Date: December 4th, 2019
1030: Hands on Data-Science
Instructor: Andras Zsom
Advisor: Joachim Krueger
Student: David Kebudi
GitHub Repository: https://github.com/Kebudi/Data1030-Final-Project

Predicting Binary Outcomes using Wisdom of Crowds

Brown University Data Science Department / Brown University Psychology Department

Data Source: Prelec, MIT

RECAP

- **Data Source:** Drazen Prelec, MIT (Princeton and MIT)
- **Idea:** Joachim Krueger, Brown University
- Classification for Wisdom of Crowds
- Classifying binary statements as TRUE or FALSE

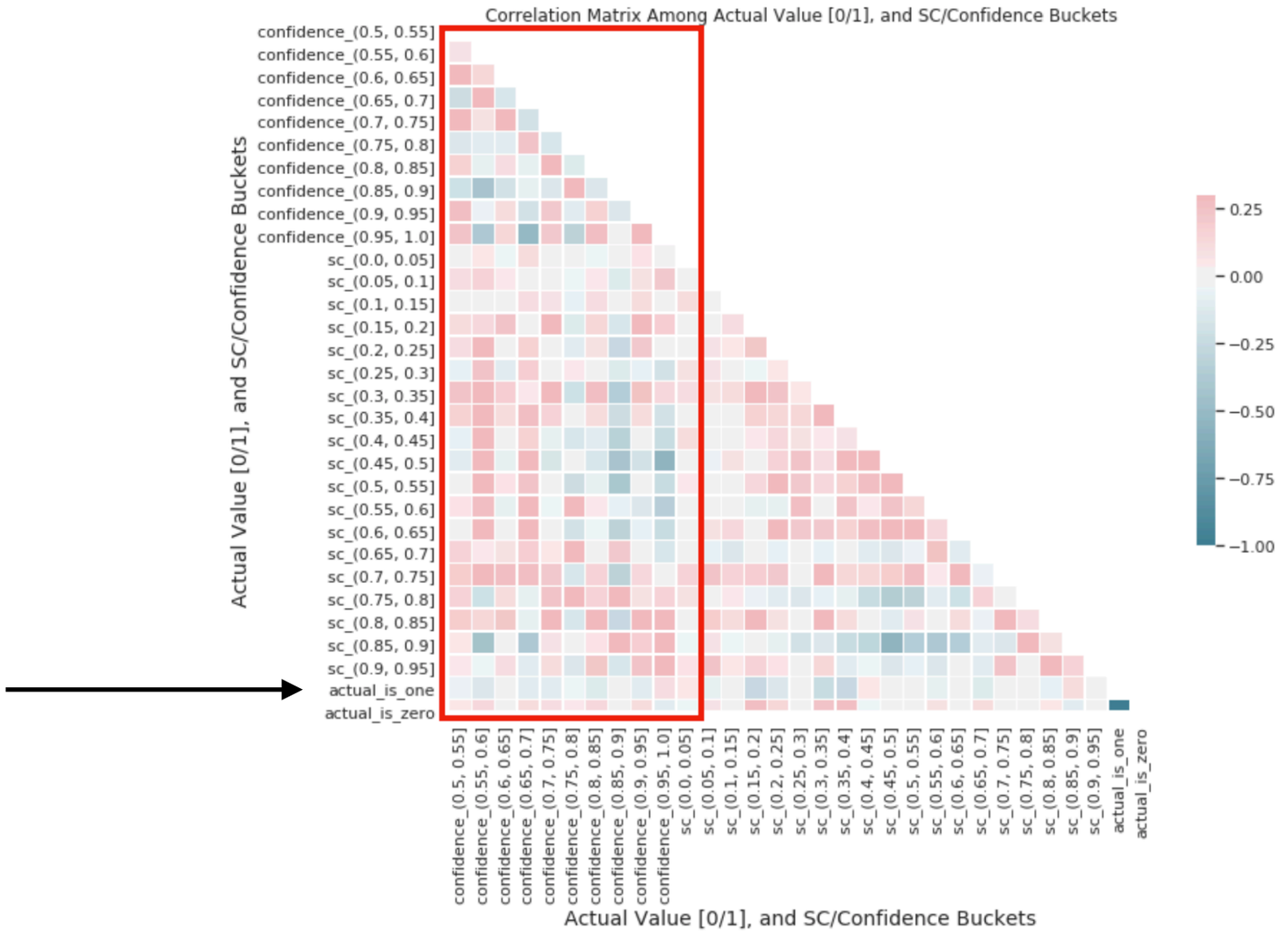
“Albany is the capital of New York State”

- **Preprocessed data size:** 210x141

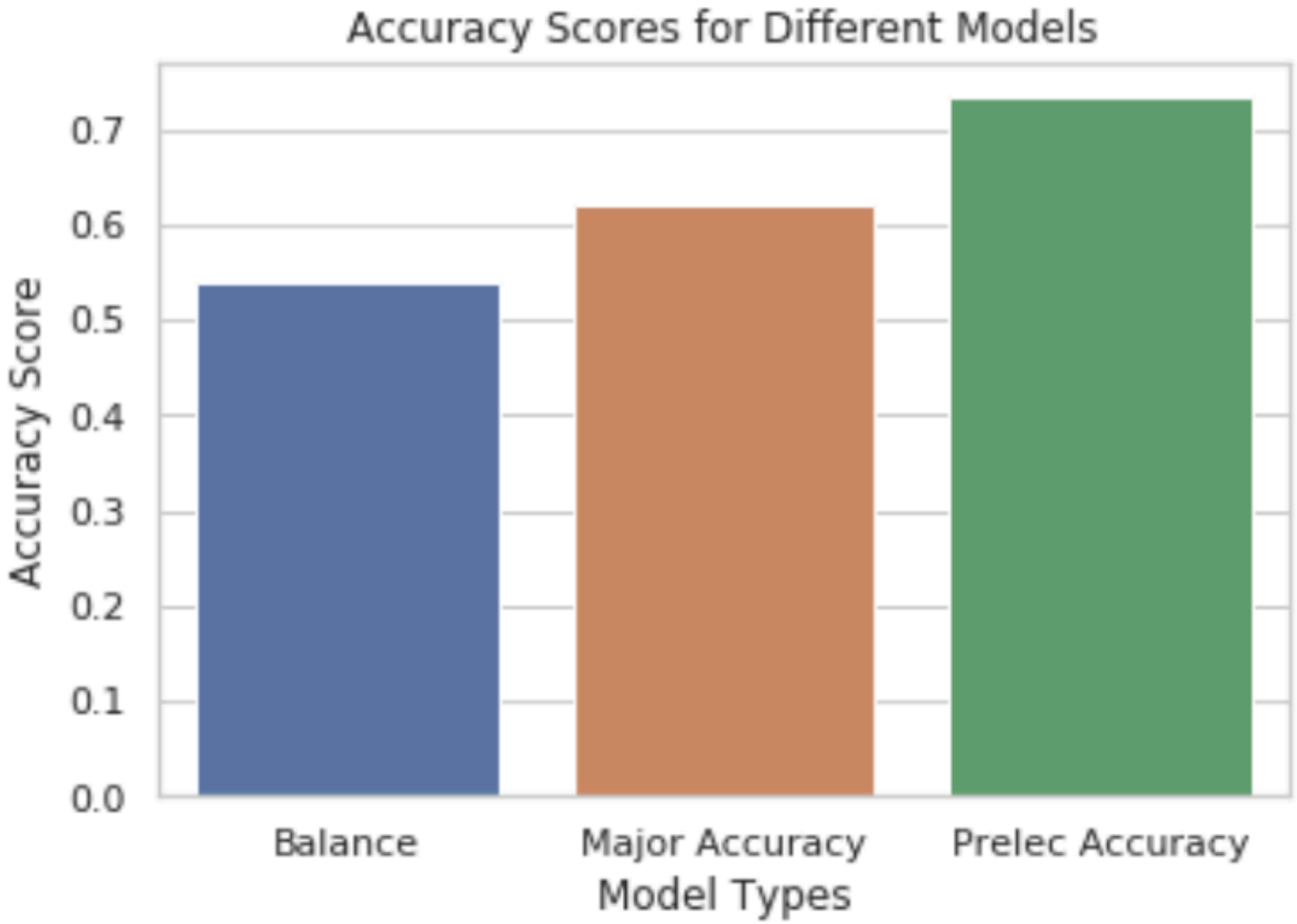
Own answer, Confidence in your answer, Meta of what % will say TRUE, Self Consensus % Meta

RECAP [EDA]

Negative corr. with ACTUAL = 0
Positive Corr. with ACTUAL = 1



RECAP [EDA]


Accuracy Scores for Different Models
Accuracy Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Balance
Major Accuracy
Prelec Accuracy
Model Types

CV Pipeline

- **KFolds** = 5, 60% **Train**, 20% **CV**, 20% **Test**

```
kf = KFold(n_splits=n_folds,shuffle=True,random_state=random_state)
```

- **Balanced Data:** %53 [0], %47 [1]

- **iid data**

```
def ML_randomforest(X_other, X_test, y_other, y_test, kf, random_state):
    import random
    param_grid = {'max_features': ['auto', 'sqrt', 'log2'],
                  'max_depth': random.sample(range(2, 100), 10),
                  'min_samples_split': random.sample(range(2, 100), 10)}
    reg = RandomForestClassifier(random_state=random_state, n_jobs=-1, n_estimators=100)
    grid = GridSearchCV(reg,param_grid=param_grid,
                        scoring=make_scorer(accuracy_score),cv=kf,
                        return_train_score=True,iid=True)
    grid.fit(X_other,y_other)
    return grid, grid.score(X_test, y_test)
```

Model

- **Nearest Neigbour**
    1. **n_neighbors:** 20 integers, randomly selected between 2 and 100
    2. **weights:** distance and uniform, giving each neighbor a different weight based on it's distance or not (equally distributed)
    3. **metric:** euclidean, manhattan, *the metrics used to measure distance*

- **Logistic Regression**
    4. **C:** 100 floats between 10^{-5} and 10^{4}
    5. **penalty:** "lasso" or "ridge"

- **Random Forrest**
    6. **max_features:** 'auto', 'sqrt', 'log2'
    7. **max_depth:** 10 integers, randomly selected between 2 and 100, *limiting over fitting by not giving a very large number*
    8. **min_samples_split:** 10 integers, randomly selected between 2 and 100

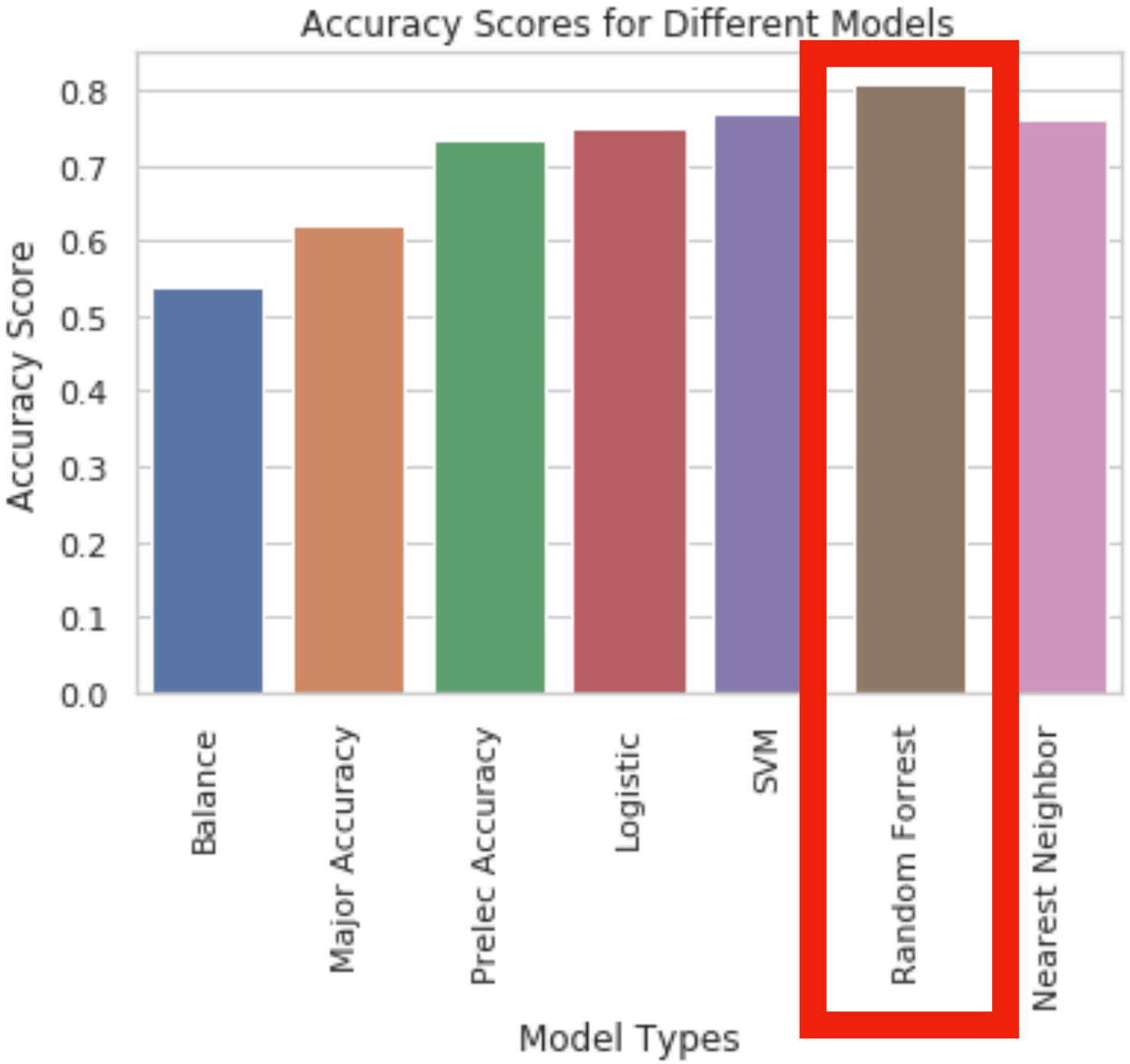
- **SVM**
    9. **C:** 30 floats between 10^{-4} and 10^{4}
    10. **gamma:** 30 floats between 10^{-4} and 10^{4}

Results [SCORES]

	Best Accuracy (Random Seed)	Average Accuracy	Standard Deviation of Accuracy
Logistic	0.75 (1090)	0.7156	0.0226
SVM	0.7738 (436)	0.7162	0.0338

	Best Accuracy (Random Seed)	Average Accuracy	Standard Deviation of Accuracy
Random Forrest	0.8154 (872)	0.7625	0.0381
Nearest N.	0.7619 (654)	0.7109	0.0294

Results [SCORES]



The parameters for the best Random Forrest Model are:

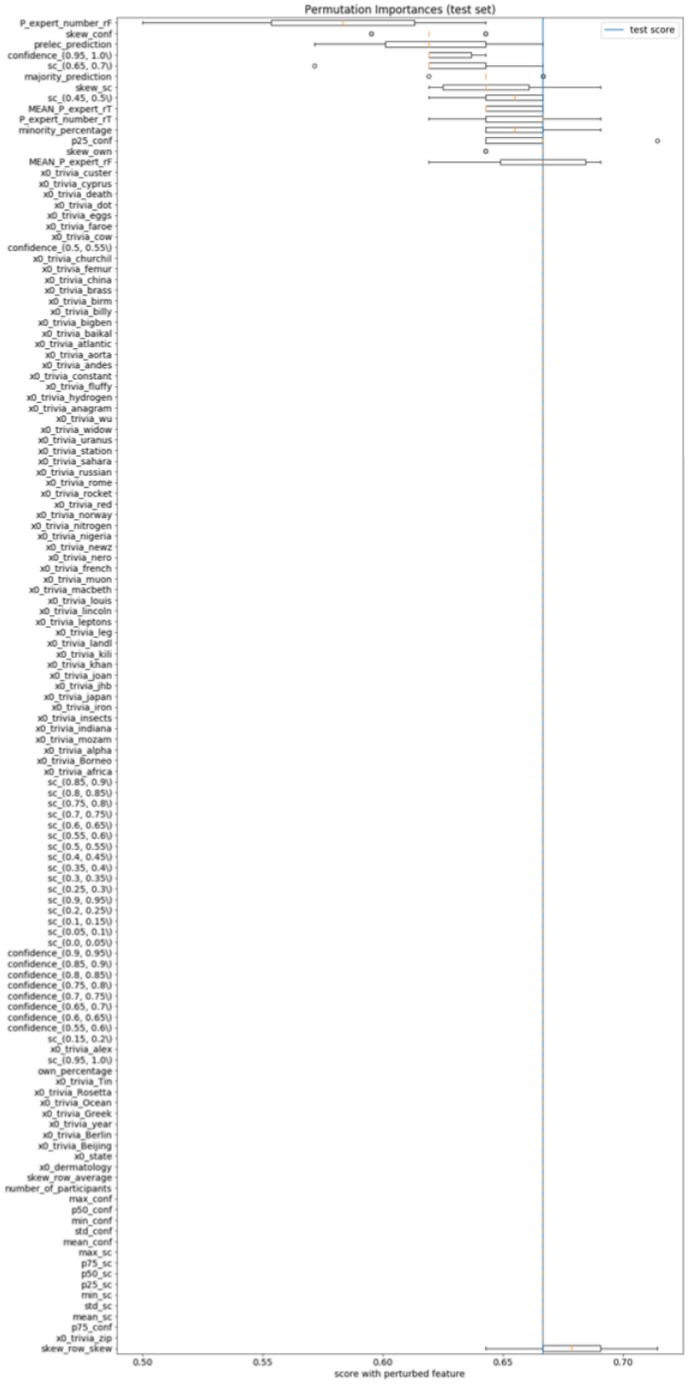
- **max_depth=11**
- **max_features='auto'**
- **min_samples_leaf=1**
- **min_samples_split=7**
- **n_estimators=100**
- **random_state=872**

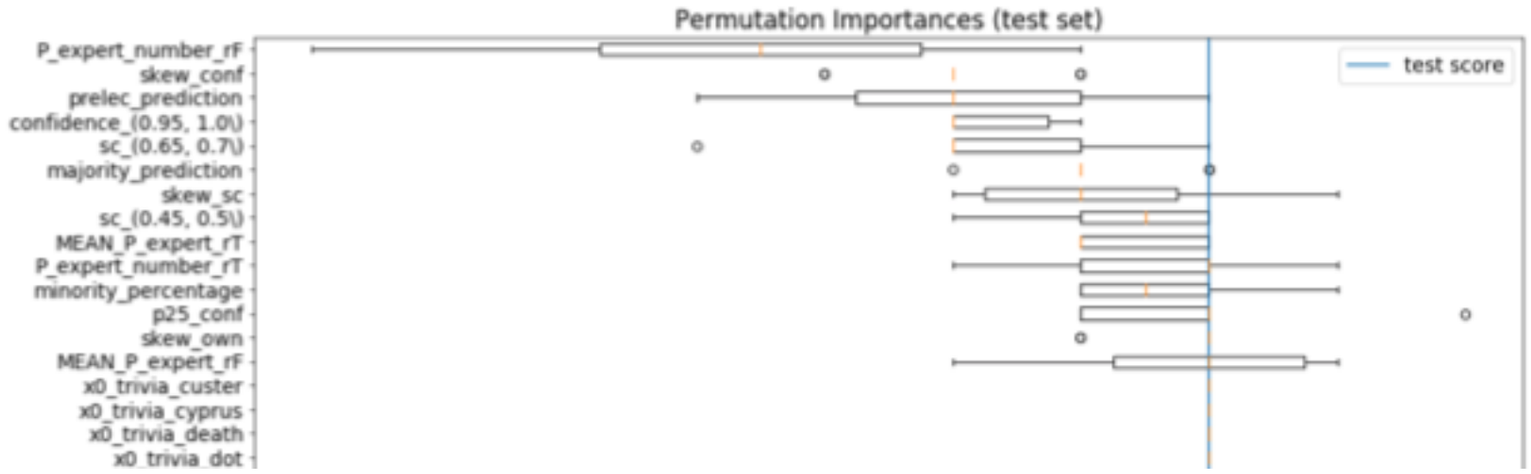
Confusion Matrix

True Values		True	False
	True	0.54	0.02
	False	0.21	0.23

Predicted Values

Results [Features Imp.]





1. **p_expert_number_rF**
2. **skew_conf**
3. **prelec_prediction**
4. **conf(0.95,1)**
5. **sc(0.65,7)**
6. **majority_prediction**

Outlook

1. **XGBoost Classifier**
2. **Data about Unknown Future Guesses** *[Markets will go up next week]*
3. **Expert Classification Model**

$$in\ minority \cap correct$$