AI-F19
Assignment 3 (A3)
Machine Learning

General Info
10 Point Individual Assignment Due by 2 pm Monday November 4th (see syllabus for collaboration & late submission policies). **Late assignments should be emailed to Mike Flanigan and Arjun Rao - this is a change from previous assignments**

Learning Objective
This assignment satisfies learning objective 2 (LO2) as specified in the syllabus. You will apply conceptual knowledge of core AI concepts by implementing AI algorithms, analyzing existing intelligent systems (including humans), and using existing AI tools to solve problems.

Getting Help
We want these projects to be rewarding and instructional, not frustrating and demoralizing. But, we don't know when or how to help unless you ask. If you find yourself stuck on something, contact us via Piazza or come by the office hours. If you can't make our office hours, let us know and we will be happy to schedule alternate times.

Submission
You should submit the deliverables on Gradescope under Assignment 2. If you encounter any difficulties during the submission process please contact us via email and also include a copy of your submission files before the deadline.

Deliverables
You will need to submit the following files:
- part1.py
- part2.py
- bestModel.output
- bestModel.model
- <Identikey>_A3_report.pdf

Instructions - MUST READ BEFORE GETTING STARTED
- For both parts we have provided initial code, please complete all questions within these files. **Do not use iPython notebooks.**
- Do not include any additional libraries that have not been discussed, this will cause the autograder to crash (you may use numpy)
- If you code does not run to completion locally, it will not pass the autograder
- If you submit a zip file, zip only the files asked for, not a directory

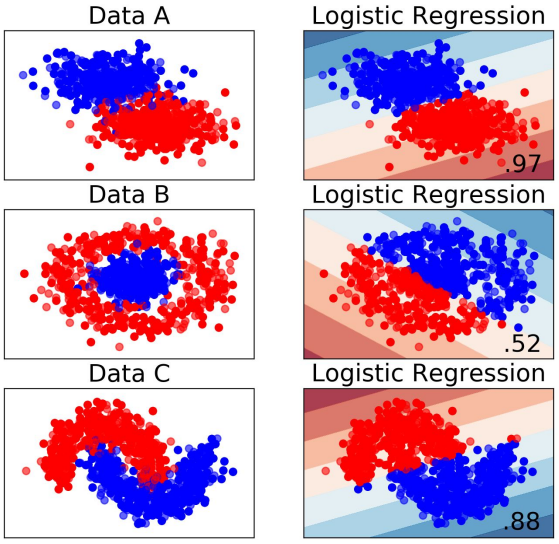
Prerequisites
For this assignment you will need some additional python libraries. You can install these using pip. You may also need to install the prerequisites for these libraries also. More information can be found on their respective manual pages.

```
pip install matplotlib, scikit-learn, pandas
```

Part 1 - Exploring Decision Boundaries [3 pts]

In part1.py, we have provided the code to produce the graph shown on the below.

Each row represents a different dataset, the left column shows the ground truth (blue or red), then the column on the right shows the classifications and decision boundary of a Logistic Regression classifier. The colour of the point indicates the classification and the colour of the background shows the decision boundaries. For example, if we consider Data C, logistic regression, a new point in the top left corner would be classified as red, whereas a point in the bottom right corder would be classified blue. The number in the bottom right corner of each graph is the accuracy of the classifier on the test set. For example, on Data C, Logistic Regression that achieved 88% accuracy on the testing set. As you add additional classifiers, the code will automatically add additional columns.



SciKit learn provides a wide range of supervised classification approaches. See here for more details

In this part of the assignment, you will use the SciKit Learn manual pages to research and implement additional classifiers. In addition to the Logistic Regression, you should implement the following classifiers:

- Naive Bayes
- SVM (SVC)
- Decision Tree
- Random Forest

Once you have found the relevant method for each classifier, you can add it to the dictionary specified on Line 20. Use the default hyperparameters for now (pass no additional parameters when declaring the classifier e.g. `LogisticRegression()`). **Don't forget to include the relevant import statements.**

In your report
Compare and contrast your five classifiers, consider each of the three datasets in turn, then consider all three together. Consider the following questions

1a. How accurate are the classifiers for the various datasets shown above?

1b. What can you conclude about the various classifiers' based on their performance on the different datasets? *Hint - a good answer will include technical details such as linear separability, decision boundaries, and so on.*

Part 2 - Implementing a Machine Learning Pipeline [7pts]

You will apply machine learning methods to an existing real-world data set as well as learn new methods along the way. The data set (credit_train.csv – available on Canvas) consists of people who have good or bad credit score (class label called Credit) based on 19 “secret” features.

Your task is to build a model (classifier) that accurately determines whether an individual (represented by one row in the dataset) should receive a good or bad credit score based on the 19 features. Your classifier should be accurate but should also be able to generalize to “new” (unseen) individuals.

A. Basic Model Building (3 points):

Using the training data (with all features) and the available learning algorithms (classifiers) in scikit learn, experiment with several classifiers for this problem. We have provided skeleton code in **part2.py**.

You should implement k fold cross validation, k=10 - scikit has some libraries to help out with this.

You should consider the following classifiers **using default hyperparameters for now**:

- Logistic Regression
- Naive Bayes
- SVM (SVC)
- Decision Tree
- Random Forest
- Two classifiers of your choice

Produce a table showing the AUROC for each classifier. (Results should be computed per fold and then averaged across folds. You should also report the standard deviation across folds).

In your report

2a.1. Include a table depicting which models you tested along with mean and standard deviation in AUROC across folds?

2a.2. Which models did you select as the “best overall model”. Why did you select it? *A good answer will consider both mean and standard deviation in performance across folds*.

2a.3. For the two classifiers of your choice, include a short description of how these classifiers work. A few sentences in your own words describing the basic ideas will suffice.

B. Hyperparameter Tuning (2 points):

A hyperparameter is a parameter of a learning algorithm (not of the model). As such, it is not affected by the learning algorithm itself; it must be set prior to training. The perfect set of hyperparameters is different for every problem.

Your task is to see if you can improve the performance of your "best" model from 2a.2 above by performing careful hyperparameter tuning on two of the algorithms (SVC and Random Forest). For each algorithm you will tune two hyperparameters **do not change the others**:

SVC: `c, gamma`

Random Forest: `max_depth, n_estimators`

For each algorithm, you will need to experiment with different values to see which combination of hyperparameters gives the best performance. SciKit already has some methods for parameter selection (see documentation- this is recommended) or you can manually try different combinations of hyperparameters (not recommended).

Hint. A nice tutorial for this process can be seen in Geron - Hands on Guide to Machine Learning with SciKit-Learn, pages 127 - 131. We have included these as a pdf on canvas.

Things to include in your report:

2b.1. Describe the technique used for hyperparameter selection. If you used a library method, describe briefly how it works.

2b.2. Briefly describe each of the four hyperparameters from above in your own words.

2b.3 Were you able to improve upon your best model from 2a.2. with hyperparameter tuning? You can use the AUROC metric for the comparison.

C. Analysis of Best Model (1 point).

Select the best model you have obtained thus far on this data set (select only one model in all - from 2a.2 or 2b.3). Carefully analyze the output of this model. You should produce a confusion matrix for this model as well as accuracy (percent correct), precision, recall, and AUROC.

File to submit:

Create a file with the output of this best model (bestModel.output) in **comma separated format** and upload this with your submission, This output should include each instance, the ground truth, and the classifier prediction.

As a check, your file should have 21 items per row (19 Features, 1 Ground Truth, 1 Predicted Label) and 501 rows (500 instances + Header), you should name the column with your predicted label "Prediction"

Things to include in your report:

2c.1. The confusion matrix along with detailed accuracy metrics (see above)

2c.2. What can you conclude about the accuracy of your model in classifying whether a person has good or bad credit?

D. Generalizability (1 point)

We have been a bit sneaky. The original data set actually contained 690 instances. But we have only made 500 of these instances (randomly selected) available to you. We have withheld the remaining 190 instances for "validation." We will use these "holdout" instances to test the generalizability of your model. **You will be scored based on how your model performs on this holdout data.**

File to submit:

You will need to submit a model object file (bestModel.model) of your best performing model from Part 2.C. Once you have chosen your hyperparameters, or used the default hyperparameters, train a model on **all of the data** (do not do cross validation) and then save that model. We have provided a function to save your model - `saveBestModel`, it takes a classifier object as a parameter.

Things to include in your report:

2d.1. Sometimes, we have difficulty loading the model files, so be sure to include the name of your model and hyperparameter values so we can rebuild it if necessary.

Part 3 - Extra Credit

The model you submit in Part 2.D will compete against the models submitted by the other students. We will use the AUROC metric. The students with the highest AUROC values will receive the following bonus points: First Place – 0.75 points; Second Place – 0.5 points; Third Place – 0.25 points. Good luck!!

APPENDIX 1 : Scoring Rubric

The scoring rubric **for your report** is based on the Kentucky General Scoring Rubric from the Kentucky Department of Education (KDE).

Score	Description
Category 4 (Score 90%-100%)	● The student completes all important components of the task and communicates ideas clearly. ● The student demonstrates in-depth understanding of the relevant concepts and/or process. ● Where appropriate, the student chooses more efficient and/or sophisticated processes. ● Where appropriate, the student offers insightful interpretations or extensions (generalizations, applications, analogies).
Category 3 (Score 70%-90%)	● The student completes most important components of the task and communicates clearly. ● The student demonstrates an understanding of major concepts even though he/she overlooks or misunderstands some less important ideas or details.
Category 2 (Score 60%-70%)	● The student completes some important components of the task and communicates those clearly. ● The student demonstrates that there are gaps in his/her conceptual understanding.
Category 1 (Score 10%-60%)	● The student shows minimal understanding. ● The student addresses only a small portion of the required task(s).
Category 0 (Score 0)	● Response is totally incorrect or irrelevant.
Blank (Score 0)	● No response.