**: To replace the 0 values in the construction year with median

# Baseline model

**Decision Tree** = One tree makes the decision.

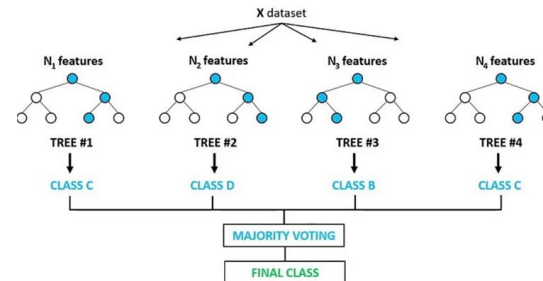
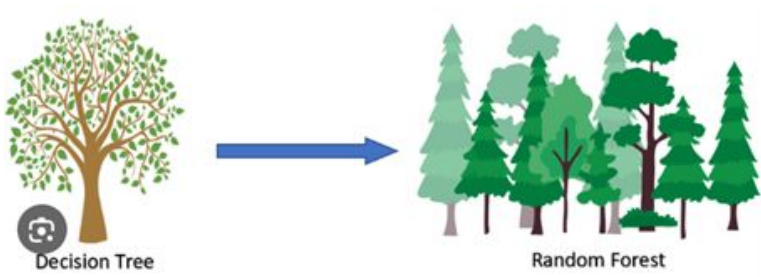
## Important Questions

- Which feature to start with
- Split value of feature



**Objective:** Minimize Impurity or Uncertainty

**Random Forest** = A forest of trees votes on the decision, and the majority wins.



# ML Model

Baseline: random forest, decision tree, xgboost

Additional feature processing:

- GeoContextImputer()
- SMOTE()

Random forest selected

- `n_estimators = 300`
- `Class_weight = 'balanced'`

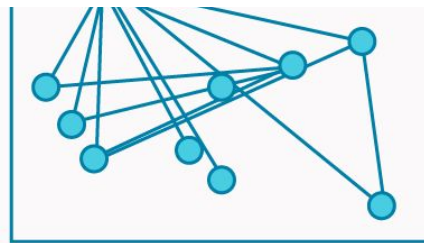
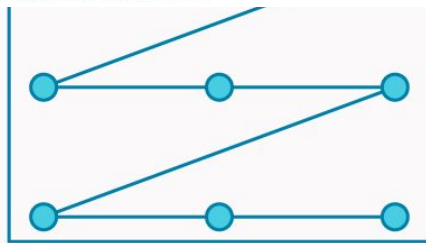
# Model Optimization

## GridSearchCV: Smaller parameter grid

```
Best parameters: {'classifier__max_depth': None, 'classifier__min_samples_split': 2, 'classifier__n_estimators': 300}  
Best cross-validation score: 0.8002244668911336
```

## RandomizedSearchCV: Larger parameter grid

```
Best parameters: {'classifier__bootstrap': True, 'classifier__max_depth': 40, 'classifier__max_features': 'sqrt', 'classifier__min_samples_leaf': 1, 'classifier__min_samples_split': 10, 'classifier__n_estimators': 620}  
Best cross-validation score: 0.7018965084294856
```



# Model Evaluation

```
from sklearn.metrics import accuracy_score









print(accuracy_score(y_test_encoded, y_pred))
```

0.8080808080808081

```
from sklearn.model_selection import cross_val_score

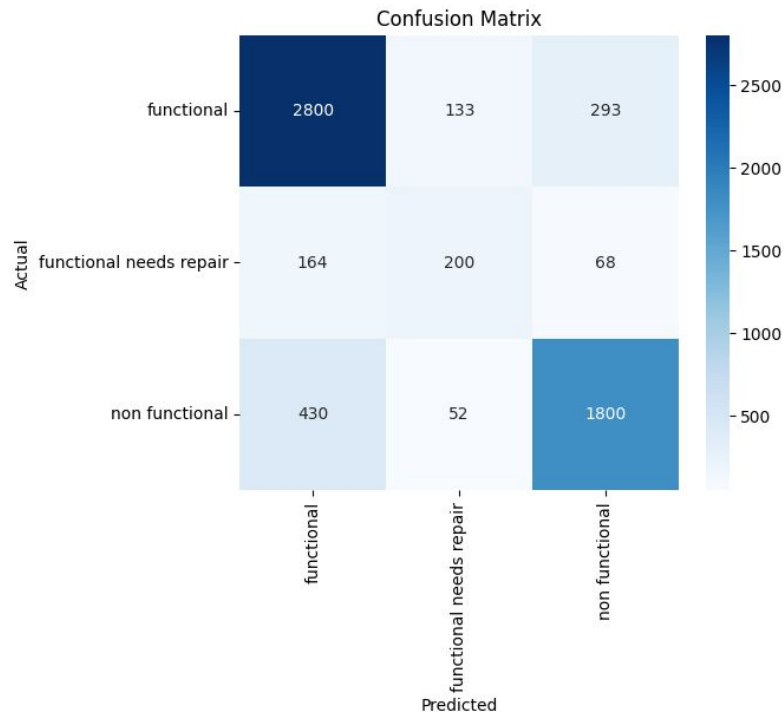
scores = cross_val_score(full_pipeline, X_train, y_train_encoded, cv=5, scoring='accuracy')
print("Average CV accuracy:", scores.mean())
```

Average CV accuracy: 0.7964272353161241

 Accuracy: 0.8080808080808081  
 Precision (macro): 0.7258044989981768  
 Recall (macro): 0.7065642188214794  
 F1 Score (macro): 0.7152602896554855  
 F1 Score (weighted): 0.8063079188461421  
 Cohen's Kappa: 0.6479878133792702  
 Log Loss: 0.5432239553263184  
 ROC AUC Score (OvR): 0.8986106308429666

## Classification Report:

	precision	recall	f1-score	support
functional	0.82	0.87	0.85	3226
functional needs repair	0.52	0.46	0.49	432
non functional	0.83	0.79	0.81	2282
accuracy			0.81	5940
macro avg	0.73	0.71	0.72	5940
weighted avg	0.81	0.81	0.81	5940



# Conclusion

Feature processing with pipelines

Random Forest Classifier

Model optimization

Cross Validation accuracy, ROC AUC, Confusion matrix to evaluate