CEM KIRAÇ Homework #3 - Advanced Data Analysis in Python

- The input data consisted of all categorical variables, only age was a numerical variable, therefore an encoding was needed for categorical data. Although some of the data was ordinal, I was unable to distinguish ordinals from others by coding. The only possible way seemed to read the descriptions and manually distinguish those ordinals such as income or education level. As a result I decided to one-hot all except age.

- For the missing data, I made an assumption and treated **refused and dont-know** responses as missing too. Those 2 could have a different contribution to the model but I couldnt come up with a fast solution so treated them as missing. Missing were one-hot encoded as well for each input.

- After that train-test split was done. There was 552 inputs at this stage and didnt reduce them at First, all of them were used in model fitting

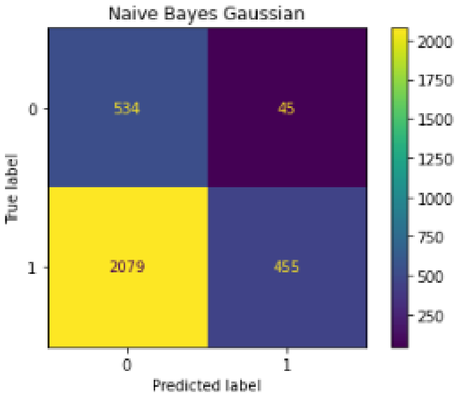
- Then after applying feature selection by logistic regression, inputs were reduced to 222, and those were used again in model fitting.

- After feature selection the generalization power of almost all models increased but their accuracy decreased, the advantage of reducing input is that your model becomes easier to interpret and implement and with reduced overfitting problem. Moreover its more robust to changes in data if some category values are not available in future.

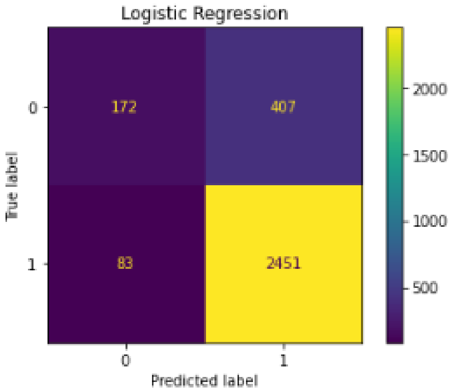
- Another very important point in this homework I want to mention is we have no business decision to make, such as lets say to predict a non-voter correctly is 10 times more valuable than predicting a voter, therefore comparing accuracy is enough for this case. You may find the results in the next two slides.

Results before feature selection

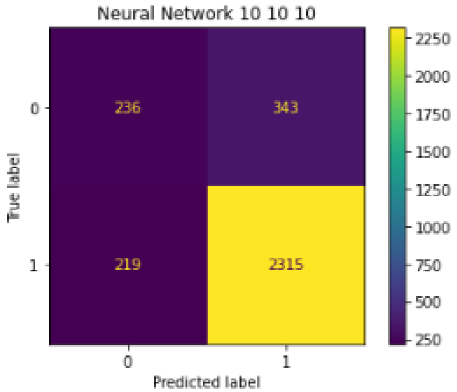
- In this 1st version all 552 inputs are included for training without feature selection
- The naive bayes performance is really poor compared to others.
- At First sight K-neigbors seems best however its misleading because there is clearly an overfitting problem , the model memorizes the train set but the performance drops drastically on the test set.
- Logistic regression is one of the best models with a %84.2 accuracy on test data
- Random forest with depth 10 is also performing nice
- Neural network with 3 hidden layers 5x5x5 has the highest accuracy with a %84.5
- Remember all inputs were included in this part, on next page number of inputs will be reduced so we may be seeing overfitting problem decreasing.
- Neural network seems to be best in this situation



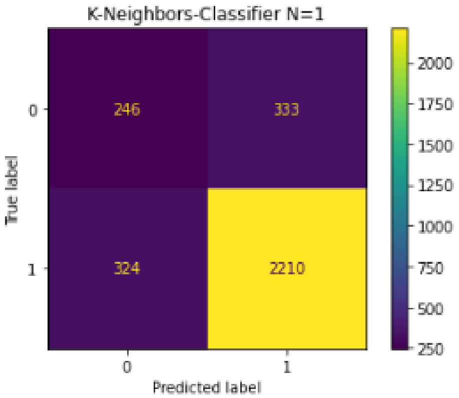
```
0.31769996787664634 TEST ACCURACY
0.3129149710858856 TRAIN ACCURACY
```



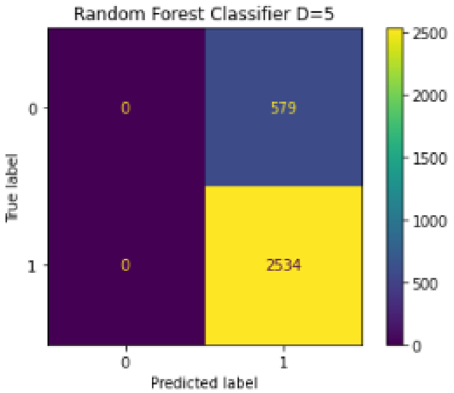
```
0.8425955669771924 TEST ACCURACY
0.8601413578924824 TRAIN ACCURACY
```



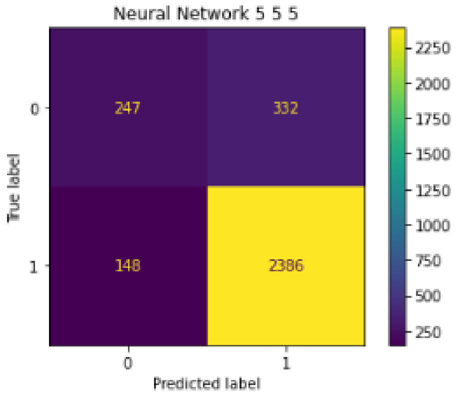
```
0.8194667523289432 TEST ACCURACY
0.9261083743842364 TRAIN ACCURACY
```



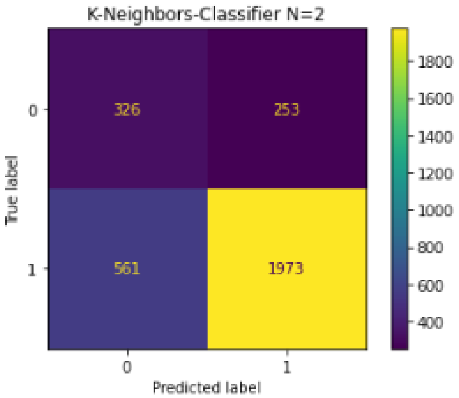
```
0.7889495663347253 TEST ACCURACY
1.0 TRAIN ACCURACY
```



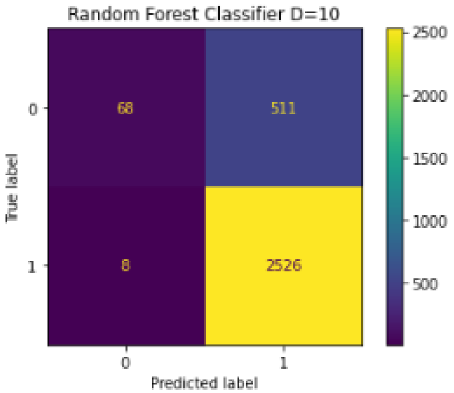
```
0.814005782203662 TEST ACCURACY
0.8237309916470337 TRAIN ACCURACY
```



```
0.8458079023450048 TEST ACCURACY
0.8954808310130649 TRAIN ACCURACY
```



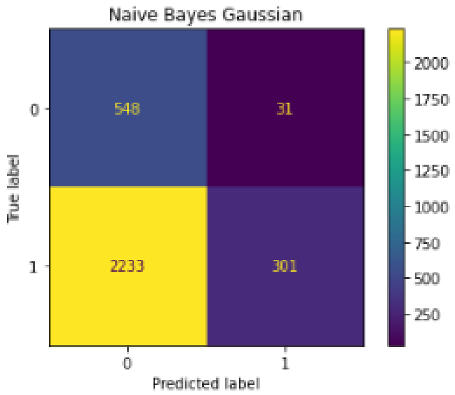
```
0.7385159010600707 TEST ACCURACY
0.9022274576997216 TRAIN ACCURACY
```



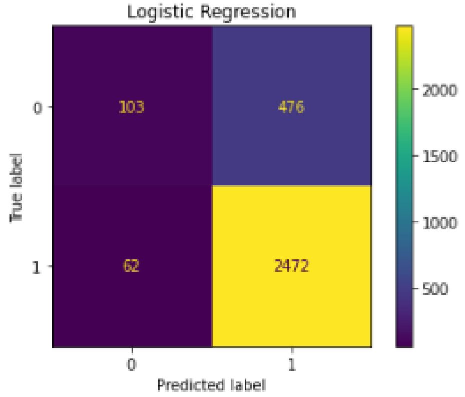
```
0.8332797944105365 TEST ACCURACY
0.8538230884557722 TRAIN ACCURACY
```

Results after feature selection

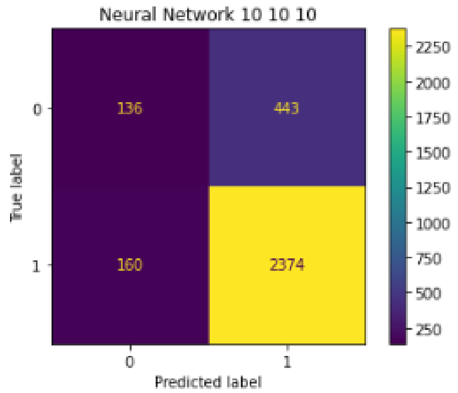
- **Feature selection applied and inputs reduced to 222**
- **The naive bayes performance is really poor compared to others.**
- **This time for K-neighbors the overfitting problem has decreased but still continues**
- **Logistic regression is still one of the best models with a %82.7 accuracy on test data**
- **Random forest with depth 10 is performing nice again**
- **Neural network with 3 hidden layers 5x5x5 has %81.6 accuracy**
- **Logistic regression is the optimal model in this situation.**



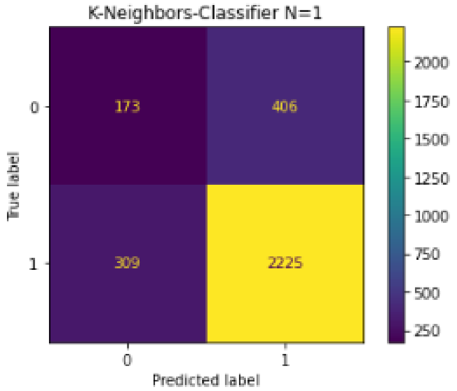
```
0.2727272727272727 TEST ACCURACY
0.270935960591133 TRAIN ACCURACY
```



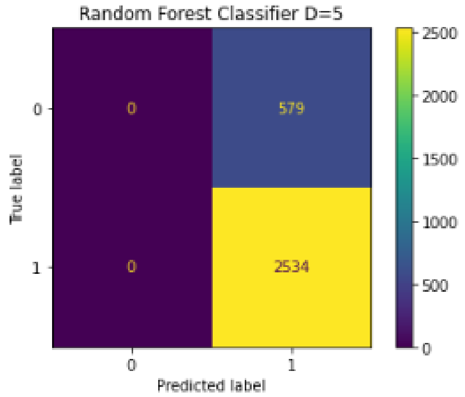
```
0.8271763572116929 TEST ACCURACY
0.8439708717070037 TRAIN ACCURACY
```



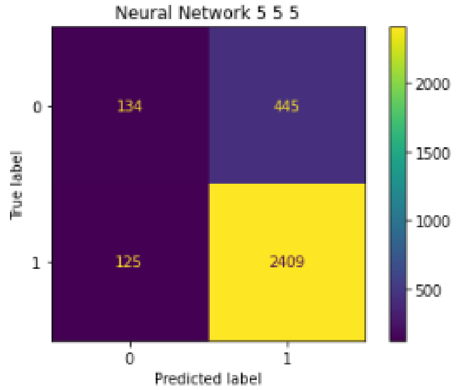
```
0.8062961773209123 TEST ACCURACY
0.8800599700149925 TRAIN ACCURACY
```



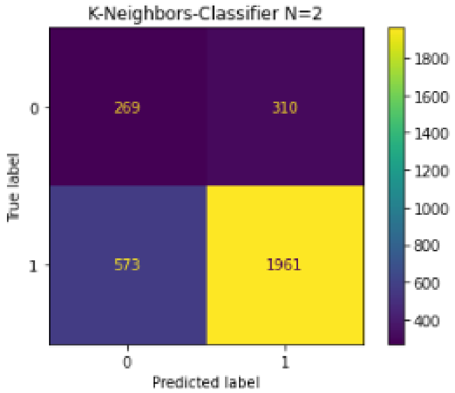
```
0.7703180212014135 TEST ACCURACY
0.886592418076676 TRAIN ACCURACY
```



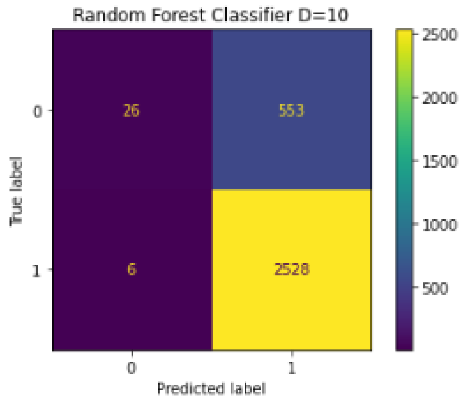
```
0.814005782203662 TEST ACCURACY
0.8237309916470337 TRAIN ACCURACY
```



```
0.8168968840346932 TEST ACCURACY
0.8556436067680445 TRAIN ACCURACY
```



```
0.7163507870221651 TEST ACCURACY
0.8240522595844935 TRAIN ACCURACY
```



```
0.8204304529392868 TEST ACCURACY
0.8345470122081816 TRAIN ACCURACY
```