DATA-445
Exam I Take-Home

Name: ______________________________

Exam Instructions

1. **Show all your work** and write complete and coherent answers.
2. Show all of the steps that you used to get your final answer. If you do not show your work I can not give partial credit in the case of incorrect answers.
3. **Please strive for clarity and organization.**
4. **You are not allowed to discuss any of the exercises in Exam 1 take-home with others. Identical submissions will receive a 0 as a grade in Exam 1 take-home.**
5. **Late submission will not be accepted, regardless of the circumstances.**

Take the time to carefully read all the questions on the exam. GOOD LUCK!

1. Consider the `College.csv` data file. This file contains a number of variables for different universities and colleges in the US. The variables are:

   - `Private`: Public/private indicator.
   - `Apps`: Number of applications received.
   - `Accept`: Number of applications accepted.
   - `Enroll`: Number of new students enrolled.
   - `Top10perc`: New students from top 10% of high school class.
   - `Top25perc`: New students from top 25% of high school class.
   - `F.Undergrad`: Number of full-time undergraduates.
   - `P.Undergrad`: Number of part-time undergraduates.
   - `Outstate`: Out-of-State tuition.
   - `Room.Board`: Room and board costs.
   - `Books`: Estimated book costs.
   - `Personal`: Estimated personal spending.
   - `PhD`: Percent of faculty with Ph.D.'s.
   - `Terminal`: Percent of faculty with terminal degree.
   - `S.F.Ratio`: Student/faculty ratio.
   - `perc.alumni`: Percent of alumni who donate.
   - `Expend`: Instructional expenditure per student.
   - `Grad.Rate`: Graduation rate.

   The goal is to build a model that we can use to predict the number of applications that a university will receive.

   (a) (4 points) Using the `pandas` library, read the csv data file and create a data-frame called `college`.

   (b) (3 points) Change the `Private` variable from a categorical variable to a numerical variable. That is, change `Yes` to 1 and `No` to 0.

   (c) (5 points) Using `Private`, `F.Undergrad`, `P.Undergrad`, `Outstate`, `Room.Board`, `Books`, `Personal`, `S.F.Ratio` and `Grad.Rate` as the predictor variables, and `Apps` as the target variable, split the data into `train` (80%) and `test` (20%).

   (d) (8 points) Using the `train` dataset, `MinMaxScaler`, and `make_pipeline`, build a linear regression model. After that, use this model to predict on the `test` dataset. Report the MSE of this model.

   (e) (12 points) Using the `train` dataset, `MinMaxScaler`, and `Pipeline`, build a ridge regression model as follows:

      - First estimate the optimal lambda via cross-validation using 5-folds. Use the following Python command to generate the set of lambdas to be considered: `np.linspace(0.001, 100, num = 100)`. Notice that `np` is the alias for the `numpy` library.

- Using the optimal lambda, build the ridge regression model.

Finally, use this model to predict on the `test` dataset. Report the MSE of this model.

(f) (12 points) Using the `train` dataset, `MinMaxScaler`, and `Pipeline`, build a LASSO regression model as follows:

- First estimate the optimal lambda via cross-validation using 5-folds. Use the following Python command to generate the set of lambdas to be considered: `np.linspace(0.001, 100, num = 100)`. Notice that `np` is the alias for the `numpy` library.
- Using the optimal lambda, build the LASSO regression model.

Finally, use this model to predict on the `test` dataset. Report the MSE of this model.

(g) (3 points) Using the results from parts (d), (e) and (f), what model would use to predict the number of applications that a university receive?

2. Consider the `churn-bigml-80.csv` and `churn-bigml-20.csv` datafile for this question. The Orange Telecom's churn dataset, which consists of cleaned customer activity data (features), along with a churn label specifying whether a customer canceled the subscription, will be used to develop predictive models. Each row represents a customer; each column contains customer's attributes. The datasets have the following attributes or features:

- `State`: state where the customer live.
- `Account_length`: number of months the account is active.
- `Area_code`
- `International_plan`: whether or not the customer has an international plan.
- `Voice_mail_plan`: whether or not the customer has a voice mail plan.
- `Number_vmail_messages`: number of voice mails.
- `Total_day_minutes`
- `Total_day_calls`
- `Total_day_charge`
- `Total_eve_minutes`
- `Total_eve_calls`
- `Total_eve_charge`
- `Total_night_minutes`
- `Total_night_calls`
- `Total_night_charge`
- `Total_intl_minutes`
- `Total_intl_calls`
- `Total_intl_charge`
- `Customer_service_calls`
- `Churn:` whether or not the customer churn.

The goal is to build models that can help Orange Telecom to flag customers who likely to churn.

(a) (5 points) Using the `pandas` library, read the csv data file and create two data-frames called: `telecom_train` (for `churn-bigml-80.csv`) and `telecom_test` (for `churn-bigml-20.csv`).

(b) (12 points) Conduct the following feature engineering:

- Change the `Churn` variable from a categorical variable to a numerical variable. That is, change `True` to 1 and `False` to 0 in both data-frames: `telecom_train` and `telecom_test`.
- Change the `International_plan` variable from a categorical variable to a numerical variable. That is, change `Yes` to 1 and `False` to 0 in both data-frames: `telecom_train` and `telecom_test`.
- Change the `Voice_mail_plan` variable from a categorical variable to a numerical variable. That is, change `Yes` to 1 and `False` to 0 in both data-frames: `telecom_train` and `telecom_test`.
- Create a new variable called: `total_charge` as the sum of `Total_day_charge`, `Total_eve_charge`, `Total_night_charge`, and `Total_intl_charge` in both data-frames: `telecom_train` and `telecom_test`.

(c) (5 points) In both data-frames `telecom_train` and `telecom_test`, only keep the following variables: `Account_length`, `International_plan`, `Voice_mail_plan`, `total_charge`, `Customer_service_calls`, and `Churn`.

(d) (20 points) Consider the `telecom_train` dataset. Using `Account_length`, `International_plan`, `Voice_mail_plan`, `total_charge`, and `Customer_service_calls` as the input variables, and `Churn` is the target variable. Do the following:

(1) Split the data into `train` (80%) and `test` (20%) taking into account the proportion of 0s and 1s in the data. That is, if Y is the target variable, in `train_test_split` function, you need to add the extra argument `stratify = Y`.

(2) Using the `MinMaxScaler` function, and `Pipeline`, transform each of the variables in the `train` dataset to a 0-1 scale:

(i) Estimate the optimal lambda for the LASSO model using default values for lambda in scikit-learn and 5-folds.

(ii) Perform LASSO as a variable selector (using the optimal lambda from previous step (i)).

Repeat steps (1)-(2) 1000 times. Store the estimated model coefficients of each iteration in a data-frame. Remove the variables, whose estimated coefficients is 0 more than 200 times, from the `telecom_train` and `telecom_test` datasets.

(e) (45 points) Consider the `telecom_train` dataset. Using `Churn` as the target variable, and the remaining variables as the input variables. Do the following:

(i) Split the data into 5-folds taking into account the proportion of 0s and 1s in the data. Notice that you can conduct k-folds splitting of the data taking into account of the proportion of 0s and 1s in the data using the `StratifiedKFold` function from `sklearn.model_selection` library.

(ii) Using MinMaxScaler, and make_pipeline transform all the input variables in the train and test datasets to 0-1 scale.

- Build a logistic regression model. Use solver = 'liblinear' and penalty = 'l1' to build the logistic regression model. After that, use the model to predict on the test dataset. Using 10% as the cut-off value, compute the recall of this model. Report the average recall score across the 5-folds.

- Build a logistic regression model. Use solver = 'liblinear' and penalty = 'l2' to build the logistic regression model. After that, use the model to predict on the test dataset. Using 10% as the cut-off value, compute the recall of this model. Report the average recall score across the 5-folds.

- Build a logistic regression model. Use solver = 'saga' and penalty = 'l1' to build the logistic regression model. After that, use the model to predict on the test dataset. Using 10% as the cut-off value, compute the recall of this model. Report the average recall score across the 5-folds.

- Build a logistic regression model. Use solver = 'saga' and penalty = 'l2' to build the logistic regression model. After that, use the model to predict on the test dataset. Using 10% as the cut-off value, compute the recall of this model. Report the average recall score across the 5-folds.

(f) (30 points) Repeat part (e) 100 times. Create a visualization that shows the recall value for each of the models at each iteration. Also, report the average recall of each of the model for the 100 repetitions. Which of the two considered logistic regression using solver = 'liblinear' models would use to predict Churn? Which of the two considered logistic regression using solver = 'saga' models would use to predict Churn?

(g) (25 points) Using the MinMaxScaler function and make_pipeline, transform each of the input variables in the telecom_train and telecom_test data-frames to a 0-1 scale. Using the telecom_train build two models: the best logistic regression model using solver = 'liblinear' from part (f) and the best logistic regression model using solver = 'saga' form part (f). Using these to two models, predict the likelihood of Churn on the telecom_test data-frame. Using 10% as the cut-off value, compute the recall of each of the two models. What model would use to predict Churn?