CS57300 Midterm — Midterm — Last update: March 10, 2016

https://www.cs.purdue.edu/homes/ribeirob/courses/Spring2016/index.html

Due: March 11, 2016

Instructions and Policy: Any questions about the homework should be asked via Piazza or during office hours. You may consult your books or other reference material or your class notes. You may NOT consult any person other than the instructor or course assistant about any aspect of this exam. You are NOT allowed to submit questions to internet discussion groups (we will be monitoring). The rules of attribution apply to take home exams: All sources should be cited. If you have any questions about any part of this exam please ask via Piazza.

ABSOLUTELY NO EXTENSIONS. You have a little over 4 days to complete this midterm. **No late assignments will be accepted**. Server or computer problems will not be accepted as an excuse for late submissions. It is your responsibility to make sure that your solution is uploaded on time.

Submit your code separately (via `https://ribeiro-www.rcac.purdue.edu/midterm`) with comments and explanations. Even if the final result is wrong, the code may allow us to find the bug and award partial credit.

You need to submit your TYPED answer in PDF via Blackboard. LaTeX is typesetting is encouraged but not required. You are also **required** to submit your R or Python codes (instructions). Please write clearly and concisely - clarity and brevity will be rewarded. Refer to known facts as necessary.

Q1 (6 pts):
Nearly 2,000 users are asking for personal loans at a bank (*test data*). You need to decide if the bank should or should not lend money to these costumers. The bank should not lend money for costumers that do not repay the full amount they borrowed. Download the training, test, and submission example data `https://www.cs.purdue.edu/homes/ribeirob/courses/Spring2016/hw/midterm/bank_data.zip`. The files `Bank_Data_Train.csv` and `Bank_Data_Test.csv` have the following data format separated by commas:

1. Record Number
2. Amount Requested
3. Interest Rate Percentage
4. Loan Length in Months
5. Loan Title
6. Loan Purpose
7. Monthly Payment
8. Total Amount Funded
9. Debt-To-Income Ratio Percentage
10. FICO Range (`https://en.wikipedia.org/wiki/FICO`)
11. Status (1 = Paid or 0 = Not Paid) [in `Bank_Data_Test.csv` this field has a question mark ? as this is the value you are trying to predict]

Create a feature vector from the data to answer the following questions. You can use existing software packages but **be careful when interpreting the output and the parameters of the package**.

(OPTIONAL) The InClass Kaggle competition `https://inclass.kaggle.com/c/purduecs573-a` may be used to validate your answers. Kaggle is not officially supported by the instructors or the TAs but we will try our best to answer questions about Kaggle on Piazza.

Hints

Note that FICO Range and Loan Purpose must be encoded with a "1-of-K" vector. In python you can use the `get_dummies` function of package `pandas` to do such encoding. But note that your encoding MUST be consistent in the training and test data. Here is an example of FICO Range 1-of-K encoding using the pandas library in python:

```
import pandas as pd
encoded = pd.get_dummies(pd.concat([train['FICO Range'],test['FICO Range']], axis=0),\
                         prefix='FICO Range', dummy_na=True)
train_rows = train.shape[0]
train_encoded = encoded.iloc[:train_rows, :]
test_encoded = encoded.iloc[train_rows:, :]
```

Questions

(a) Describe how we can use a Naïve Bayes Classifier (NBC) classifier for this problem.

  (i) (Theory) Describe the equations, the inputs, and how we can classify out-of-sample items with the output.

  (ii) (Theory) Describe the importance of priors in NBC.

(b) Describe how we can use a logistic regression classifier for this problem.

  (i) (Theory) Describe the equations, the inputs, and how we can classify out-of-sample items with the output.

  (ii) (Theory) What happens if $\Phi^T R \Phi$ (slide 34 of Lecture 14) is singular? What is the reason and how can you solve this issue?

(c) Describe how we can use a SVM classifier for this problem.

  (i) (Theory) Describe the equations, the inputs, and how we can classify out-of-sample items with the output.

  (ii) (Theory) What is the advantage of a SVMs over linear regression for classification?

  (iii) (Empirical) Which kernel, linear or Gaussian seems to work best with the data? Can you give a short likely explanation of why?

(d) (Empirical) Use K-fold validation for $K \in \{5, 10, N-1\}$ to estimate the average F1 score of each of the above classifiers (NBC, Logistic, SVM including linear and Gaussian kernels), where N is the number of observations. Remember to describe K-fold validation and how you obtained the average F1 score (the average F1 score is just the average of the obtained K F1 scores). For the **Logistic Regression** classifier plot the ROC curve and give the Area Under the Curve (AUC) score.

(e) (Empirical) Using 50-fold validation perform a paired t-test to decide which classifier has the best F1 score. Is the F1 score of the best classifier significantly different (at $\alpha = 0.05$ level) than the other three classifiers?

(f) (Empirical) Choose what you think is your best solution and submit before the Friday March 11, 11:59pm deadline as your *official solution*. To submit your best solution you MUST use the link `https://ribeiro-www.rcac.purdue.edu/midterm`. Your output should have the same format as `Submission_Example.` with

- Record Number (same record numbers as `Bank_Data_Test.csv`)
- Status (1 = Paid or 0 = Not Paid)

Toy example in Python of SVM. This script works at scholar.rcac.purdue.edu where you should have an account (if you do not have an account, please contact itap@purdue.edu).

```
import numpy as np
from sklearn import svm

X = np.array([[1,2],
             [5,8],
             [8,8],
             [9,10],
             [1,0.6],
             [9,11]])

y = [0,1,0,1,0,1]

clf = svm.SVC(kernel='rbf', C = 1)
# if you want a linear kernel clf = svm.SVC(kernel='linear', C = ??)
# C = <number> is the relaxation penalty that you must choose (do not run the code with ??)

clf.fit(X,y)

print(clf.predict([0.58,0.76]))
print(clf.predict([10.58,10.76]))
```

Q2 (4 pts):
In this question you are given an **undirected** graph G_0 with edges $\langle node1, node2 \rangle$ described in `https://www.cs.purdue.edu/homes/ribeirob/courses/Spring2016/hw/midterm/edges.zip`. Nodes in G_0 with more than 10 neighbors have one edge missing at random. Answer the following questions:

(a) (Theory) Describe how we can use Shortest Paths, Common Neighbors, Jaccard Similarity, Adamic-Adar score, Katz score, and PageRank to obtain candidates for these missing edges. Can you give advantages and disadvantages of each these methods?

(b) (Empirical) For each node with more than 10 neighbors in G_0 give 5 candidates (using the method of your choosing) for the missing edge. Choose what you think is your best solution and submit before the Friday March 11, 11:59pm deadline as your official solution. To submit your best solution you MUST use the link http://ribeiro- www.rcac.purdue.edu/midterm. Your output should have the same format as edges.csv:

```
<node_id1>,<candidate_node1>
<node_id1>,<candidate_node2>
...
```

Hint: You may want to create a training and a validation sets to test the accuracy of your candidate solutions.

Hint for better accuracy (Not Required, Advanced Users Only): You may want to use the training and validation sets with the above scores to build a SVM classifier. This is not required but it might give you better performance. Another way to get better accuracy is reconstructing the adjacency matrix via SVD decomposition using the $K \ll N$ principal values, where N is the number of nodes in the graph.