**#The following is the methodology I used**

1. Loaded the data

2. Cleaned the data as per below:

1. Checked what columns had "unknown" values.

2. If number of unknows is too less, then got rid of those data points

3. Converted categorical values to numeric using get\_dummies() method

4. Deleted columns that did not make sense to run the analysis, such as how the bank reached out to the customer, via telephone or cellular, which day they reached out etc. These are not critical factors for someone to subscribe to a term deposit

3. Once the data cleanup is completed, I created the train and test dataset. I used sklearn's train\_test\_split and split the data into training (80%) and validation testing (20%). Since the data was ordered I shuffled it. I should have used stratify so that the data in train and test having comparable yes and no outcomes. However, I did not do that in my original run and didn't have the time to rerun with this parameter. But something that I would do in practice.

4. Ran the analysis on training data and test data using 4 different classifiers. The following are the test scores.

1. Logistic regression - 89.34% (tn - 90% and tp = 64%)

2. KNearestNaighbors - 89.36%

3. Decision Tree - 89.46%

4. SVC - 89.44%

5. Dummy - 88.42%

5. Since there were more than 35 features after I used get\_dummies() on many categorical features, I used PCA to reduce the features.

1. Found that 1st 10 pca components, were able to account for over 60% of the variance in data

2. 1st 24 components, were able to account for over 85% of the variance in the data

6. Ran all the analysis once again using the principal components. For ease of use, I used only the top 2 PCA components. This should have only accounted for 20% of the variance, but I was very surprised with the results as they performed pretty closely to using the whole dataset.

1. Logistic regression - 89.17% (tn - 89.6% and tp = 68.5%)

2. KNearestNaighbors - 89.16%

3. Decision Tree - 89.13%

4. SVC - 89.3%

**Here are my learnings/findings**

1. All types of classifiers yield pretty similar results. Although the output is similar, we need to consider the size of the data to determine, which approach may be suitable.

2. SVC was very slow performing and Decision trees were very fast. KNearestNeighbor also was very computationally intensive when the trying to compare with more neighbors.

3. Decision trees are very efficient in finding the optimal solution within a short depth. In my case, tree\_depth was just 3, where it found the optimal solution

4. PCA needs to be used, when there are a greater number of features and large data sets. In my case, even with 2 PCA components, the results were very close to using the whole data set. Again, the type of classifier did not make much difference in the results. All the classifiers scored approximately the same. Given the results were similar, it makes sense to use Decision trees as they were very fast. SVC was the slowest.

5. Although the models performed at 90% or so, even the dummy classifier performed at 88.5%, hence the models weren't that much smarter in classifying the data. I would get additional data and try to get a balance of 'yes' and 'no' outcome and then try out the model again to see if the dummy classifier goes down in scoring and our models predict at the same or better level.

**URL to the notebook**

https://github.com/rnachloo/RajeshAIML/blob/main/Capstone\_Bank\_Project.ipynb