

**Cairo University**



**Faculty Of Engineering**

CMPN451 – Data Mining, Big Data and Analytics

Project Proposal

**Names:**

Ahmed Mohamed Ismail Nabeel - 1180501

Moaaz Mohamed Elsherbini - 1180528

Mostafa Ashraf Ahmed Kamal - 1180406

Nader Youhanna Adib Khalil - 1180477

# Problem Description

Our problem consists of predicting whether a credit card transaction is fraudulent given some information about the transaction like:

* distance\_from\_home - the distance from home where the transaction happened.
* distance\_from\_last\_transaction - the distance from last transaction happened.
* ratio\_to\_median\_purchase\_price - Ratio of purchased price transaction to median purchase price.
* repeat\_retailer - Is the transaction happened from same retailer.
* used\_chip - Is the transaction through chip (credit card).
* used\_pin\_number - Is the transaction happened by using PIN number.
* online\_order - Is the transaction an online order.

# Project Pipeline

1. Read the dataset into a dataframe
2. Plot the data and get insights about it from plots to know which type of preprocessing to apply
3. Remove missing and illogical data
4. Handle Outliers
5. Discretize data for A priori algorithm only
6. Standardize the data (Z-Score Normalization)
7. Feature Engineering: PCA
8. Split the data into train, validation and test sets
9. Supervised Learning algorithms:
   1. Logistic Regression
   2. Minimum Distance Classifier
   3. Naïve Bayes
   4. KNN (with and without MapReduce)
10. Unsupervised Learning algorithms:
    1. K-means clustering (using MapReduce)
    2. A priori algorithm

# Analysis and Solution to the problem

1. Data Preprocessing:
   1. Remove missing data
   2. Remove illogical data (negative values for distance)
      * + The data is large enough that w do not have to impute
   3. Handle Outliers: Remove data that exceeds the 99th percentile or is below the 1st percentile for numerical columns
   4. Standardize the data: Z-Score Normalization

where: is the mean and is the standard deviation

* 1. Feature Extraction: Principal Component Analysis (PCA)

Where is the original data, is the vector of means and is the vector of Eigenvectors of the covariance matrix of

* 1. Split the data:
     + - 70%: train set
       - 15%: validation set
       - 15%: test set

1. Data Visualizations:

A blue circle with a pink triangle and black text

Description automatically generated with low confidence

A screen shot of a graph

Description automatically generated with low confidence

A screen shot of a white square

Description automatically generated with low confidence

A screen shot of a graph

Description automatically generated with medium confidence

A picture containing colorfulness, square, rectangle, screenshot

Description automatically generated

1. Extracting insights from data:

After application of the A priori algorithm, we find that the following rules have the highest confidence (and lift closest to 1):

(All the rules are kept in apriori.txt. We have discovered 43 rules with confidence higher than 50%)

('offline', 'no\_pin', 'repeat', 'no\_chip') => ['not\_fraud'] (confidence: 0.99, lift: 1.09)

('not\_fraud', 'offline', 'repeat', 'no\_chip') => ['no\_pin'] (confidence: 0.90, lift: 1.00)

('chip', 'online', 'no\_pin', 'repeat') => ['not\_fraud'] (confidence: 0.90, lift: 0.98)

('offline', 'repeat', 'no\_chip') => ['no\_pin', 'not\_fraud'] (confidence: 0.89, lift: 1.10)

('not\_fraud', 'online', 'repeat', 'no\_chip') => ['no\_pin'] (confidence: 0.88, lift: 0.98)

* We can note that the expected rule (‘far from home’, ‘far from last transaction’… ) => [‘fraud’] does not appear as none of these items are frequent in the dataset. In fact, no rule has ‘fraud’ in it. As we have seen from the plots, 8.7% of the transactions are fraudulent.
* This means that while we cannot confidently predict if a transaction is fraudulent, we can, however, predict with high confidence that it is not fraudulent given some conditions.

1. Model Classifier/Training:

We have used the following classification techniques:

* + 1. Logistic Regression with cross-validation, max iterations = 1000
    2. Minimum Distance Classifier
    3. Naïve Bayes with Gaussian conditional probabilities:

Where:

* is the vector of means
* is the covariance matrix
* is the number of fetures, 7 in our case
  + 1. KNN (with and without MapReduce)

KNN without MapReduce takes too long to run, and this is because we have 700,000 rows in the train set and 150,000 rows in the validation set; so a sequential approach would need at least iterations each of which needs to compute the distance between two points, which itself needs 7 computations.

So we have run it on a reduced set and used the validation set to get the best value of k, but we found no difference between values of k = 3, 5 and 7

We have also implemented:

1. Kmeans clustering using MapReduce (PySpark)
2. A priori algorithm

# Results and Evaluation

1. Logistic Regression: Accuracy = 96.36%
2. Minimum Distance Classifier: Accuracy = 91.30%
3. Naïve Bayes: Accuracy = 87.35%
4. KNN:
   1. On validation set:
      1. K = 3: 91.14%
      2. K = 5: 91.14%
      3. K = 7: 91.14%
   2. On test set: 91.39%
5. Kmeans: WSS = 3311305539.2

# Unsuccessful trials

KNN without PySpark would not finish execution. We have however included the function without ever calling it.

KNN with PySpark would take tremendous runtime though is faster (we could see this with the tqdm library in python that tracks the progress of a loop). We have run it on a reduced set.

# Enhancements and future work

Accuracies could be enhanced by collecting more data and by developing more sophisticated methods such as neural networks.