Kübra Ergün
INTL 550

Homework 3 Report

Feature selection and one-hot encoding

First, exploring the dataset, I checked the correlation map and the strongly correlated variables. The highest correlation with the target variable 'voted' was found to be 'age' with a correlation coefficient of 0.26. So, even the highest correlation could not be classified as a strong correlation. Then, I continued with ANOVA feature selection for categorical outputs (SelectKBest) and listed the best 10 features: 'D2011', 'D2012', 'D2013', 'D2014', 'D2018', 'D2019', 'D2021', 'D2025', 'D2028', 'age'. I combined the majority of these with a few other additional theory-driven variables. The dataset was then narrowed down to the following variables: 'D2002', 'D2003', 'D2005', 'D2010', 'D2012', 'D2013', 'D2014','D2018', 'D2020', 'D2021','D2025', 'D2028', 'D2031', 'age', 'voted'.

After checking the codebook of CSES Wave Four for each selected variable, I changed the missing values (coded as 9, 99 or 999), "refused"(7, 97, 997) and "don't know" (8, 98, 998) answers to NaN values to be imputed later. I also encoded the categorical variables that are not ordered.

Imputation of NaN: Building a preliminary model with logistic regression, I experimented with different imputation strategies, such as median, mode, KNN, and MICE. I decided to submit the version in which I yield the most accurate results: MICE. However, since MICE renders float numbers, I used np.round feature to round them to the closest categorical value. Then, I scaled the X data.

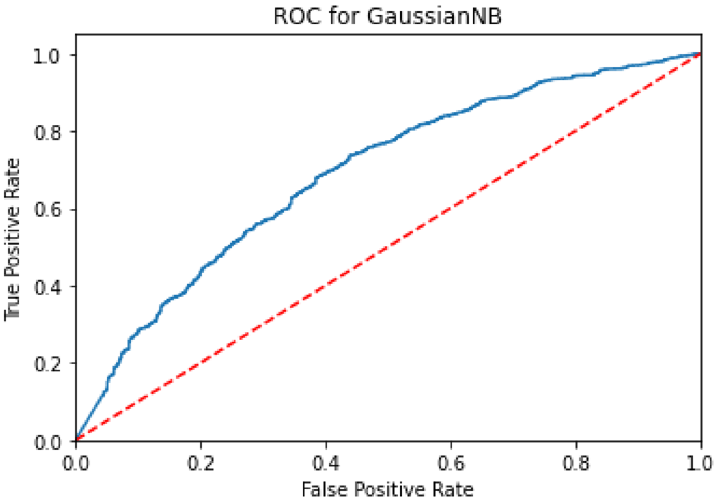
GaussianNB and LogisticRegression Models

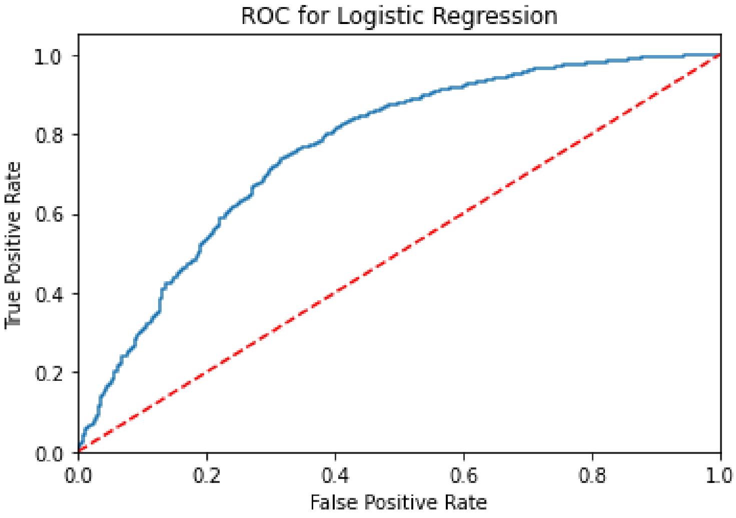
After splitting the training and test data (with test size being 0.3), I built two models with GaussianNB and LogisticRegression. Checking their accuracy assessment reports and accuracy scores, I have seen that logistic regression model has better accuracy (0.84) and better mean value (0.79).

```
GaussianNB - Mean Accuracy Score: 0.61
Logit - Mean Accuracy Score: 0.79

Confusion Matrix for GaussianNB:
 [[ 325  372]
 [ 643 2396]]

Confusion Matrix for Logit:
 [[ 148  549]
 [  59 2980]]

Accuracy Assessment for GaussianNB:
              precision    recall  f1-score   support

       False       0.34      0.47      0.39       697
        True       0.87      0.79      0.83      3039

    accuracy                           0.73      3736
   macro avg       0.60      0.63      0.61      3736
weighted avg       0.77      0.73      0.74      3736
```

```
Accuracy Assessment for Logit:
              precision    recall  f1-score   support

       False       0.71      0.21      0.33       697
        True       0.84      0.98      0.91      3039

    accuracy                           0.84      3736
   macro avg       0.78      0.60      0.62      3736
weighted avg       0.82      0.84      0.80      3736
```

ROC curves:



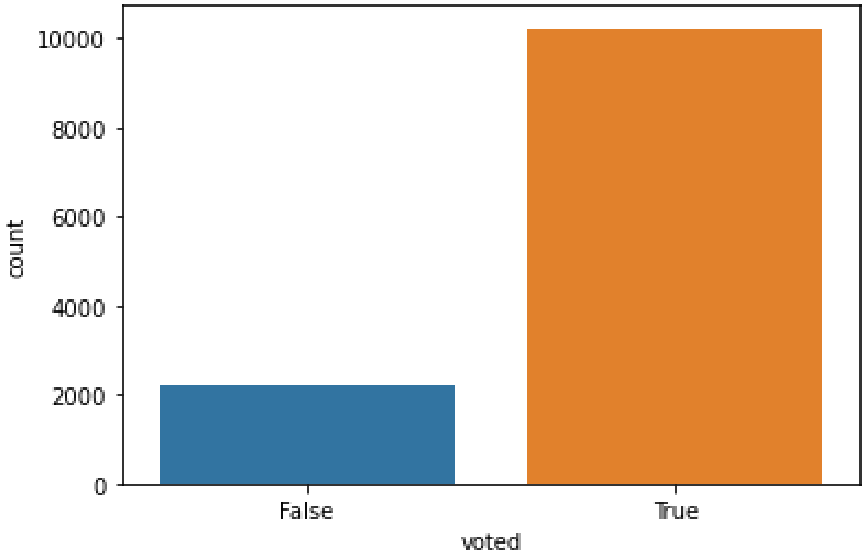


Yet, calculating the area under the ROC curve, I detected that GaussianNB is slightly a better classifier.

```
ROC AUC for GaussianNB: 0.6273506585597184
ROC AUC for LogisticRegression: 0.5964621564803418
```

I wondered if it is possible to obtain lower AUC score with a higher accuracy score, I've learned that these are not actually comparable units because AUC looks at all possible decision thresholds whereas in the accuracy assessment (based on model.predict) the threshold is set to 0.5 by default. In addition, if the feature of interest is imbalanced, accuracy may not be a meaningful criterion. And in our case, the distribution of True and False values in the voted column is indeed imbalanced, with a lot more True values.

```
sns.countplot(x='voted', data=data)
```


count
False
True
voted