# Prudential Life Insurance Assessment

https://www.kaggle.com/c/prudential-life-insurance-assessment

## Description

Picture this. You are a data scientist in a start-up culture with the potential to have a very large impact on the business. Oh, and you are backed up by a company with 140 years' business experience.

Curious? Great! You are the kind of person we are looking for. Prudential, one of the largest issuers of life insurance in the USA, is hiring passionate data scientists to join a newly-formed Data Science group solving complex challenges and identifying opportunities. The results have been impressive so far but we want more.

## The Challenge

In a one-click shopping world with on-demand everything, the life insurance application process is antiquated. Customers provide extensive information to identify risk classification and eligibility, including scheduling medical exams, a process that takes an average of 30 days.

The result? People are turned off. That’s why only 40% of U.S. households own individual life insurance. Prudential wants to make it quicker and less labor intensive for new and existing customers to get a quote while maintaining privacy boundaries.

By developing a predictive model that accurately classifies risk using a more automated approach, you can greatly impact public perception of the industry. The results will help Prudential better understand the predictive power of the data points in the existing assessment, enabling us to significantly streamline the process.

## Evaluation

Submissions are scored based on the quadratic weighted kappa, which measures the agreement between two ratings. This metric typically varies from 0 (random agreement) to 1 (complete agreement). In the event that there is less agreement between the raters than expected by chance, this metric may go below 0.

The response variable has 8 possible ratings. Each application is characterized by a tuple (ea,eb), which corresponds to its scores by Rater A (actual risk) and Rater B (predicted risk). The quadratic weighted kappa is calculated as follows.

First, an N x N histogram matrix O is constructed, such that Oi,j corresponds to the number of applications that received a rating i by A and a rating j by B. An N-by-N matrix of weights, w, is calculated based on the difference between raters' scores:

An N-by-N histogram matrix of expected ratings, E, is calculated, assuming that there is no correlation between rating scores. This is calculated as the outer product between each rater's histogram vector of ratings, normalized such that E and O have the same sum.

## File descriptions

* train.csv - the training set, contains the Response values
* test.csv - the test set, you must predict the Response variable for all rows in this file
* sample\_submission.csv - a sample submission file in the correct format

## Data fields

Variable - Description

* **Id -** A unique identifier associated with an application.
* **Product\_Info\_1-7 -** A set of normalized variables relating to the product applied for
* **Ins\_Age -** Normalized age of applicant
* **Ht -** Normalized height of applicant
* **Wt -** Normalized weight of applicant
* **BMI -** Normalized BMI of applicant
* **Employment\_Info\_1-6 -** A set of normalized variables relating to the employment history of the applicant.
* **InsuredInfo\_1-6 -** A set of normalized variables providing information about the applicant.
* **Insurance\_History\_1-9 -** A set of normalized variables relating to the insurance history of the applicant.
* **Family\_Hist\_1-5 -** A set of normalized variables relating to the family history of the applicant.
* **Medical\_History\_1-41 -** A set of normalized variables relating to the medical history of the applicant.
* **Medical\_Keyword\_1-48 -** A set of dummy variables relating to the presence of/absence of a medical keyword being associated with the application.
* **Response -** This is the target variable, an ordinal variable relating to the final decision associated with an application

### The following variables are all categorical (nominal):

* Product\_Info\_1, Product\_Info\_2, Product\_Info\_3, Product\_Info\_5, Product\_Info\_6, Product\_Info\_7
* Employment\_Info\_2, Employment\_Info\_3, Employment\_Info\_5, InsuredInfo\_1, InsuredInfo\_2, InsuredInfo\_3, InsuredInfo\_4, InsuredInfo\_5, InsuredInfo\_6, InsuredInfo\_7
* Insurance\_History\_1, Insurance\_History\_2, Insurance\_History\_3, Insurance\_History\_4, Insurance\_History\_7, Insurance\_History\_8, Insurance\_History\_9
* Family\_Hist\_1
* Medical\_History\_2, Medical\_History\_3, Medical\_History\_4, Medical\_History\_5, Medical\_History\_6, Medical\_History\_7, Medical\_History\_8, Medical\_History\_9, Medical\_History\_11, Medical\_History\_12, Medical\_History\_13, Medical\_History\_14, Medical\_History\_16, Medical\_History\_17, Medical\_History\_18, Medical\_History\_19, Medical\_History\_20, Medical\_History\_21, Medical\_History\_22, Medical\_History\_23, Medical\_History\_25, Medical\_History\_26, Medical\_History\_27, Medical\_History\_28, Medical\_History\_29, Medical\_History\_30, Medical\_History\_31, Medical\_History\_33, Medical\_History\_34, Medical\_History\_35, Medical\_History\_36, Medical\_History\_37, Medical\_History\_38, Medical\_History\_39, Medical\_History\_40, Medical\_History\_41

### The following variables are continuous:

* Product\_Info\_4
* Ins\_Age, Ht, Wt, BMI
* Employment\_Info\_1, Employment\_Info\_4, Employment\_Info\_6
* Insurance\_History\_5
* Family\_Hist\_2, Family\_Hist\_3, Family\_Hist\_4, Family\_Hist\_5

### The following variables are discrete:

* Medical\_History\_1, Medical\_History\_10, Medical\_History\_15, Medical\_History\_24, Medical\_History\_32
* Medical\_Keyword\_1-48 are dummy variables.

## Proposed Solution

Combination of Neural Network and Ensemble Modelling

* Input Layer: Binned input variables.
* Hidden Layer 1: 6 Decision Tree Classifiers, 6 Random Forest Classifiers, and 6 XG Boost Classifiers.
* Hidden Layer 2: KNN model is used the combine the outputs of the Decision Trees, another KNN model is used to combine the outputs of the Random Forest, and one more KNN model is used to combine the outputs of the XB Boost.
* Output Layer: KNN model is used the combine the outputs of the above KNN models.