# Module 17 – Asssignment 17.1 (prompt\_III.ipynb)

## Business understanding

A bank’s customers are offered term deposits, and the given data consists of 20 input variables and one output binary variable which takes the values ‘yes’ or ‘no’, according to whether the customer opted for the term deposit offer or not. Based upon the dataset, it is required to create an efficient binary classifier that can predict whether a particular customer will opt for the term deposit.

## Data understanding

The data consists of 20 input variables as given in the accompanying notebook (df.info() command). While there are no NaN values in the data, quite a few categorical input columns have ‘unknown’ as a value. The corresponding 10700 rows were removed from the full dataset of 41188 resulting in clean 30488 rows of data. This was done after estimating that the maximum % of such rows is about 30%. This approach was taken instead of the other two options, viz., imputation and treating ‘unknown’ as a category, so that the models trained on the clean, unambiguous data would be more representative of the reality.

## Data preparation

The data was checked for NaNs and none was found. As explained above the rows with at least one ‘unknown’ value in any of the columns were removed. The dataframe created was split into training and test dataframes.

## Modeling Evaluation

A baseline model was created with accuracy = 0.8734256100760955

Following five **simple** models were created.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Model** | **Train Time**  **(wall time in %time command)** | **Train Accuracy** | **Test Accuracy** |
| 1 | Simple Logistic Regression – Numeric features only, with SelectFromModel | 0.0987 s | 0.9001137059389487 | 0.8953030700603516 |
| 2 | Simple Logistic Regression – ALL features with SelectFromModel | 1.89 s | 0.9026502230385726 | 0.8981894515875098 |
| 3 | Simple K-Nearest Neighbors – All features | 0.107 s | 0.9200559783084055 | 0.888743112044083 |
| 4 | Simple Decision Tree– All features | 0.329 s | 1.0 | 0.8722120178430858 |
| 5 | Simple SVM– All features and default kernel = ‘rbf’ | 12.8 s | 0.9154202746435756 | 0.8989766465494621 |

Out of the 5 simple models above, “Simple Logistic Regression – ALL features with SelectFromModel” seems to be the optimal model as it has 107 ms training time with second highest test accuracy. The SVM model has the highest test accuracy, however it is significantly slower than the other four models.

### Question: should we keep the gender feature? Why or why not?

### Answer: No

The list of features selected using SelectFromModel in the LogisticRegression classifier (refer the attached notebook), does not include gender feature. This indicates that the weight of the gender feature is not significant and hence it can be safely discarded.

## Improving the Model

The four classifiers were then subjected to GridSearchCV with the following results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Model** | **Train Time**  **(wall time in %time command)** | **Train Accuracy** | **Test Accuracy** |
| 1 | Logistic Regression – ALL features with SelectFromModel  best\_params = {'lgr2\_\_C': 0.5, 'lgr2\_\_fit\_intercept': True} | 131 s / 10  = 13.1 s | 0.9017318289162949 | 0.898583049068486 |
| 2 | K-Nearest Neighbors – All features  best\_params = {'knn2\_\_n\_neighbors': 4} | 15 s / 5  = 3 s | 0.9190501180792443 | 0.9186565205982682 |
| 3 | Decision Tree– All features  best\_params = {'dtc2\_\_max\_depth': 5} | 7.23 s / 5  = 1.446 s | 0.9091664480013995 | 0.9060614012070323 |
| 4 | SVM– All features and default kernel = ‘rbf’  best\_params = {svm2\_\_gamma': 10.0, 'svm2\_\_kernel': 'rbf'} | 32 m 5 s / 3  = 1925 s / 3  = 641.67 s | 0.9975946820607015 | 0.9971136184728417 |

## Summary of Findings

The SVC classifier with kernel = ‘rbf’ and gamma = 10 as shown by GridSearchCV has the highest test accuracy of 0. 9971136184728417, very close to 1. However, it is slower by order of ~ 500 compared to other classifiers. Taking into consideration the optimal combination of speed and accuracy, Decision Tree classifier with max\_depth=5, seems to be the best out of the four classifiers analyzed.

## Next steps

Run SVM with GridSearchCV on a faster processor and try variations of kernel types such as 'linear', 'sigmoid' and 'poly'. Since 'poly' has default degree=3 and there are 47 input features, care must be taken to select only the top 4 or 5 most significant features so that the computation can be completed in reasonable amount of time.