**DSBA/MBAD 6211 Assignment 2: Advanced Predictive Models**

**Instructions:** This is an individual assignment. The submitted solution and answers should be your own. The data file for this homework is **BankChurn.csv**, which is to be downloaded from Canvas. You are asked to use Python to build Random Forest and SVM models and answer the given questions. Create a new Word document and save it as Predictive\_xxxx (where xxxx is your ninernet login name). Write your full name on the first page of the Word document.

Where required, write your answers or paste screenshots in this Word document. You need to submit both the Word document and Python Code file. **Your Python code should run correctly for your assignment to be graded. Code that generates error will result in loss of points (up to a maximum of 20%)**

# Variables and models naming requirements:

* Include your ***name initials*** to the data frame names as well as model names in your Python coding. This is required for your work to be graded.
* For instance, my initials are **CS**, and in my coding, I would name the data frames as ***dfCS, dfCS.train***, and ***dfCS.test.*** I would also name the models as ***RFCS, SVMCS***, etc.

**Problem description and questions:** We will analyze a dataset **BankChurn.csv** which had data of a bank’s customers and their churn. The descriptions of the columns are given below. Run both Random Forest and Support Vector Machines models on the data to predict whether a customer will churn or not (i.e., the target variable is “churn”) and answer the questions that follow on the next page.

* + customer\_id: Unique identifier for the customer.
  + credit\_score: Customer's credit score.
  + country: The country of the customer (e.g., France, Spain, Germany).
  + gender: Customer's gender.
  + age: Customer's age.
  + tenure: Number of years the customer has been with the bank.
  + balance: Customer's account balance.
  + products\_number: Number of bank products the customer is using.
  + credit\_card: Indicates whether the customer has a credit card with the bank (1 = Yes, 0 = No).
  + active\_member: Indicates whether the customer is an active member (1 = Yes, 0 = No).
  + estimated\_salary: Customer's estimated salary.
  + churn: Indicates whether the customer has left the bank (1 = Yes, 0 = No).

# Questions & Problem Tasks

1. Are there any variables which cannot be used in your model? Explain why? (2 pts)

- customer\_id would be the one we cannot use as its only use would be to be a complete identifier and if used, would create a model that is close to near perfect.

A screenshot of a computer

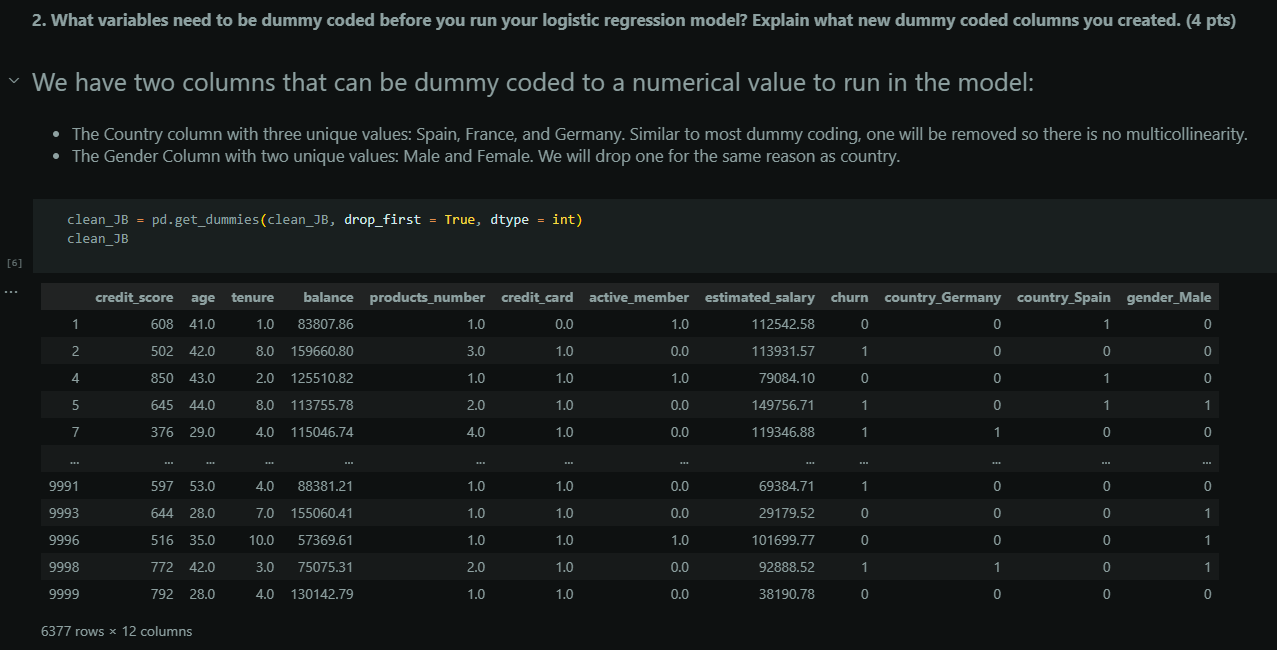
Description automatically generated

1. What variables need to be dummy coded before you run your logistic regression model? Explain what new dummy coded columns you created. (4 pts)

We have two columns that can be dummy coded to a numerical value to run in the model:

- The Country column with three unique values: Spain, France, and Germany. Similar to most dummy coding, one will be removed so there is no multicollinearity.

- The Gender Column with two unique values: Male and Female. We will drop one for the same reason as country.



1. Do you have to consider missing values in your dataset? How did you handle the presence of missing values, if any? (4 pts)

- There are multiple nulls in many different columns, primarily in the column "balance" with 3617. We should consider them and make sure to clean the data so there is no potential skewing or inaccuracies by the nulls. We will remove all the nulls from all columns to be sure.

A screenshot of a computer program

Description automatically generated

1. First run the decision tree classifier using all independent variables in the dataset, after dropping the ones identified in question 1. You can use any of the criteria of gini index or information gain. What is the accuracy of this classifier? Show the confusion matrix with proper labels. (5 pts)

After dropping ID, I split it into train and test data, with my y variable just being churn while my X being everything but ID (including the dummy variables that weren’t dropped). Then we create the Decision Tree Classifier specifying ‘gini’ as our criterion. We then train the model and predict the responses, using that we calculate the accuracy of and the confusion matrix.

A screenshot of a computer program

Description automatically generated

A chart with numbers and a yellow and purple squares

Description automatically generated with medium confidence

# Running Random Forest Classifier: (15 pts)

* 1. Run a Random Forest Classifier. Choose the appropriate set of hyperparameters. What is the accuracy of this classifier? Show the confusion matrix with proper labels. Does this classifier perform better than the decision tree in question 4 Yes it does perform better than the decision tree in question 4 by around 7%, this makes sense as it is a tree with a more parameter optimized method to find the best split accuracy out of all possible trees.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Re-run the Random Forest Classifier with a different set of parameters. What is the accuracy of this classifier? Show the confusion matrix with proper labels. Does this classifier perform better than the first two you ran?

The accuracy is pretty much the same as the initial one, with a very minute increase in the accuracy of about .0020. The random forest classifier still performs better than the decision tree.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Fine tune the Random Forest model.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

* + 1. What are the optimal parameter settings for the Random Forest Classifier?

Best parameters: {'criterion': 'gini', 'max\_depth': 10, 'max\_features': 'sqrt', 'n\_estimators': 500}

Best estimator: RandomForestClassifier(max\_depth=10, n\_estimators=500, random\_state=10)

* + 1. List the variables in the decreasing order of their importance. Do you agree with the rankings, based on your knowledge of why customers quit their banks?

X\_JB: ['credit\_score', 'age', 'tenure', 'balance', 'products\_number',

'credit\_card', 'active\_member', 'estimated\_salary', 'country\_Germany',

'country\_Spain', 'gender\_Male']

Feature importance: [0.08906299 0.31338362 0.04740451 0.11747388 0.18958945 0.01139614 0.06686553 0.09400465 0.04092393 0.00787648 0.02201882]

List of variables in the decreasing order of their importance:

age 0.313384

products\_number 0.189589

balance 0.117474

estimated\_salary 0.094005

credit\_score 0.089063

active\_member 0.066866

tenure 0.047405

country\_Germany 0.040924

gender\_Male 0.022019

credit\_card 0.011396

country\_Spain 0.007876

Based on the outputs, the importance of makes sense as the three strongest ones being age, product number, and balance. Age makes sense as a person grows, they choose to stay with a bank that gives them the most benefit, age would be a huge influence. If a person has many products at the bank that they use, they are less inclined to leave and vice versa if they don’t have many. Balance determines whether if you can keep money there, if a person’s balance is almost nonexistent, then their reason to stay at the bank decreases. Situations like these influence many people depending on their situation.

* + 1. Show the accuracy and confusion matrix for this classifier.

A screenshot of a computer

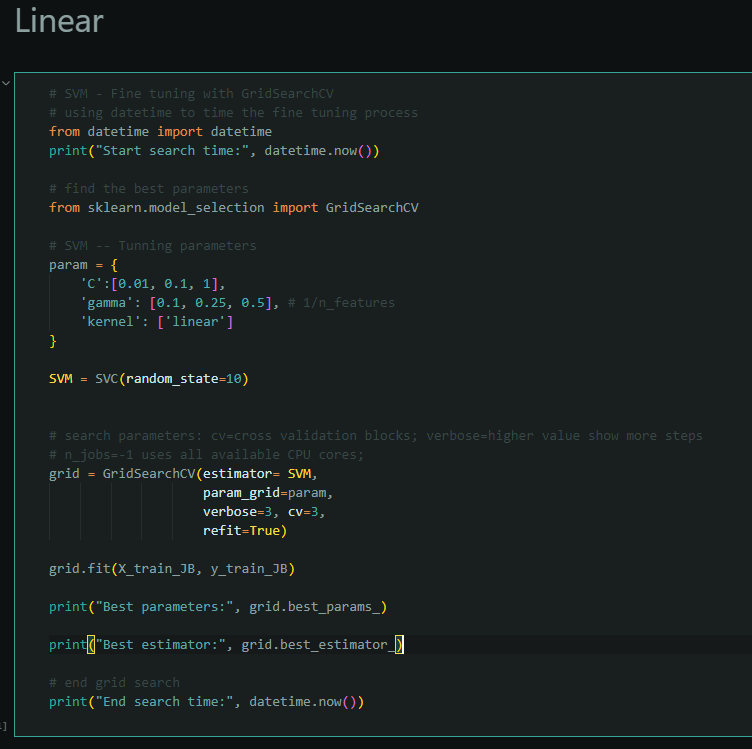
Description automatically generated

# Running SVM Classifier: (15 pts)

* 1. Run an SVM Classifier, first with a linear kernel and then with a radial kernel. Give your own values for the other hyperparameters but keep them the same across the two runs of SVM. Which kernel gives a better performance for this dataset? Explain.

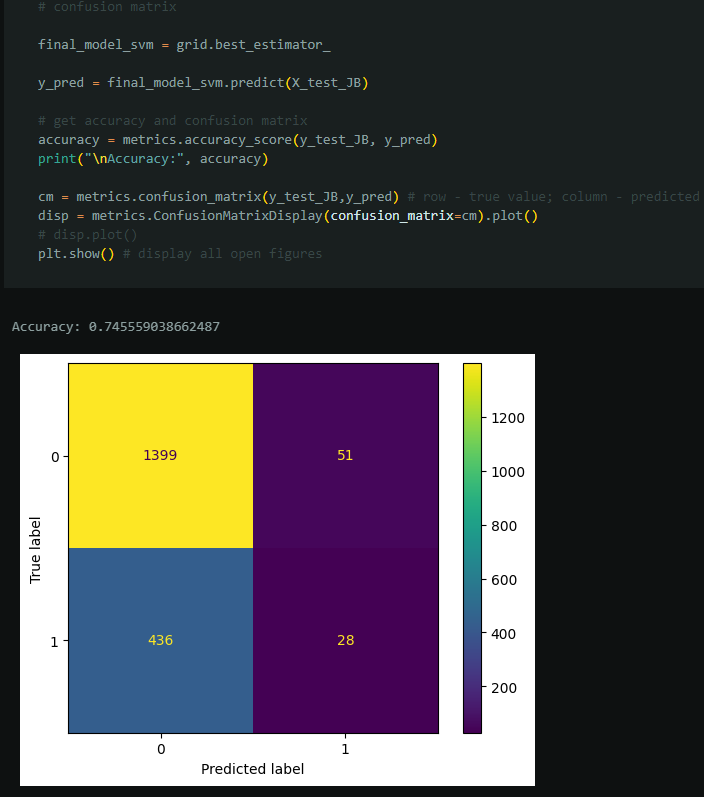
Between the two, the radial kernel has a higher accuracy compared to the linear, the confusion matrix on the radial however has 0 for False positive and True positive making it less inclined in its findings based on the fitting. However both accuracies for the kernels are very similar with rbf having a slight advantage.

Linear:



A screen shot of a computer

Description automatically generated



Radial:

A screenshot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Fine tune the SVM model.

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

* + 1. What are the optimal parameter settings for the SVM classifier?

Best parameters: {'C': 1e-05, 'gamma': 1e-05, 'kernel': 'rbf'}

Best estimator: SVC(C=1e-05, gamma=1e-05, random\_state=10)

* + 1. Show the accuracy and confusion matrix for selected classifier by the fine tuning.

A screenshot of a computer

Description automatically generated

1. Overall, which model (random forest or SVM) perform better for this dataset? Explain. (5 pts)

I would choose the random forest model for multiple reasons:

- The accuracy for random forest is much better than using either of the classifiers

- The time for the linear SVM, which has a better overall confusion matrix than rbf, is terrible even at a cross fold validation of 3, so random forest beats it in that regard.

- The selection of parameters for random forest was more representive in the long run in the actual parameter tuning

Submit your Word document and the python code through Canvas.