Applied Machine Learning in Economics
Honors Project

The goal of the honors project is to learn how to analyze a data set that is too large to fit into memory and to provide insights into the predictive power of credit score on future mortgage performance. The only guidance provided by the instructor is clarification on what is included in this document.

The inability to load all of the data provides a few challenges:

- Cleaning the data into a proper format
- Creating a training set that can fit into memory.
- Evaluating test set performance on a test set that cannot be simultaneously loaded into memory

The prediction problem provides insights into predictive power of the credit score on mortgages becoming delinquent and foreclosed. There will be two labels (ever_delinquent and foreclosed) to predict with three different models (**a total of six models**). The features for each model for each label are:

1. Null
2. Credit score = combine borrower and co-borrower score at origination into one feature
3. Credit score + additional features of your choosing

Project Milestones

- Download 2003Q3 from Single-Family Loan Performance from https://loanperformancedata.fanniemae.com/lppub/index.html
- Convert the R code (last page) into Python code to aide you in loading the data
    - *Hint: character = string and numeric = float or integer*
- You should have trouble loading the data. Use the following code to determine why

```
infile = open('Path/to/file.csv', 'r')
print(infile.readline())
infile.close()
```

- Clean your data
- Determine how many loans were acquired in 2003 Q3
- Create a randomly selected training set with 100,000 observations from all acquired loans
    - The remaining loans will used as a test set
- Using the codebook from the download page (file layout & glossary), perform EDA to determine which additional features you will use
- Use **sensitivity** as your performance metric
- Train your models
    - Model 1: determine what class you are going to predict for each label
    - Model 2: a neural network with an architecture of your choosing using only the credit score feature defined above
    - Model 3: a neural network with credit score *and* your additional chosen features
- Testing
    - ***Note:*** *once cleaned, you may be able to fit the entire data set into memory. When testing,* ***you must*** *test ~100,000 observations at a time. Specifically, load ~100,000*

observations subset of the cleaned testing data, evaluate model performance on the testing subset, delete the subset, and load the next.

- Evaluate and store all six model performance metrics

- Submission
  - Provide your Jupyter Notebook(s) used in the project, purged of irrelevant code
  - Provide a one page write up with a reasonable format containing: a brief introduction; one figure (may contain multiple subplots); a table containing the six performance metrics; a discussion of the different model performances and why you are using sensitivity instead of accuracy; and a brief conclusion.

Grading

- Pass/fail

```
column_names = c("POOL_ID", "LOAN_ID", "ACT_PERIOD", "CHANNEL", "SELLER", "SERVICER",
                 "MASTER_SERVICER", "ORIG_RATE", "CURR_RATE", "ORIG_UPB", "ISSUANCE_UPB",
                 "CURRENT_UPB", "ORIG_TERM", "ORIG_DATE", "FIRST_PAY", "LOAN_AGE",
                 "REM_MONTHS", "ADJ_REM_MONTHS", "MATR_DT", "OLTV", "OCLTV",
                 "NUM_BO", "DTI", "CSCORE_B", "CSCORE_C", "FIRST_FLAG", "PURPOSE",
                 "PROP", "NO_UNITS", "OCC_STAT", "STATE", "MSA", "ZIP", "MI_PCT",
                 "PRODUCT", "PPMT_FLG", "IO", "FIRST_PAY_IO", "MNTHS_TO_AMTZ_IO",
                 "DLQ_STATUS", "PMT_HISTORY", "MOD_FLAG", "MI_CANCEL_FLAG", "Zero_Bal_Code",
                 "ZB_DTE", "LAST_UPB", "RPRCH_DTE", "CURR_SCHD_PRNCPL", "TOT_SCHD_PRNCPL",
                 "UNSCHD_PRNCPL_CURR", "LAST_PAID_INSTALLMENT_DATE", "FORECLOSURE_DATE",
                 "DISPOSITION_DATE", "FORECLOSURE_COSTS", "PROPERTY_PRESERVATION_AND_REPAIR_COSTS",
                 "ASSET_RECOVERY_COSTS", "MISCELLANEOUS_HOLDING_EXPENSES_AND_CREDITS",
                 "ASSOCIATED_TAXES_FOR_HOLDING_PROPERTY", "NET_SALES_PROCEEDS",
                 "CREDIT_ENHANCEMENT_PROCEEDS", "REPURCHASES_MAKE_WHOLE_PROCEEDS",
                 "OTHER_FORECLOSURE_PROCEEDS", "NON_INTEREST_BEARING_UPB", "PRINCIPAL_FORGIVENESS_AMOUNT",
                 "ORIGINAL_LIST_START_DATE", "ORIGINAL_LIST_PRICE", "CURRENT_LIST_START_DATE",
                 "CURRENT_LIST_PRICE", "ISSUE_SCOREB", "ISSUE_SCOREC", "CURR_SCOREB",
                 "CURR_SCOREC", "MI_TYPE", "SERV_IND", "CURRENT_PERIOD_MODIFICATION_LOSS_AMOUNT",
                 "CUMULATIVE_MODIFICATION_LOSS_AMOUNT", "CURRENT_PERIOD_CREDIT_EVENT_NET_GAIN_OR_LOSS",
                 "CUMULATIVE_CREDIT_EVENT_NET_GAIN_OR_LOSS", "HOMEREADY_PROGRAM_INDICATOR",
                 "FORECLOSURE_PRINCIPAL_WRITE_OFF_AMOUNT", "RELOCATION_MORTGAGE_INDICATOR",
                 "ZERO_BALANCE_CODE_CHANGE_DATE", "LOAN_HOLDBACK_INDICATOR", "LOAN_HOLDBACK_EFFECTIVE_DATE",
                 "DELINQUENT_ACCRUED_INTEREST", "PROPERTY_INSPECTION_WAIVER_INDICATOR",
                 "HIGH_BALANCE_LOAN_INDICATOR", "ARM_5_YR_INDICATOR", "ARM_PRODUCT_TYPE",
                 "MONTHS_UNTIL_FIRST_PAYMENT_RESET", "MONTHS_BETWEEN_SUBSEQUENT_PAYMENT_RESET",
                 "INTEREST_RATE_CHANGE_DATE", "PAYMENT_CHANGE_DATE", "ARM_INDEX",
                 "ARM_CAP_STRUCTURE", "INITIAL_INTEREST_RATE_CAP", "PERIODIC_INTEREST_RATE_CAP",
                 "LIFETIME_INTEREST_RATE_CAP", "MARGIN", "BALLOON_INDICATOR",
                 "PLAN_NUMBER", "FORBEARANCE_INDICATOR", "HIGH_LOAN_TO_VALUE_HLTV_REFINANCE_OPTION_INDICATOR",
                 "DEAL_NAME", "RE_PROCS_FLAG", "ADR_TYPE", "ADR_COUNT", "ADR_UPB")


column_classes = c("character", "character", "character", "character", "character", "character",
                   "character", "numeric", "numeric", "numeric", "numeric",
                   "numeric", "numeric", "character", "character", "numeric", "numeric",
                   "numeric", "character", "numeric", "numeric", "character", "numeric",
                   "numeric", "numeric", "character", "character", "character",
                   "numeric", "character", "character", "character", "character",
                   "numeric", "character", "character", "character", "character",
                   "character", "character", "numeric", "character", "numeric",
                   "numeric", "numeric", "character", "character", "character",
                   "numeric", "numeric", "numeric", "numeric", "numeric", "numeric",
                   "numeric", "numeric", "numeric", "numeric", "numeric", "character",
                   "numeric", "character", "numeric", "numeric", "numeric", "numeric",
                   "numeric", "numeric", "character", "numeric", "numeric", "numeric",
                   "numeric", "character", "numeric", "character", "numeric", "character",
                   "numeric", "numeric", "character", "character", "numeric", "numeric",
                   "numeric", "numeric", "numeric", "numeric", "numeric", "numeric",
                   "numeric", "numeric", "numeric", "numeric", "numeric", "character",
                   "character", "character", "character", "character",
                   "character", "numeric", "numeric")
```