Credit Card Fraud Detection Lab

Please read all of the guidelines carefully before submitting the lab. ☺ Each step is **10 points**, the report is **30 points** and there are ***130 points*** in total.

Due date: Wednesday, April 14, 11:59 PM. Late submissions will be accepted with a penalty!

Deliverables:

1) The code of the project in **.py** or **.ipynb** format (in several files (one file for each question) or one single file)
2) The lab report written with ***LaTeX*** and exported in **.pdf** format (one file)

Guidelines – Before You Start

1) You will be using the **`Python`** programming language for this lab. You need to write your codes in an empty **.py** OR **.ipynb** file. Don't use the files we used for labs to submit your homework or other assignments (those submissions will not be graded.).
2) Make sure that you provide many comments to describe your code and the variables that you created.
3) Please use the ***IEEE*** journal template on ***overleaf.com***. Here is the link: https://www.overleaf.com/latex/templates/preparation-of-papers-for-ieee-sponsored-conferences-and-symposia/zfnqfzzzxghk
   To be able to work on ***overleaf.com***, you will need to register first (you can also compile your ***LaTeX*** file locally.)
4) For some of the code, you may need to do a little bit of "**Googling**" or review the documentation.

What is a credit card fraud and how to detect it?

Credit card fraud is the unauthorized use of another person's credit card—or card information—to make purchases or access funds through cash advances using the victim's account.

Companies detect fraud is by looking for unusual activity. They know what your usual charging habits are -- what towns and regions you typically spend your money in, what stores you frequent, what amounts you tend to charge, and so on. For instance, if you live in Georgia but all of a sudden the company spots transactions happening in a distant location, such as Minnesota, it might decline the next transaction until it gets verification that you're really behind the charge. Similarly, if you typically charge between $1,000 and $2,000 per month and there's suddenly a $4,500 charge, the card company may well refuse the transaction or require you to verify it.

Credit card companies use a combination of technology and humanity to fight fraud, employing automated fraud detection algorithms across massive amounts of data collected from millions of

customers and hundreds of millions of cards. Once a transaction is flagged as a possible problem, humans can follow up, contacting the customer. Even this is being increasingly automated, with some card holders receiving texts asking them to verify a suspicious transaction on their account.

Lab - Coding

In this assignment, you will be working with a dataset on credit card fraud. The dataset can be found in the assignment folder (**creditcard.csv**). More information about the data set can be found on *Kaggle*: https://www.kaggle.com/mlg-ulb/creditcardfraud.

This assignment will be different from the previous assignments, since you will be working with data that is not clearly labeled due anonymity and privacy issues. We will be trying to understand the unknown. Here are the features in the dataset:

Time: Number of seconds elapsed between the latest transaction and the first transaction in the dataset

V1 – V28: 28 different features about users and the nature of transaction that have not been clearly labeled for privacy and security issues

Amount: Transaction amount

Class: a label, 1 for fraudulent transactions, 0 otherwise

Open the Excel files and take a look at them before starting. The names of the columns should be self-explanatory. For any questions, please refer to the link where the data was downloaded from or send an e-mail. Before you start with the questions below, create a new and empty folder called **data_350_creditcard_lab**. Call the file where you will write your code **data_350_creditcard_lab**, as well.

Recording the time

In this lab, we will be putting into practice some of the very recent techniques we have learned. We will also be recording how much computational time we need for some questions in the analysis. Please make sure that you record the time it takes to answer when noted in the question.

Measuring the elapsed time can easily be done with the `Time` module. At the start of the code, add `time.time()` to your code and finish your code with by finding the difference between current time and the starting point:

```
start_time = time.time()
# your code
elapsed_time = time.time() - start_time
```

Part I: Exploratory Analysis and Feature Engineering (40 points)

Q1: Let's compare fraudulent and non-fraudulent cases very simply:

(i) Split your dataset into two parts, ***fraudulent*** and ***non-fraudulent*** transactions. Excluding the class label (***Class*** variable), compare the means and medians of the following columns:

V1 through ***V 28***
Amount
Time

By looking at the columns ***V1*** through ***V28***, report the **means** and **medians** of the top five (5) ***V**** variables for which the average of the differences seem to be the greatest along with ***Amount*** and ***Time***. The results (one table for mean, and one for median) should look like:

	Amount	**Time**	**V…**	**V…**	**V…**	**V…**	**V…**
Fraudulent							
Non-fraudulent							

(ii) Now do a ***min-max normalization*** on your dataset such that every variable has a range between 0 and 1, add the scaled values as new columns to your dataset. We will be using this data later. And repeat the same analysis you did in (i). Summarize your findings.

Q2: Let's do some "downsizing" in our dataset. Excluding the fraudulent cases, from among the non-fraudulent part of the dataset draw 10,000 random samples (**without replacement**). Use the `pandas.DataFrame.sample` function and set **random_state = 350**. and Save this dataset as a new file. And let's do some "dimensionality reduction" on this smaller dataset.

(i) **Please report the elapsed time for this question**. Do ***PCA*** on the new dataset, set the number of principal components to two (2). Visualize the new data in a 2D scatter plot. In the scatter plot, use different colors for ***fraudulent*** and ***non-fraudulent cases***. Answer the following:

a. Are there "some" number of clusters you can identify? How many clusters are there?
b. Is there a particular cluster where the ***fraudulent*** cases seem to be located?

(ii) **Please report the elapsed time for this question**. Now repeat the same analysis by using another dimensionality reduction technique: **t-SNE**. You can read more about t-SNE on Wikipedia: https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding
Repeat the analysis you have done above in (i). Is there any difference?

(iii) After having read about t-SNE, do you think one of those dimensionality reduction methods is better than the other method for this dataset?

Q3: Using the ***min-max scaled variables*** in the smaller dataset we created, let's compare some distances between the observations in our dataset. We will be comparing distances using `Manhattan` and `Euclidean` distances.

Please report the elapsed time for each distance calculation separately in this question. And please do the following:

(i) Find the average pairwise `Manhattan` and `Euclidean` distances between ***fraudulent*** and ***non-fraudulent*** cases.

(ii) Find the average pairwise `Manhattan` and `Euclidean` distances between ***fraudulent*** and ***fraudulent*** cases.

(iii) Please report your findings in a table similar to the one below. Do you think there are important differences between fraudulent and non-fraudulent cases in terms of distance?

Distance	**Manhattan**	**Euclidean**
Fraudulent to Fraudulent		
Fraudulent to Non-fraudulent		

Q4: Let's take a look at the ***Amount*** and ***Time*** columns. Divide both columns into ***deciles*** and report the count values of fraudulent and non-fraudulent cases in each decile. You need to create two (2) tables, one for each feature. Each table should look like the following:

Counts	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Fraudulent										
Non-fraudulent										

At the end, please report if you can see any trends in the table. Are there particular deciles in the dataset where the count difference between ***fraudulent*** and ***non-fraudulent*** cases seems to be important?

Part II: Clustering and Classification (40 points)

Please use the smaller dataset and the five scaled V*'s you obtained in Q1 (ii) for the rest of the analysis.

Q5: Run a ***kmeans++*** clustering algorithm using the min-max scaled variables by setting the number of clusters **k = 2**. Set **random_state = 350**. **Please report the elapsed time for running the kmeans++ algorithm.** Please also calculate the ***precision***, ***recall, F1 score, Silhouette Index, and Calinski-Harabasz Index*** for your results and record their values.

Hint: For kmeans++, the cluster results with the lower number of cases should be 1 (if 0 is assigned automatically, please convert 0's into 1's.)

Q6: Now run a **DBSCAN** algorithm. Set **`eps`** to the average `Euclidean` pairwise distance between fraudulent and non-fraudulent cases you found in **Q3**. Set **`minPts`** to be equal to **10**. **Please report the elapsed time for running the DBSCAN algorithm.** Please also calculate the ***precision***, ***recall, F1 score, Silhouette Index, and Calinski-Harabasz Index*** for your results and record their values.

Hint: For DBSCAN, the cluster results with the lower number of cases should be 1 (if 0 is assigned automatically, please convert 0's into 1's.)

Q7: Let's also run a **logistic regression** and "**zero-inflated**" model by using all of the variables in the dataset. These models are similar to Logistic Regression, but they model zero and non-zero values differently. They are used in contexts, when most observations are labeled as zeros.

Here is a page that you can take a look at to run a zero-inflated model using `Python`:
https://bryansweber.com/2018/10/26/python-and-zero-inflated-models/

Please report the elapsed time for running the logistic regression and zero-inflated algorithm. As in previous questions, please also calculate the ***precision***, ***recall, F1 score, Silhouette Index, and Calinski-Harabasz Index*** for your results and record their values.

Model	**Precision**	**Recall**	**F1 Score**	**Silhouette Index**	**Calinski-Harabasz Index**
Kmeans++					
DBSCAN					
Zero-inflated					
Logistic regression					

Q8: Finally, let's code a **kmeans** algorithm manually (not kmeans++). Compare the performance of your manually coded algorithm with `sklearn.cluster.KMeans` (**init = 'random'** and **random_state = 350**). **Please report the elapsed time for running each of these algorithms**. Which one is faster?

Part III: Ensemble Learning (10 points)

Q9: Let's check a few basic ensembling techniques to see if they would make our model any better. Please do the following.

(i) In **Part II**, we ran four different classification algorithms. Using the **`max-voting`** ensembling technique, please re-consider the labels you assigned to each class and update the labels if necessary. Please also calculate ***precision***, ***recall, F1 score, Silhouette Index, and Calinski-Harabasz Index*** for your results and record their values.

(ii) Now, let's do **`stacking`**. Run a `logistic regression` model by using the three lists of clustering results you obtained in **Part II** as your input variables. Please also calculate ***precision***, ***recall, F1 score, Silhouette Index, and Calinski-Harabasz Index*** for your results and record their values. Finally, report your findings in a table format:

Model	**Precision**	**Recall**	**F1 Score**	**Silhouette Index**	**Calinski-Harabasz Index**
Max-Voting					
Stacking					

Part IV: Visualization (10 points)

Q10: Finally, let's do a few visualizations:

- Create a **pairplot** where you look at ***Amount***, ***Time***, and the five scaled ***V****'s you obtained in **Q1**. Use a different color for ***fraudulent*** and ***non-fraudulent*** cases.
- Create two histograms overlaid with distribution plots for ***Amount*** and ***Time*** variables. Use different colors for the two graphs.

Part V: Creating the lab report (30 points)

Let's write a lab report with a special emphasis on the performance of algorithms we have applied. Did anything work? Was our analysis helpful at all?

Q11: Write a report (2 pages) that includes all of your findings and the visuals that you created. The report that you write should use the *IEEE format* and include the following sections:
Abstract: A short summary of your report
Introduction: A summary of what you expected and did, and two-three of your most significant findings
Data: A description of your dataset (number of observations, what kind of information is there, descriptive statistics table etc.)
Results: A summary of the analyses you completed for the lab + a discussion on the ***results*** that talks about these topics:

- What are some of the main findings you have?
- Which variable(s) seem(s) to explain the outcomes the most?
- What would be some of the ways to improve this analysis?
- Do you see any typical causal inference problems (***selection bias***, ***simultaneity***, ***omitted variable bias***) in this analysis?

Final step:
Send what you saved in your submission folder **(data_350_creditcard_lab)** through *Notebowl*.