**Project Name**

Group #7

Nathaniel Miller [nmill120@kent.edu](mailto:nmill120@kent.edu)

Joseph Putaturo [jputatur@kent.edu](mailto:jputatur@kent.edu)

Daniel Write [dwrite@kent.edu](mailto:dwrite@kent.edu)

**1. Introduction**

* Motivation examples of this project

I(Nathaniel Miller) am planning on becoming an actuary. One of the things that an actuary would have to do is determine the risk of lending to a specific person. This risk can be measured by the probability of a given person defaulting on their loan given several characteristics.

**2. Project Description**

* Brief descriptions of your project
* Challenges and technical contributions (new problems or new solutions?) in your project
* The workload distribution for each member in your team

We will analyze a dataset of 308,000 borrowers and create a model to predict the probability that a given borrower will default on their loan. This includes 25,000 borrowers who defaulted and 283,000 borrowers who did not default. For each borrower, 120 pieces of information are known, including amount of income, amount of the loan, age, type of residence, number of children, and gender. We will find the amount of correlation between each piece of information and the result of the loan, as well as the correlation between each pair of the attributes. We will also find the likelihood of current borrowers to default in the future.

Nathaniel Miller will do the design specification and program some statistical algorithms.

Joseph Putaturo will do predictions of current borrowers defaulting in the future.

Daniel Write will work on finding or generating data and arranging the graphical display thereof as well as some of the programming aspects of this project.

<https://www.kaggle.com/datasets/gauravduttakiit/loan-defaulter/data?select=columns_description.csv>

**3. Background**

* Related papers (or surveys for graduate teams)
* Software tools (DBMS, GUI, IDE, existing library, …)
* Required hardware
* Related programming skills (functions, Internet programming, object-oriented programming, distributed environment, etc.)

Need some data on how this could be done, this is a public dataset so see if there are other approaches.

The data is given as a CSV file, so a simple parser will be written. The program for processing the data and finding correlations will be written in Rust for ease of parsing and quick execution time. Due to this, the entire system will just be running on my personal computer. The data is sufficiently small that no internet connection is needed, and all records can be read directly from the file.

**4. Problem Definition**

* Formal (mathematical) definitions of problems
* Challenges of tackling the problems
* A brief summary of general solutions in your project

A record is given as a tuple of 120 values. This project will find an estimate for the probability function for any record v by finding all points in the dataset p where , where . For numerical data, , with being the standard deviation of and being the correlation coefficient of and , using . For categorical data, when and 0 otherwise, with being the Cramer’s V between and ([Cramér's V - Wikipedia](https://en.wikipedia.org/wiki/Cram%C3%A9r%27s_V#Calculation)). will be chosen such that most data points have at least 20 points in the dataset within a distance of . Let , then , where if c is a record that defaulted and if c is a record that did not default. Dimensions where is small will be skipped when calculating to allow faster querying. The choice of L1 norm was chosen to allow simple short circuiting of attributes, so once the accumulated , the algorithm will stop calculating p and move on to the next point. For dimensions which have high correlations, one of the dimensions will be pruned out to avoid double-counting correlations (correlations between numericals will be done using Pearson correlation, between categoricals by Cramer’s V, and between a numeric and a categorical will be done using the Pearson correlation between each point and the mean of the category of each point). Dimensions are searched from highest value of to the lowest to accelerate this pruning. Outliers can be detected by comparing with , and values where the difference is large should be set as outliers. These outliers may be removed and the accuracy of the model will be tested with and without outliers.

**5. The Proposed Techniques**

* Framework (problem settings)
* Details of major techniques (e.g., pruning methods in lemmas/theorems; illustrated with toy examples)
* Encoding or indexing of data
* Query processing algorithms (pseudo code) and query optimizations
* This section can be split into multiple sections if you have many contents to present

**6. Visual Applications**

* GUI design
* Design modules (with descriptions, figures, and/or flowcharts)

Graph points considered may be graphed, values given in pie chart to visualize probability.

**7. Experimental Evaluation**

* Experimental settings
  + Descriptions of real/synthetic data sets
  + Competitors (baseline method, or existing techniques to compare with)
  + Parameter settings
  + Evaluation measures
* The performance report (pruning power, recall/precision/f-measure, CPU time, I/O cost, communication cost, index construction time/space, etc.)
* Screen captures

**8. Future Work**

* Possible project extensions

**9. References**

[1] FirstName1 LastName1, FirstName2 LastName2, and FirstName3 LastName3. Conference paper title. In *XXX*, pages XXX-XXX, 20XX.

[2] FirstName1 LastName1, FirstName2 LastName2, and FirstName3 LastName3. Journal paper title. In *XXX*, Vol. X, No. X, pages XXX-XXX, 19XX.