Classification Project

Water Quality

Dataset from Kaggle

By
Shani Bronshtein
Renato Dunaevits

June 2023

Fun (Sad) facts:

- 70% of the earth’s surface is water but only 3% is considered freshwater
- Most of freshwater is inaccessible
- That leaves us with 0.4% drinkable
- Around 770 millions people don’t have access to safe drinking water
- 2 billions don’t have access at all or have access to contaminated water

Feature Description

1. **ph**: pH of 1. water (0 to 14).
2. **Hardness**: Capacity of water to precipitate soap in mg/L.
3. **Solids**: Total dissolved solids in ppm.
4. **Chloramines**: Amount of Chloramines in ppm.
5. **Sulfate**: Amount of Sulfates dissolved in mg/L.
6. **Conductivity**: Electrical conductivity of water in μS/cm.
7. **Organic_carbon**: Amount of organic carbon in ppm.
8. **Trihalomethanes**: Amount of Trihalomethanes in μg/L.
9. **Turbidity**: Measure of light emitting property of water in NTU.
10. **Potability**: Indicates if water is safe for human consumption.

Potable - 1 and Not potable - 0 - **The target**

Remember:
In this dataset, false positive is better than false negative.

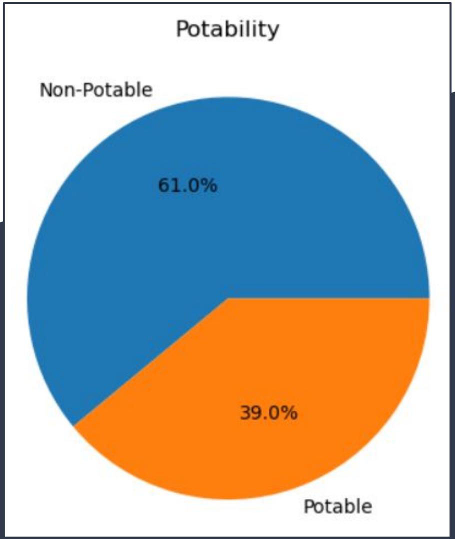
Feature Engineering

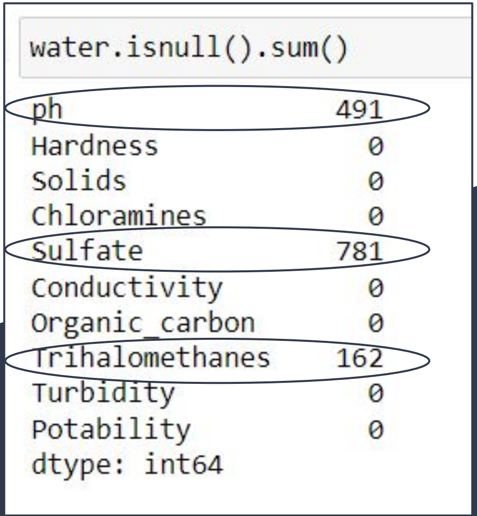
- Rows: 3276
  Column: 10

- The data set contain a numerical features only.

- About 30% from the rows contain null values.

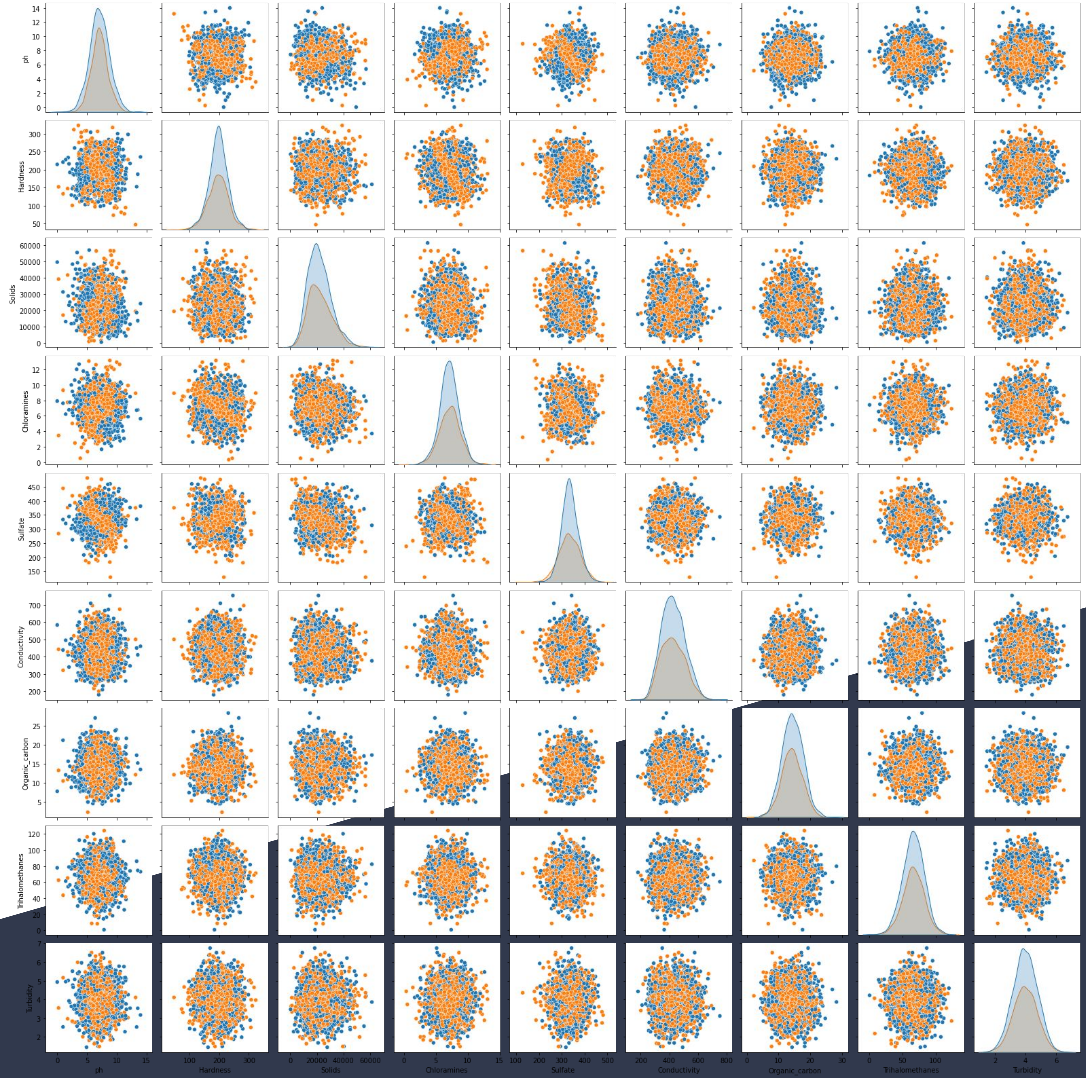
- Ratio features - target —> 60% - 40% (respectively)





```
water.isnull().sum()
```

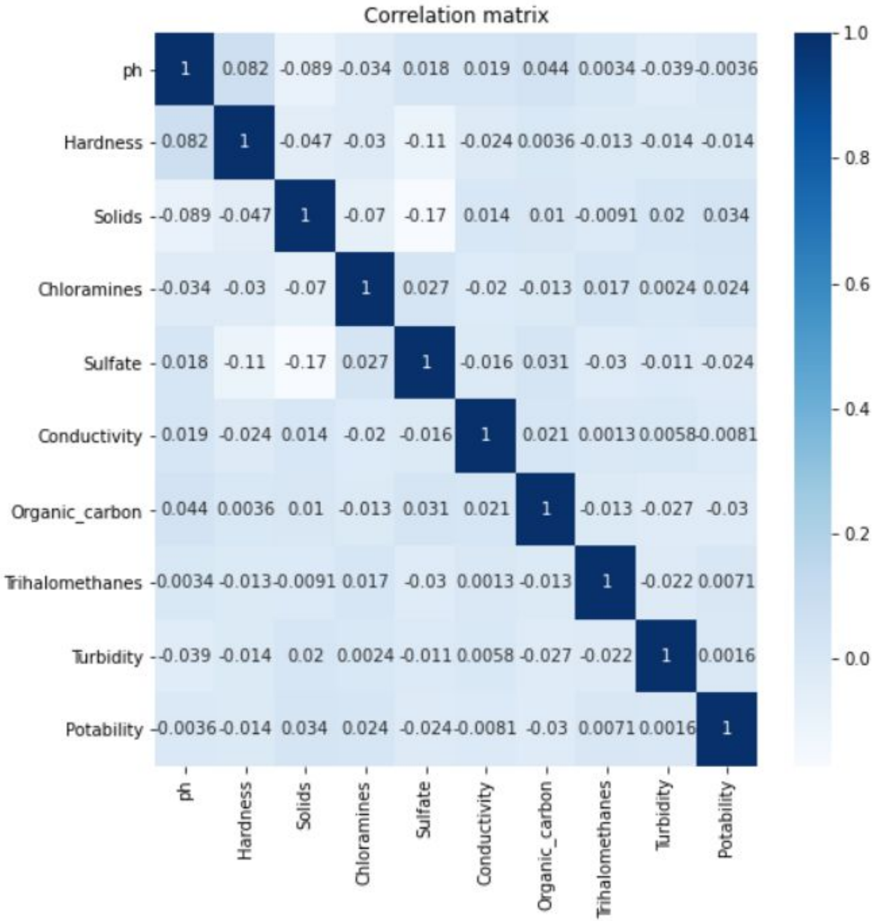
```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```



Feature Engineering

pH and sulfate show high correlation between them. In other hand, the correlation is just with potable water, so maybe it's better to stay with them, don't lose data and impute values.

Feature Engineering



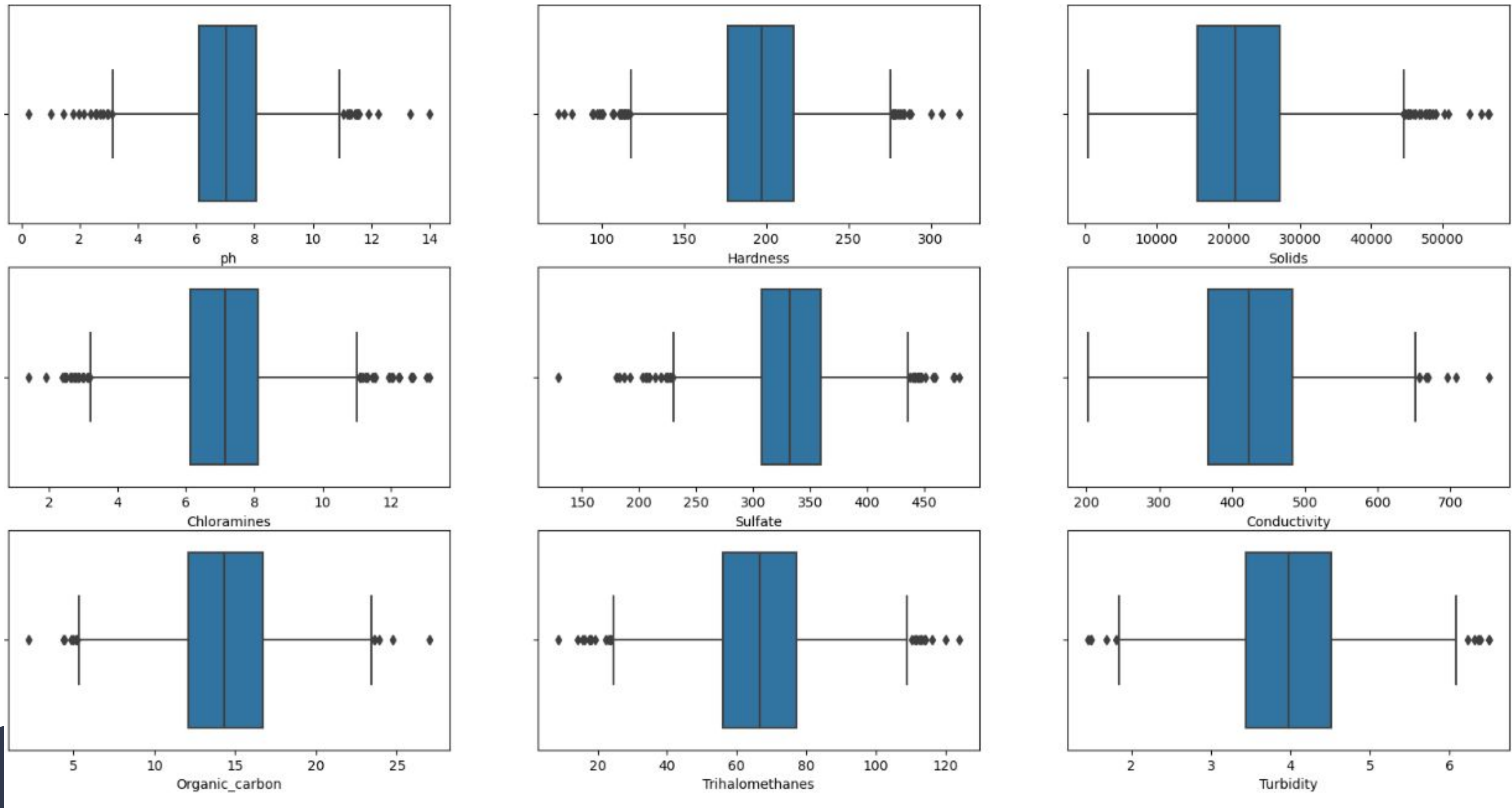
Correlation matrix

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
ph	1	0.082	-0.089	-0.034	0.018	0.019	0.044	0.0034	-0.039	-0.0036
Hardness	0.082	1	-0.047	-0.03	-0.11	-0.024	0.0036	-0.013	-0.014	-0.014
Solids	-0.089	-0.047	1	-0.07	-0.17	0.014	0.01	-0.0091	0.02	0.034
Chloramines	-0.034	-0.03	-0.07	1	0.027	-0.02	-0.013	0.017	0.0024	0.024
Sulfate	0.018	-0.11	-0.17	0.027	1	-0.016	0.031	-0.03	-0.011	-0.024
Conductivity	0.019	-0.024	0.014	-0.02	-0.016	1	0.021	0.0013	0.0058	-0.0081
Organic_carbon	0.044	0.0036	0.01	-0.013	0.031	0.021	1	-0.013	-0.027	-0.03
Trihalomethanes	0.0034	-0.013	-0.0091	0.017	-0.03	0.0013	-0.013	1	-0.022	0.0071
Turbidity	-0.039	-0.014	0.02	0.0024	-0.011	0.0058	-0.027	-0.022	1	0.0016
Potability	-0.0036	-0.014	0.034	0.024	-0.024	-0.0081	-0.03	0.0071	0.0016	1

It Seems like it doesn't have any big correlation, just small ones.

In addition, we don't have any high correlation with the target column. The highest correlation with the target is Solids in a positive way and Organic__carbon in a negative way.

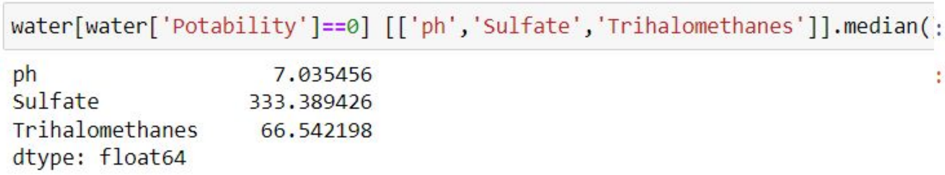
Feature Engineering



It has some outliers in parts of the features, but we decided that is better to leave them in the data.

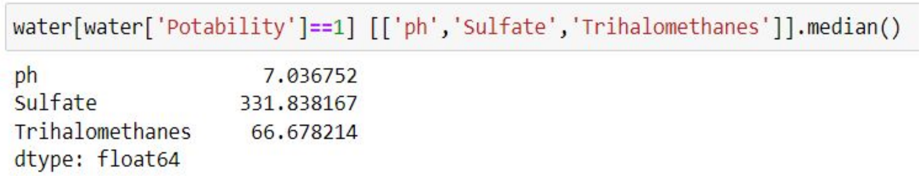
Dealing with missing values

The difference between average and median is minimal.
In addition, the correlation between thus features with the target are low.
we assume that impute null values with median will not has a high influence on the model.



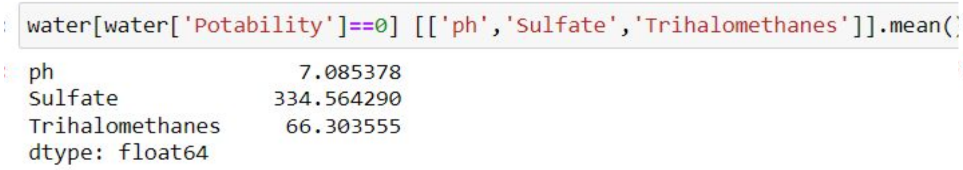
```
water[water['Potability']==0] [['ph','Sulfate','Trihalomethanes']].median(

ph                   7.035456
Sulfate            333.389426
Trihalomethanes     66.542198
dtype: float64
```



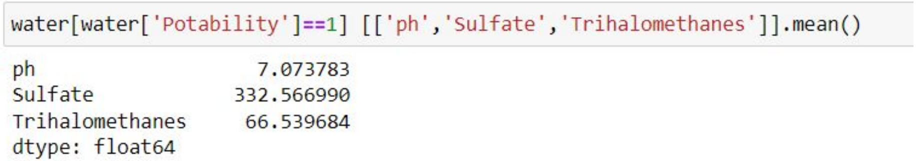
```
water[water['Potability']==1] [['ph','Sulfate','Trihalomethanes']].median()

ph                   7.036752
Sulfate            331.838167
Trihalomethanes     66.678214
dtype: float64
```



```
water[water['Potability']==0] [['ph','Sulfate','Trihalomethanes']].mean()

ph                   7.085378
Sulfate            334.564290
Trihalomethanes     66.303555
dtype: float64
```

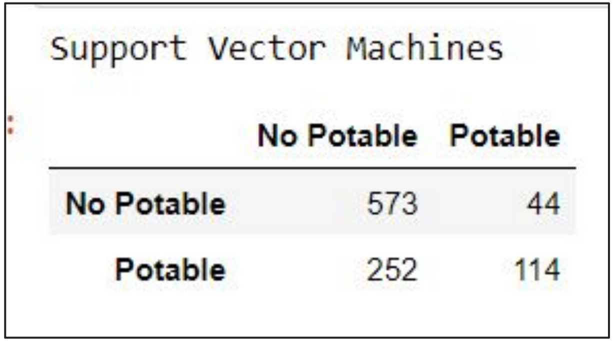


```
water[water['Potability']==1] [['ph','Sulfate','Trihalomethanes']].mean()

ph                   7.073783
Sulfate            332.566990
Trihalomethanes     66.539684
dtype: float64
```

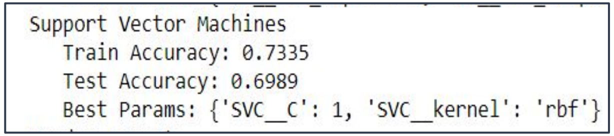
Modeling

SVM model - confusion matrix & Accuracy score



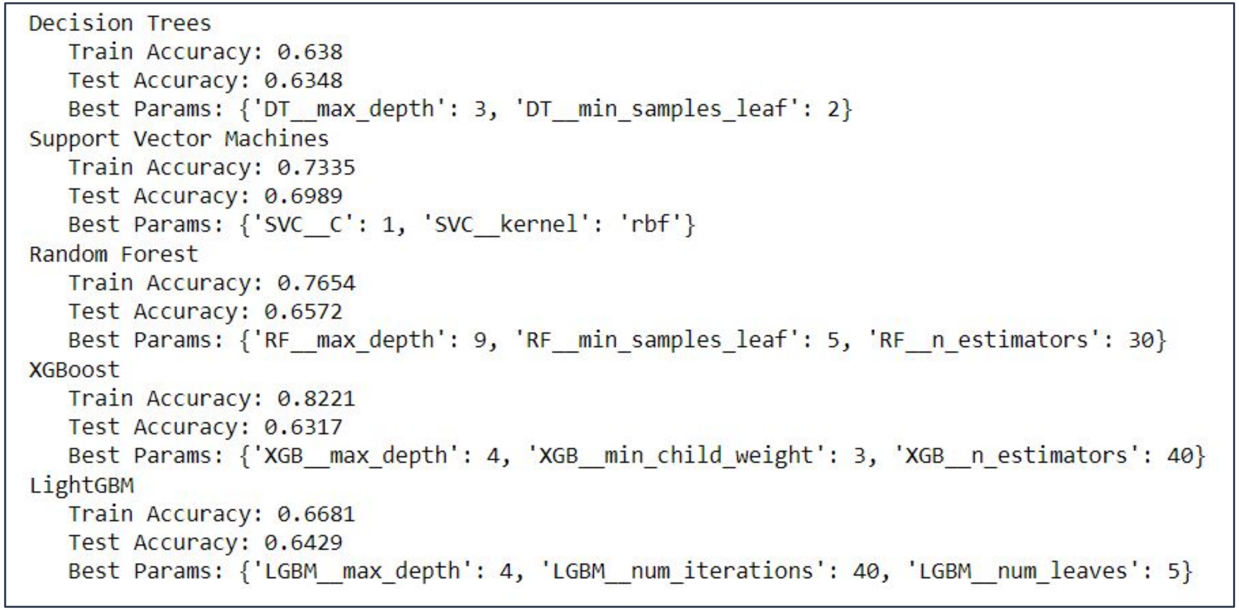
Support Vector Machines

	No Potable	Potable
No Potable	573	44
Potable	252	114



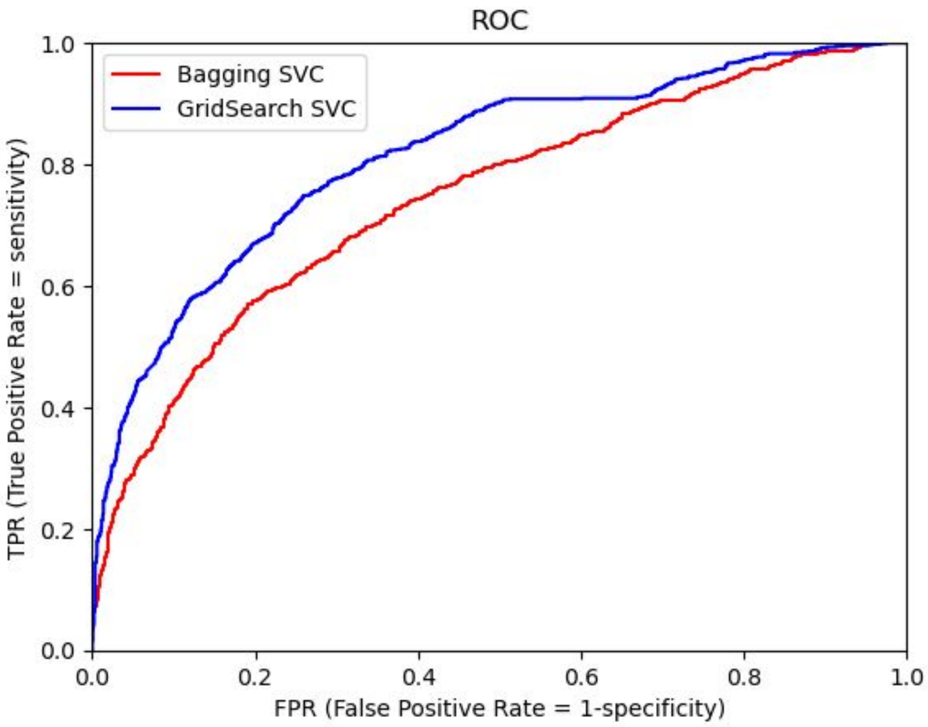
```
Support Vector Machines
   Train Accuracy: 0.7335
   Test Accuracy: 0.6989
   Best Params: {'SVC__C': 1, 'SVC__kernel': 'rbf'}
```

Additional models with grid search



```
Decision Trees
   Train Accuracy: 0.638
   Test Accuracy: 0.6348
   Best Params: {'DT__max_depth': 3, 'DT__min_samples_leaf': 2}
Support Vector Machines
   Train Accuracy: 0.7335
   Test Accuracy: 0.6989
   Best Params: {'SVC__C': 1, 'SVC__kernel': 'rbf'}
Random Forest
   Train Accuracy: 0.7654
   Test Accuracy: 0.6572
   Best Params: {'RF__max_depth': 9, 'RF__min_samples_leaf': 5, 'RF__n_estimators': 30}
XGBoost
   Train Accuracy: 0.8221
   Test Accuracy: 0.6317
   Best Params: {'XGB__max_depth': 4, 'XGB__min_child_weight': 3, 'XGB__n_estimators': 40}
LightGBM
   Train Accuracy: 0.6681
   Test Accuracy: 0.6429
   Best Params: {'LGBM__max_depth': 4, 'LGBM__num_iterations': 40, 'LGBM__num_leaves': 5}
```

ROC Curve


ROC
Bagging SVC
GridSearch SVC
TPR (True Positive Rate = sensitivity)
FPR (False Positive Rate = 1-specificity)
0.0
0.2
0.4
0.6
0.8
1.0

Conclusion

We tried to run several models, and to improve them by using grid search.

Seemingly, the XGB model has a good train accuracy score, but it isn’t validate.
The prediction in the other models show a large amount of errors.

The SVM model was the best one - It has a good train accuracy and also valid.

Thank you