Prudential Life Insurance – Analysis and Classification of High and Low Risk Applicants

Ashok Bhatraju	Nikita Bawane	Ritu Gangwal
UIN : 670248723	UIN : 661069000	UIN : 670646774

Abstract

Risk assessment is a vital element in the life insurance business to categorize their applicants. Data in the insurance market is characterized by asymmetric information and the proficiency of the insurance companies is based on how well they analyze the data and perform the risk classification of their applicants. Companies generally perform the underwriting process to make decisions on applications and to price policies accordingly. With the increase in the amount of data and advances in data analytics, the underwriting process can be fastened by developing good predictive models which categorize the risk applicants efficiently. This paper aims at analyzing different classification models developed using Prudential Life Insurance dataset from Kaggle in order to find the risk profile of an applicant.

Concept of dimensionality reduction is introduced in this paper. The data dimension has been reduced by feature selection techniques and feature extraction namely, Correlation-Based Feature Selection (CFS) and Principal Components Analysis (PCA). Machine learning algorithms, namely K-Nearest Neighbors (KNN), Logistic Regression, Naïve Bayes, and Support Vector Machine (SVM) were implemented on the dataset to predict the risk level of applicants.

Keywords: Life Insurance, Machine learning, Predictive analytics, Correlation, Principal components, Dimensionality reduction, Support Vector Machine, K-Nearest Neighbors, Naïve Bayes, Logistic Regression, Confusion Matrix

1 Introduction

In the recent years, Insurance has become an indispensable part of our lives as it helps protect individuals from unforeseen losses. There are several companies which provide different insurance such as Life Insurance, Auto Insurance, Renter's Insurance etc. A life insurance pays out to one's beneficiary upon their death. It is essential for the life insurance companies to examine an individual's lifestyle to determine their life expectancy. This form of risk analysis enables them to categorize their applicants into various categories (such as High or Low). The efficiency of the insurance companies is based on how well they can evaluate applicants' profile for risk and calculate their premiums accordingly. Indicators used to evaluate applicants are their family health history, personal health status etc. Age, gender, dangerous hobbies significantly increases an applicant's premium as well. These parameters weigh heavily as future indicators of cost for the insurance companies.

Over the years, life insurance companies have strived towards selling their products efficiently, and before accepting an application, a series of tasks are undertaken during the underwriting process. A general process of applying for a life insurance involves four major steps: One, application form with personal information such as applicant's medical history and medical details, beneficiary information is submitted. Next, a medical exam is held, after which insurance company reviews the personal information as well as medical history in order to make an insurance policy for an individual.

2 Problem Statement

The process of underwriting involves an extensive gathering of information about the applicant. After a series of medical tests and evaluation, underwriter assesses the risk profile of the applicant and subsequently

calculates the premiums. This process takes an average of 30 days which plays out negatively for the companies as people are reluctant in buying slow services. As the process of underwriting is lengthy and time-consuming, customers can switch to a competitor or not buy an insurance at all. This consequently leads to customers being unsatisfied and a decrease in policy sales. Lack of proper underwriting practices can consequently lead to customers being unsatisfied and a decrease in policy sales. Thus, it is crucial improving the underwriting process to enhance customer acquisition and customer retention.
Predicting the important factors which impact the risk assessment process can help to streamline the process, making it more efficient and economical. Therefore, it is extremely crucial to automate the risk evaluation of applicants in order to increase customer satisfaction and customer retention. For this project, we have used prudential life insurance data from Kaggle. The challenging part for the company is that the application process time is antiquated and the goal for this project is to help them to enhance the efficiency of processing time as well as reduce labor intensive analysis for new and existing customers.

3 Related Work

In order to predict the probability of default Yeh and Lien, 2009 used 6 different data mining techniques namely, - K-nearest neighbor (KNN), Logistic Regression, Naïve Bayes, Artificial Neural Networks, Classification trees and Discriminant Analysis. Their study focused on predicting the probabilities rather than just classifying the customers as defaulters and non-defaulters.

It applied the Sorting Smoothing Method (SSM) for estimating the real probability of default from the model. For evaluating and comparing the performance of different models the Area ratios in the lift charts were used. From the lift curves and the accuracy rate of the 6 techniques, they observed that on the training data, K-nearest neighbor classifiers and classification trees had the lowest error rate with KNN having a higher area ratio than other models. However, on the validation data, Artificial Neural Networks achieved the best performance with lowest area ratio and relatively lower error rate. The above-mentioned research paper used scatter plot diagrams, regression line and R square to estimate the real default probability. Of the above methods, only Artificial Neural Networks had the highest explanatory ability in terms of R square as well as regression line.

In this we try to study 4 data mining techniques: KNN, Naïve Bayes, Logistic Regression and Support Vector Mechanism (SVM) and check their performance through cross validations, ROC, and confusion metrics. Following section explains the models and methods followed by the comparative analysis of the experimental results.

4 Methods and Techniques

4.1 Dataset Description

After an extensive search for datasets, we selected Prudential Life Insurance dataset from Kaggle, which is a platform for analytics competitions.

The data set is pre-separated among training and test sample in the ratio of 3:1 and have a random sampling being done. The train data set is a transactional data consists of 59,381 customers and 128 variables as predictors. This set was further divided in 70:30 ratio as training and validation set. The test dataset contains the same variables for another set of 19,765 customers. The data set comprises of nominal, continuous, as well as discrete variables, which are anonymized. Table 1 describes the variables present in the data set.

4.2 Data pre-processing

Data pre-processing, also identified as the data cleaning process, implicates that the outliers or noisy data are removed from the target dataset. This stage also incorporates the development of any strategies required to deal with the inconsistencies in the target data. This process aims at reducing the data size, find the relations between the data points, normalize the data, remove outliers and impute the missing values with the appropriate values like median in the dataset.

Table 1 Data set description

Attributes	Type	Description
Product_Info_1-7	Categorical	7 normalized attributes concerning the product applied for
Ins_Age	Numeric	Normalized age of an applicant
Ht	Numeric	Normalized height of an applicant
Wt	Numeric	Normalized weight of an applicant
BMI	Numeric	Normalized Body Mass Index of an applicant
Employment_Info_1-6	Numeric	6 normalized attributes concerning employment history of an applicant
InsuredInfo_1-6	Numeric	6 normalized attributes offering information about an applicant
Insurance_History_1-9	Numeric	9 normalized attributes relating to the insurance history of an applicant
Family_Hist_1-5	Numeric	5 normalized attributes related to an applicant's family history
Medical_History_1-41	Numeric	41 normalized variables providing information on an applicant's medical history
Medical_Keyword_1-48	Numeric	48 dummy variables relating to the presence or absence of a medical keyword associated with the application
Response	Categorical	Target variable, which is an ordinal measure of risk level, having 8 levels.

4.3 Data exploration

Before even contemplating the relationship between dependent and independent variables, it is first crucial to become familiar with the contents of the modeling data by exploring the distributional properties of each variable. Descriptive statistics such as min, max, mean, median, mode, and frequency provide useful understanding. This process tells modelers what they have to work with and informs them of any data issues they must address before proceeding.

After the initial distributional analysis, the univariate (one variable at a time) review is extended to examine relationship with the target variable. One-by-one, the correlation between predictive and target variable is calculated to preview of each variable's predictive power. The variables that stand out in this process will be highly correlated with the target, well populated, sufficiently distributed, and thus are strong candidates to include in the final model. We have used R Studio for data preprocessing, exploration, and modeling.

4.4 Variable Generation

Variable generation is the process of creating variables from the raw data. Every field of data loaded into the system, including the target and predictive variables, is assigned a name and a data format. At times this is a trivial process of mapping one input data field to one variable with a descriptive variable name. However, this step can require more thought to build the most effective predictive models. Individual data fields can be combined in ways that communicate more information than the fields do on their own. These synthetic variables, as they are called, vary greatly in complexity. Simple example includes combining height and weight to calculate BMI.

4.5 Dimensionality Reduction

The dimensionality reduction involves reducing the number of features used for efficient modelling. It can be categorized into feature selection and feature extraction. Feature selection involves selecting important features for modelling, whereas the feature extraction is a process that transforms high dimensional data into fewer dimensions. Therefore, dimensionality reduction can be used to train the models faster and increase their accuracies by reducing over-fitting.

Feature extraction derives new features from the original features in order to remove redundant and irrelevant attributes (we use attributes and features interchangeably).

This project uses two methods, namely the correlation - based feature selection method and principal component analysis - based feature extraction method. These methods have been discussed in the subsections below-

4.5.1 Correlation-based feature selection:

Correlation-based feature selection (CFS) evaluates subsets of features such that they are highly correlated with the class but uncorrelated to each other. This method is easy and fast to implement. It also removes noisy data which improves the performance of the model. This method generates the optimal number of features by itself and does not require the analyst to specify the number of features.

4.5.2 Principal components analysis feature extraction:

Principal components analysis (PCA) is an unsupervised linear feature extraction technique designed to reduce the size of the data by extracting information into new features known as Principal Components. These principal components are treated as new attributes and used to develop models. Generally, the principal components have better explaining power than the individual attributes. The variance ratio gives a measure of how much information is retained in the principal components.

4.6 Supervised Learning Algorithm

This section will elaborate on the different algorithms implemented on the data set to build the predictive models. The techniques namely, K-Nearest Neighbors, Naïve Bayes, Logistic Regression and Support vector machine.

a. Naïve Bayes – Baseline Model

Naïve Bayes model is a straightforward probabilistic prediction model. They are a set of extremely fast and simple supervised classification algorithms based on applying Bayes' theorem. This model is often suitable for very high-dimensional datasets. Since they are extremely fast and have very few tuning parameters, they become useful as baseline for classification problems. Bayes' theorem states the following relationship, given class variable 'y' and dependent feature vector x1 through xn

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)P(x_1, \ldots x_n \mid y)}{P(x_1, \ldots, x_n)}$$

Using the naive conditional independence assumption that

$$P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$$

for all i, this relationship is simplified to

$$P(y \mid x_1, \ldots, x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

Since P(x1,…,xn) is constant given the input, we can us the following classification rule:

$$\hat{y} = \arg\max_y P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

b. K-Nearest Neighbors (KNN) Model

KNN is a non-parametric and a lazy algorithm. It uses the dataset to predict and classify the new sample points. It is considered non-parametric because it does not make any assumption about the underlying data. This is especially useful is classifying the "real world" data as most of the data does not follow typical theoretical assumptions. The method of k-nearest neighbors makes very mild structural assumptions: its predictions are often accurate but can be unstable.

Nearest-neighbor methods use those observations in the training set which is the closest in input space to x to form ˆ Y. Specifically, the k-nearest neighbor fit for ˆ Y is defined as follows:

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i,$$

KNN performs classification of the data points (where output is a class membership) by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

c. Logistic Regression Model

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

The cost function represents optimization objective i.e. we create a cost function and minimize it so that we can develop an accurate model with minimum error.

$$J(\theta) = -\frac{1}{m} \sum \left[y^{(i)} \log(h\theta(x(i))) + \left(1 - y^{(i)}\right) \log(1 - h\theta(x(i))) \right]$$

Ridge regression uses L2 regularization which adds the following penalty term to the OLS equation.

$$+\lambda \sum_{j=0}^{p} w_j^0$$

Lasso regression uses the L1 penalty term and stands for **Least Absolute Shrinkage and Selection Operator**.

$$+\lambda \sum_{j=0}^{p} |w_j|$$

Elastic Net - A third commonly used model of regression is the Elastic Net which incorporates penalties from both L1 and L2 regularization:

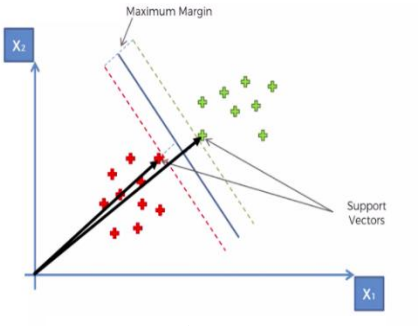
$$\frac{\sum_{i=1}^{n} (y_i - x_i^J \hat{\beta})^2}{2n} + \lambda \left(\frac{1 - \alpha}{2} \sum_{j=1}^{m} \hat{\beta}_j^2 + \alpha \sum_{j=1}^{m} |\hat{\beta}_j| \right)$$

Elastic net regularization - In addition to setting and choosing a lambda value elastic net also allows us to tune the alpha parameter where $\boldsymbol{\alpha} = 0$ corresponds to ridge and $\boldsymbol{\alpha} = 1$ to lasso. Simply put, if you plug in 0 for alpha, the penalty function reduces to the L1 (ridge) term and if we set alpha to 1 we get the L2 (lasso) term. Therefore, we can choose an alpha value between 0 and 1 to optimize the elastic net. Effectively this will shrink some coefficients and set some to 0 for sparse selection.

d. Support Vector Machine (SVM) Model

The main objective of support vector machines is to determine an optimal separating hyperplane, which correctly classifies the data points of different classes. The dimensionality of the hyperplane is equal to the (number of input features -1). One can chose many possible separating hyperplanes, but the objective is to find a plane that has the maximum margin. The data points closest to the separating hyperplanes are the support vectors.

The input to SVM is a set of (input, output) training pair samples. The features are denoted as x1, x2…xn, and the output result y. There can be high number of input features xi. The output is given as a set of weights wi for each feature, whose combination predicts the value of y.



The optimization function for SVM is as given below –

$$\min_{\gamma,w,b} \quad \frac{1}{2}||w||^2$$
$$\text{s.t.} \quad y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m$$

The given optimization problem has concave quadratic objective function and only linear constraints. We can solve this problem by Lagrange duality.

$$L = \frac{1}{2}\|\vec{w}\|^2 - \sum_i^n \alpha_i \left[y_i(\vec{w}\cdot\vec{x} + b) - 1\right]$$

$$\frac{\partial L}{\partial w} = \vec{w} - \sum_i^n \alpha_i y_i x_i = 0$$

$$\boxed{\vec{w} = \sum_i^n \alpha_i y_i x_i}$$

$$\frac{\partial L}{\partial b} = -\sum_i^n \alpha_i y_i = 0$$

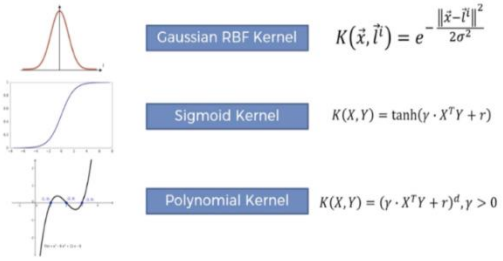
$$\boxed{\sum_i^n \alpha_i y_i = 0}$$

$$L = \sum_i^n \alpha_i - \frac{1}{2}\sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

The optimization of maximizing the margin is done to reduce the weights to a few, which correspond to the important features. These features are important in deciding the hyperplane and they correspond to the support vectors because they "support" the hyperplane.

There are two types of classification SVM algorithms - *Hard Margin*, which finds the separating hyperplane without any tolerance to any form of misclassification and *Soft Margin*, which allows misclassification of data points and permits the functional margin to be less that 1 (1-epsilon).

Kernels: Kernels are used to solve non-linear problems by using a linear classifier. It takes a low dimensional input space and transforms it into a higher-dimensional space. A kernel function is applied to each data point to map the non-linear observation to higher dimension where it can be separated linearly. Types of Kernels are listed below:



<u>*Gaussian Radial Basis Function*</u> – This kernel is a general-purpose kernel and is used when there is no prior knowledge about the data. Depending on σ, this kernel can either provide a good fit or an overfit. If the values of σ is larger than distance between the classes, it can give an overly flat discriminant surface. On contrary, if σ is smaller, this will over-fit the samples. A good value for σ will be comparable to the distance between the closest members of the two classes.

<u>*Polynomial*</u>- The Polynomial kernel is defined by equation shown below. Here d is the order of the kernel and 'r' is a constant that allows to trade off the influence of the higher order and lower order terms. Higher order kernels tend to overfit the training data and do not generalize well.

<u>*Linear*</u> - This is used when the data is Linearly separable, that is, it can be separated using a single line. It is one of the most common kernels to be used. It is mostly used when there are many Features in the dataset. Training a linear SVM model is generally faster than any other kernel.

$$k(x, y) = x^T y + c$$

<u>*Sigmoid*</u> - This kernel uses tanh function and it is generally used as a proxy for neural networks. Alpha and constant c(intercept) are the two adjustable parameters in sigmoid. A common value for alpha is 1/N, where N is the dimension of the dataset.

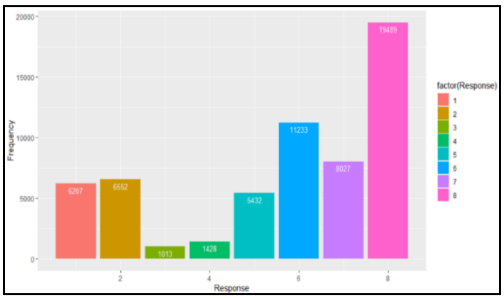
Cost C (Regularization): C is a penalty parameter and it represents how much misclassification is bearable. Through C, one can control the trade-off between decision boundary and misclassification.

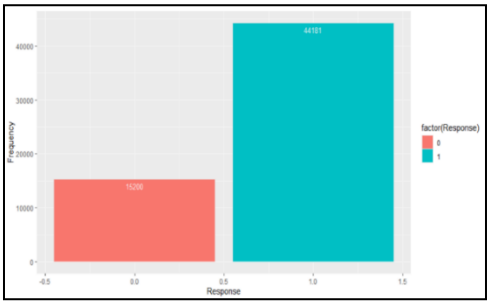
Gamma: Gamma defines how far the influence of single training example reaches. This value in set for different Kernels - 'rbf', 'poly' and 'sigmoid'.

5 Modelling Process

5.1 Data Preprocessing and Exploration

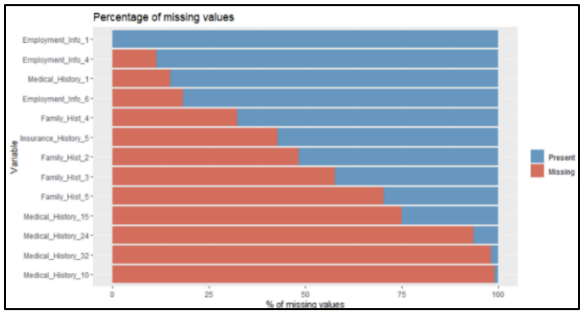
Initially our training data set consists of Response variable ranging from 1-8. This range helps the firm in identifying the chances of insurance claim by the customer. We have converted it to a binomial variable. The frequency distribution plot of the Response variable is shown below:





We converted the multinomial response variables into binary (0,1). 0 represents "Low Risk Applicant" and 1 represents "High Risk Applicant". Preference should be given to 1.

5.2 Missing Value Treatment



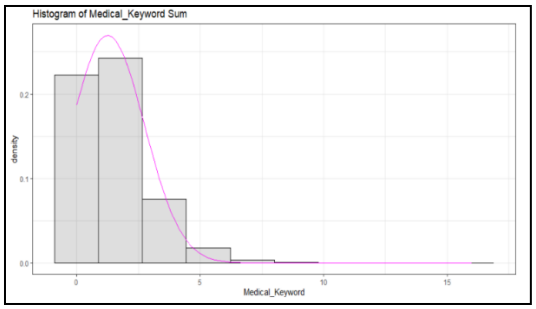
The missing value graph depicts that there 13 variables which have missing values in them. The variables with more than 70% missing values are removed during model development.

The missing values for other variables have been replaced by their median values after analyzing their normal distribution curves.

5.3 Feature Engineering

There are two sets of variables – Medical Keyword and Medical History which mostly consists of zero values. We have experimented with these set of variables and introduced below mentioned new variables:

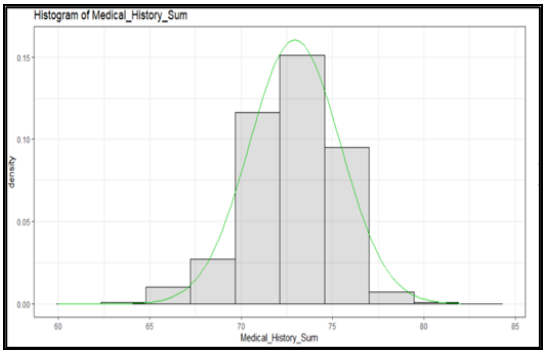
a. MEDICAL_KEYWORD



There are 48 'Medical_Keyword_' variables in the dataset. These variables mostly consist of 0's and 1's. Individually, these variables have very less predictive power. Hence, they are combined, by taking the sum, to form a single variable 'Medical_Keyword'.

The graph on the left shows that the derived variable in not normally distributed. The maximum number of values fall in the range of 0 to 5.

b. MEDICAL_HISTORY_SUM – Medical history is spread across 41 different variables. We combine variables from 3 to 41 to form a single variable Medical History Sum. Initial 41 variables were then deleted from data.



The graph on the left shows that the derived variable in normally distributed. The graph on the right shows that Medical_History_Sum distribution for each response variable (0 and 1) and conclude that the new variable has good predictive power.

This method of summing has led to large variable reduction with all the information intact. More the value of new derived variables, more should be chances of risk as it refers to severe medical conditions in the past and in present.

5.4 Data Exploration for Continuous Variables

In order to understand the relation between continuous independent variables i.e. Height, Weight, BMI, Product Info2 and Age; and response variables, we have used box plots and correlation table analyses the relationship. These variables are treated independently and examined closely with that of the target variable Response. We have then inferred from the correlation plot that BMI and Wt are highly correlated

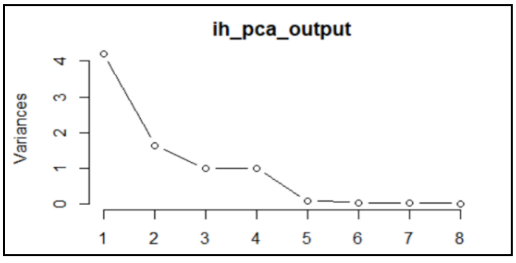
5.5 Principal Component Analysis – Dimensionality Reduction

We performed Principal Component Analysis on different groups of variables (as mentioned in the table below) in order to reduce the dimension of the feature space. We found that PCA was most effective for 'Insurance_History' variables as its 4 principal components retained the maximum information of the 9 variables. Hence, the 4 principal components are added to the dataset and original 9 Insurance History variables are removed.

Variables	PCA Output Analysis
Product_Info (1-7)	By selecting 6 Principal Components(PC) out of 7 PC , we can only preserve 89.46% of the total variance of the Employment_Info data
Employment_Info (1-6)	By selecting 5 Principal Components(PC) out of 6 PC , we can preserve 96.10% of the total variance of the Employment_Info data
InsuredInfo (1-7)	By selecting 6 Principal Components out of 7 PC , we can preserve 93.02% of the total variance of the Insured_Info data
Insurance_History (1-9)	By selecting 4 Principal Components out of 8 PC, we can preserve 98.06% of the total variance of the Insured_History data

PCA for Insurance History

```
Importance of components:
                          PC1    PC2    PC3
Standard deviation     2.0517 1.2817 0.9996
Proportion of Variance 0.5262 0.2053 0.1249
Cumulative Proportion  0.5262 0.7315 0.8564
                          PC4     PC5     PC6
Standard deviation     0.9966 0.27816 0.18710
Proportion of Variance 0.1241 0.00967 0.00438
Cumulative Proportion  0.9806 0.99027 0.99464
                          PC7    PC8
Standard deviation     0.17334 0.1132
Proportion of Variance 0.00376 0.0016
Cumulative Proportion  0.99840 1.0000
```



5.6 Correlation among other variables

The variables which had correlation value more than or equal to 0.7 were removed (as shown in the output image). Also, variable 'Id' is removed as it does not add to prediction power of the model. The total number of variable remaining is 32.

```
All correlations <= 0.7
[1] "Wt"                "InsuredInfo_6"
[3] "Employment_Info_3" "Employment_Info_5"
```

5.7 Random Forest for Variable Importance

After data exploration and feature engineering, we run Random Forest to determine the top 25 important variables (from Variable Importance). This is performed in order to get only those variables that play significant role in our modelling. As PCA and summation method cannot be applied to all buckets of variables, this was an important step to recognize essential variables from family history, insured info, employment info and product info.

6 Experiments and Results

The data from Prudential Life Insurance was used to establish Naïve Bayes Classifier model. This final data consists of 26 columns and 59,381 lines. Each column, in succession, indicates demographics, medical history, product info, medical keyword and insurance history and response are customers with high risk or not, where 1 indicates that person has high risk and should apply for life insurance and 0 indicates that the person is at low risk level.

6.1 Naïve Bayes

In our dataset, we define *X* as predictor those are data from column 1 up to column 25 i.e. all attributes related to applicants, and we define *y* as independent variable that is data from column 26 i.e. Response variable. Then we split the data into training data and testing data. After that we input the data into the Naïve Bayes Classifier model and make prediction and with confusion matrix, we get the accuracy of the prediction.

```
> confusionMatrix(Predict_val, pdVal$Response)
Confusion Matrix and Statistics

          Reference
Prediction  Low Risk High Risk
  Low Risk      1014       754
  High Risk     3596     12450

               Accuracy : 0.7558
                 95% CI : (0.7494, 0.7621)
    No Information Rate : 0.7412
    P-Value [Acc > NIR] : 3.988e-06

                  Kappa : 0.2037

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.21996
            Specificity : 0.94290
         Pos Pred Value : 0.57353
         Neg Pred Value : 0.77589
             Prevalence : 0.25879
         Detection Rate : 0.05692
   Detection Prevalence : 0.09925
      Balanced Accuracy : 0.58143

       'Positive' Class : Low Risk
```

By applying the algorithm Naïve Bayes discussed above and using R studio we can produce good accuracy. Starting with the baseline model and no Laplace smoothening and setting Kernel as False, we got the 75.17% accuracy. The performance of the Naïve Bayes classification system can be searched using confusion matrix.

After grid search and several trials and we have tried to achieve that the best accuracy is 75.58% with 70% of the training data. Out of the 75% actual instances of high risk, the classifier predicted that 12450 of them positive with high risk and 754 negatives with low risk. Out of the 25% actual instances of low risk, the classifier predicted that 3596 of them negative with low risk and 1014 positive with low risk. The data has a good specificity but less sensitivity.

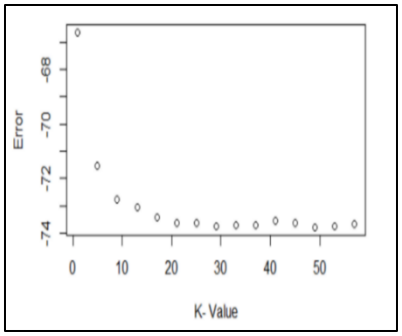
This best accuracy was attained with Kernel = false, Laplace smoothening factor = 2 and adjust = 3.

6.2 K-Nearest Neighbors (KNN)

This is a further improvement over Naïve Bayes as it returns majority class of the k-nearest neighbors. Here we developed models with different values of K and found that the accuracy is maximum at K = 29 as seen from the below output. The first image shows accuracy at various values of K and second image depicts that error is least at K = 29.

Results from KNN grid search for different K values is shown on the right side.

```
1 = 66.75087
5 = 72.763
9 = 73.95868
13 = 74.50881
17 = 74.72213
21 = 74.76704
25 = 74.98597
29 = 75.12069
33 = 75.04772
37 = 75.05333
41 = 75.0421
45 = 74.97474
49 = 74.96913
53 = 74.96913
57 = 74.92983
```



6.3 Logistic Regression

```
          Reference
Prediction  Low Risk High Risk
  Low Risk      1910      1551
  High Risk     2700     11653

               Accuracy : 0.7614
                 95% CI : (0.755, 0.7676)
    No Information Rate : 0.7412
    P-Value [Acc > NIR] : 3.095e-10

                  Kappa : 0.3231

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.4143
            Specificity : 0.8825
         Pos Pred Value : 0.5519
         Neg Pred Value : 0.8119
             Prevalence : 0.2588
         Detection Rate : 0.1072
   Detection Prevalence : 0.1943
      Balanced Accuracy : 0.6484

       'Positive' Class : Low Risk
```

Logistic regression has further improved the results in terms of sensitivity, but the training accuracy tends to be in the same bracket. We have done various regularization methods i.e. L1, L2 and elastic-net found that Ridge is performing better among the three.

After parameter tuning via grid search and several trials, we have tried to achieve that the best accuracy is 76.14% with 70% of the training data. Out of the 75% actual instances of high risk, the classifier predicted that 11653 of them positive with high risk and 1551 negatives with low risk.

Out of the 25% actual instances of low risk, the classifier predicted that 2700 of them negative with low risk and 1910 positive with low risk. The data has a good specificity and better sensitivity.

The confusion matrix with L1 regularization can be seen on left.

6.4 Support Vector Machine (SVM)

To find the optimal hyper-parameters for SVM models, we create a grid of hyper-parameters and generate models for all possible combinations. The grid search is performed on hyper parameters like – Kernel, Cost, Gamma and Degree. Few results from the grid search have been tabulated below-

SVM - Grid Search Result						
Hyper Parameters				Best Model		
Kernel	Cost	Gamma	Accuracy	Test Accuracy	Sensitivity	Precision
RBF	0.001	0.0001	0.74284	0.7782	0.95	0.79
		1	0.74284			
	10000	1.00E-05	0.74688			
		1.00E-06	0.74684			
	50000	1.00E-05	0.75475			
		1.00E-06	0.74686			
	1000000	1.00E-05	0.77227			
		1.00E-07	0.77772			
Sigmoid	10	1.00E-05	0.74308	0.7511	0.9754	0.7595
	10000	100	0.65657			
	100000	0.1	0.67721			
	10000	1.00E-05	0.74681			
Linear	0.025	1.00E-05	0.74681	0.7511	0.98	0.76
	1.00E-05	1.00E-05	0.74284			
Kernel	Cost	Degree	Accuracy	Test Accuracy	Sensitivity	Precision
Poly	1	4	0.7537	0.7694	0.97	0.78
	0.025	4	0.7527			
	1	5	0.7451			
	0.025	5	0.6824			

Best Model Analysis: For Kernel RBF, we have experimented with various combinations of Cost(C) (ranging from 0.001 to 1000000) and Gamma (ranging from 1.00E-10 to 1) values.

Grid search is performed on 2 cross-fold validation. Form the test accuracy, sensitivity, and precision values we can see that the best model is with parameters – Cost – 10^6, Gamma = 1.00E-05.

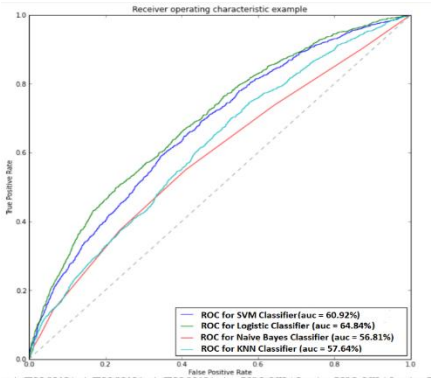
Since the data has extremely high dimensionality, the performance of linear kernel is less than the baseline classifier. Poly and Sigmoid kernels report accuracies in the range of 75-76%. Sensitivity reported by their best models is as high as 97% which is slightly higher than the sensitivity from RBF. Taking into consideration all the parameters, SVM model with RBF kernel is most optimum.

Confusion Matrix: The confusion matrix tells us that 12586 applicants in the validation data are correctly categorized as the High-Risk applicants and 1223 applicants are Low Risk applicants with a sensitivity of 0.95 and precision of 0.78, respectively. The number of support vectors used to create the decision boundary is 13304 for Low Risk class and 4511 for the high risk. Hence, we can clearly see that the sensitivity has increased with a great amount from our baseline model to our final best model.

```
Prediction  Low Risk High Risk
  Low Risk      1223       618
  High Risk     3387     12586

               Accuracy : 0.7752
                 95% CI : (0.769, 0.7813)
    No Information Rate : 0.7412
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2716

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9532
```

6.5 Computational results

Following the dimensionality reduction, the reduced data set was exported and used for building the prediction models using machine learning algorithms discussed in the previous section. Model validation has been performed using a k-folds (tenfold) cross-validation. Below four models were developed, and accuracy, sensitivity and precision measures along with ROC are shown in table below.



AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represent degree or measure of separability. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. As we can see both Logistic and SVM have AUC > 60% in the ROC curves shown.

After all the analysis, we chose SVM kernel model to be best for our data with good accuracy, good ROC, higher precision and best sensitivity of 95%.

Best Models - Method	Accuracy	Sensitivity	Precision	ROC
Naïve Bayes with Laplace as 2	75.58%	21.99%	57.35%	56.81%
KNN with K=29	75.12%	58.86%	12.01%	57.64%
Logistic – Ridge L1 regularization	76.14%	41.43%	55.19%	64.84%
SVM – Radial kernel	77.52%	95.32%	78.80%	60.92%

7 Future Scope

Future work relates to the more in-depth analysis of the problem and new methods to deal with specific mechanisms. Customer segmentation is the division of the data set into groups with similar attributes can be implemented to segment the applicants into groups with similar characteristics based on the attributes present in the dataset.

This can be further investigated by performing unsupervised methods like Clustering. For example, similar employment history, insurance history, and medical history can be grouped under one cluster. Following the grouping of the applicants, predictive models can be implemented to contribute to a different data mining approach for the life insurance customer data set.

8 Conclusion

In this paper, Prudential Life Insurance problem was investigated using several well-known classification models in machine learning. Due to non-linearity nature of the dataset, nonlinear classifiers outperform linear models. All models including KNN, Naïve Bayes, Logistic Regression and SVM provided around 76 percent prediction accuracy for this problem.

However, by considering sensitivity and ROC, SVM indicated the highest performance with respect to other models. Moreover, the effect of hyperparameters on the prediction accuracy of different models was investigated and lastly, dimensionality reduction is applied as a preprocessing step to these models. By investigating the prediction accuracy of different models after the PCA, it turned out PCA may improve the performance of all models.

Hence, for this dataset, SVM radial kernel with dimensionality reduction resulted in the best results in terms of accuracy, ROC and sensitivity.

References

Yeh, I. C., & Lien, C. H. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications, 36(2), 2473-2480.

Y. Lin, Y. Lee, and G. Wahba. Support vector machines for classification in non-standard situations. Machine Learning, 46:191–202, 2002.

K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In Proceedings of ICML, 1999.

Sivarajah U, Kamal M, Irani Z, Weerakkody V (2017) Critical analysis of big data challenges and analytical methods. J Bus Res 70:263–286

Joly Y, Burton H, Irani Z, Knoppers B, Feze I, Dent T, Pashayan N, Chowdhury S, Foulkes W, Hall A, Hamet P, Kirwan N, Macdonald A, Simard J, Hoyweghen I (2014) Life Insurance: genomicsStrat- ification and risk classification. Eur J Hum Genet 22:575–579

Sharifzadeh S, Ghodsi A, Clemmensen L, Ersbøll B (2017) Sparse supervised principal component analysis (SSPCA) for dimension reduction and variable selection. Eng Appl Artif Intell 65:168–177

Taguchi Y, Iwadate M, Umeyama H (2015) Principal component analysis-based unsupervised feature extraction applied to in silico drug discovery for posttraumatic stress disorder-mediated heart dis- ease. BMC Bioinf 16:1–26

Fan C, Wang W (2017) A comparison of underwriting decision making between telematics-enabled UBI and traditional auto insur- ance. Adv Manag Appl Econ 7:17–30

Goleiji L, Tarokh M (2015) Identification of influential features and fraud detection in the Insurance Industry using the data mining techniques (Case study: automobile's body insurance). Majlesi J Multimed Process 4:1–5

Joudaki H, Rashidian A, Minaei-Bidgoli B, Mahmoodi M, Geraili B, Nasiri M, Arab M (2016) Improving fraud and abuse detection in general physician claims: a data mining study. Int J Health Policy Manag 5:165–172

Nian K, Zhang H, Tayal A, Coleman T, Li Y (2016) Auto insurance fraud detection using unsupervised spectral ranking for anomaly. J Fin Data Sci 2:58–75

Mishr K (2016) Fundamentals of life insurance theories and applications. In: 2nd ed, Delhi: PHI Learning Pvt Ltd,

Mamun DMZ, Ali K, Bhuiyan P, Khan S, Hossain S, Ibrahim M, Huda K (2016) Problems and prospects of insurance business in Bangladesh from the companies' perspective. Insur J Bangladesh Insurance Acad 62:5–164

Harri T, Yelowitz A (2014) Is there adverse selection in the life insurance market? Evidence from a representative sample of pur- chasers. Econ Lett 124:520–522

Hedengren D, Stratmann T (2016) Is there adverse selection in life insurance markets? Econ Inq 54:450–463

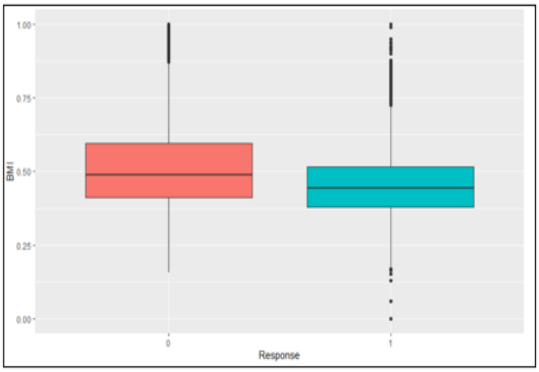
The Kaggle Website. [Online]. https://www.kaggle.com/c/prudential-life-insurance-assessment/data

Appendix:

Data Exploration for Continuous Variables:

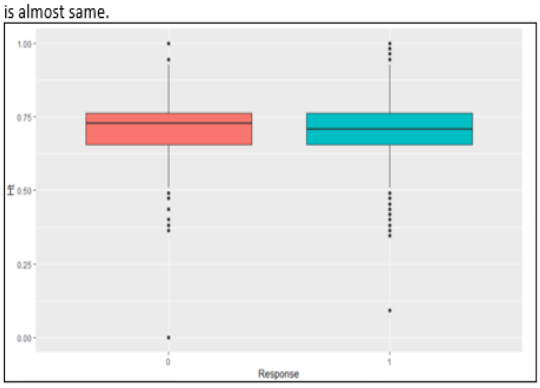
BMI vs Response

BMI has fair predictive power as the medians for the response variable 0 and 1 are slightly varying.



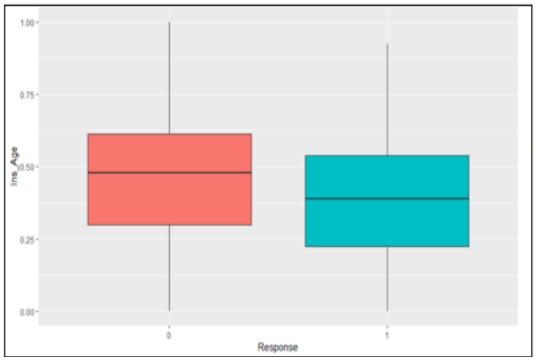
Ht vs Response

Ht has very less prediction power. The medians of Ht for the response variable is almost same.



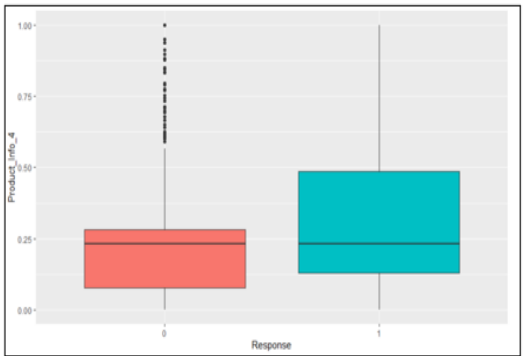
Ins_Age vs Response

Ins_Age has fair predictive power.



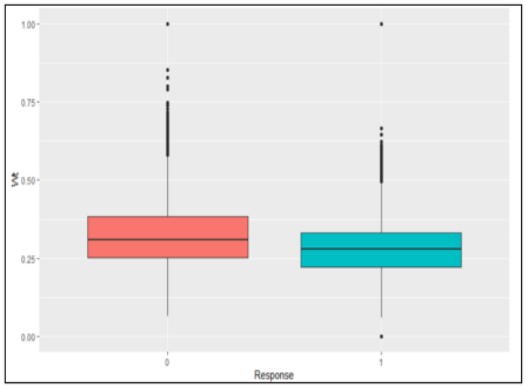
Product_Info_4 vs Response

Product_Info_4 has very less predictive power.



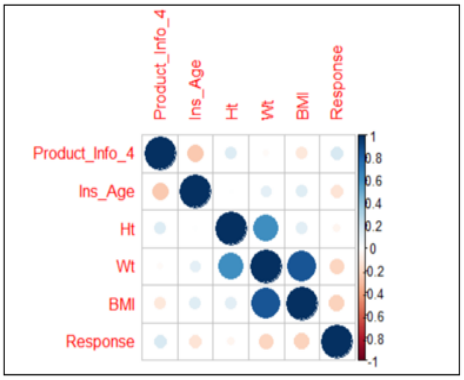
Wt vs Response

Wt has less predictive power.



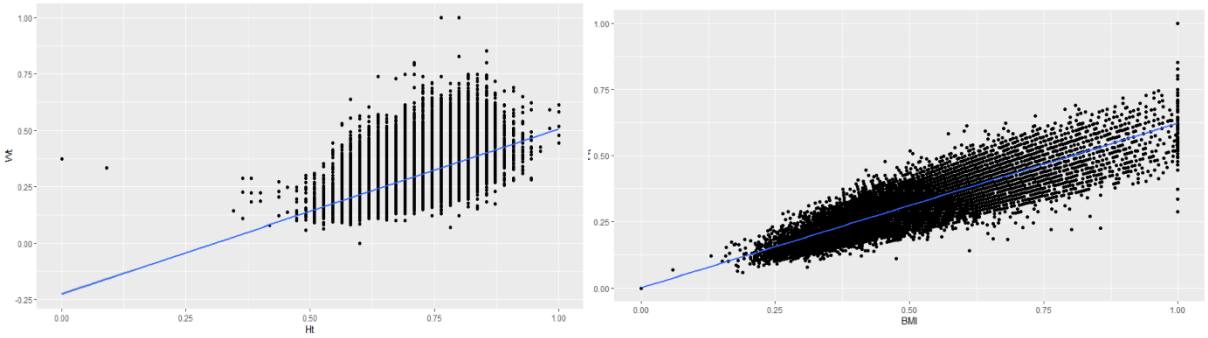
Correlation Between Important Continuous Variables

BMI and Wt are highly correlated (<value>).

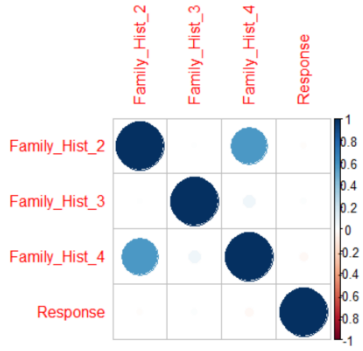


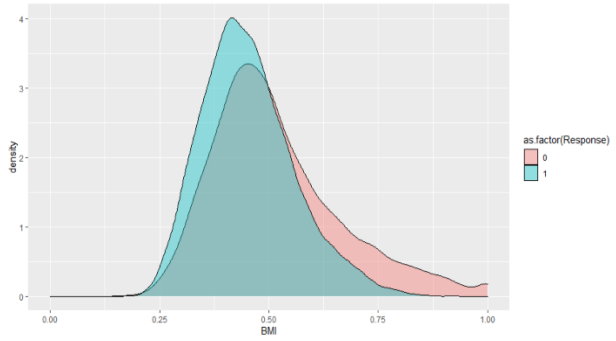
Correlation and Density plots:

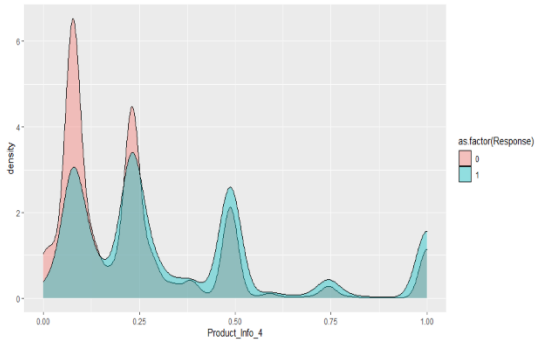
1. Weight, height and BMI

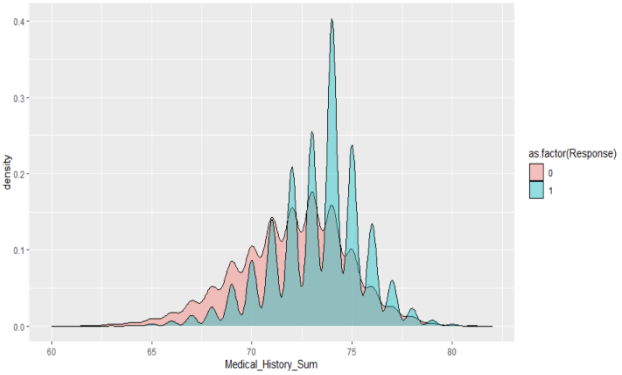


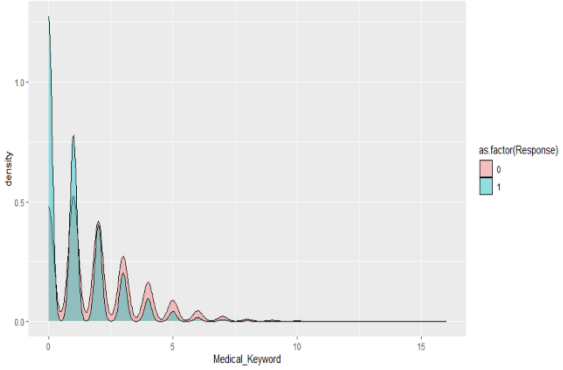
2. Correlation plots between Response and various variable buckets:











3. Grid Search Results for SVM with RBF, Sigmoid and Poly Kernels:

SVM - Grid Search Result						
Hyper Parameters				Best Model		
Kernel	**Cost**	**Gamma**	**Accuracy**	**Test Accuracy**	**Sensitivity**	**Precision**
RBF	0.001	0.0001	0.74284271	0.778164468	0.95	0.79
		1	0.74284271			
		2	0.74284271			
	100	0.0001	0.74688447			
		1	0.74139922			
		2	0.74284271			
	1000	0.0001	0.75366886			
		1	0.74142328			
		2	0.7428427			
	10000	1.00E-05	0.74688447			
		1.00E-06	0.74683636			
	50000	1.00E-05	0.75475148			
		1.00E-06	0.74686041			
	100000	1.00E-06	0.75256219			
		1.00E-07	0.75253813			
		1.00E-10	0.74284271			
	200000	1.00E-06	0.76411009			
		1.00E-07	0.76326806			
		1.00E-10	0.74284271			
	1000000	1.00E-05	0.77226579			
		1.00E-07	0.7777154			
Sigmoid	10	1.00E-05	0.74308329	0.751052484	0.9754	0.7595
	1000	1.00E-05	0.7468123			
	10000	1.00E-05	0.7468123			
	100000	0.1	0.67721214			
	200000	0.1	0.67694751			
	10000	1.00E-05	0.7468123			
	10000	100	0.65657027			
Kernel	**Cost**	**Degree**	**Accuracy**	**Test Accuracy**	**Sensitivity**	**Precision**
Poly	10	4	0.74604244	0.769407802	0.97	0.78
	1000	4	0.74471924			
	0.025	5	0.74661983			
	0.00025	5	0.7604292			
	1	4	0.75366886			
	0.025	4	0.7527306			
	0.00025	4	0.75044508			
	1	5	0.74505606			
	10	5	0.68238464			
	1000	5	0.68202377			