**Background**

Financial scorecards are statistical tools used by banks to evaluate a potential borrower’s risk of defaulting. Scorecards are statistical models based on historical lender data and can be consulted to determine whether to approve a loan. This process helps banks maximize profits on loans and prevents customers from taking on unmanageable debt.

**Method**

I started by researching the theory behind scorecard models. A major resource was Naeem Siddiqi’s book Credit Risk Scorecards which details both the theory and development behind credit scorecards. The book covers all the basic theoretical bases of scorecard modeling. Another research paper, in Chinese, details the process of calculating the score from the output of the logistic regression: <https://wenku.baidu.com/view/6a1ca267cc22bcd127ff0cdb.html>.

Afterwards, I searched for data to build the model around. I found an ongoing online competition on home credit default risk on Kaggle.com, a data science competition hosting website (<https://www.kaggle.com/c/home-credit-default-risk/data>). The main file used was application\_train.csv. The data is structured into rows, each row representing one loan. The first column contains the unique loan ID. The second column contains the Target, representing whether that loan was defaulted or not (1 = defaulted, 0 = not defaulted). The rest of the many columns contain details on various aspects of the loan recipient. It is important to note that some features are categorical while others are continuous.

I then began coding the scorecard using python. I primarily learned and used the ‘pandas’ package for data manipulation and the ‘sklearn’ package for statistical analysis. The code is documented step by step via comments so reading through it is the best way to understand the implementation. There is also a supplemental PowerPoint detailing individual sections of the code. Components implemented so far are:

* Reading in the data and converting it to a manipulable dataframe

**Results**

**Future Development**

* Create functionality for differentiation of categorical and continuous values, and deal with each appropriately. I already implemented WOE for continuous features.
* Verify the score calculation function is correct. It is implemented but not yet tested.
* Test the model by metrics such as confusion matrix, ROC, AUC, etc.
* Calculate the proper factor and offset for the score calculation
* Select best features
* Dimensionality reduction