Research problem

Financial institutions are lending money and there is always risk that customer will not pay back. Purpose of this research is to explore given customer data using statistical techniques, find most relevant columns and predict if customer will pay back bank loan. We will also compare how much prior probability impacts results.

Data

Source of data is the dataset from Kaggle. It contains background information about customers who got a loan. Original dataset contains over 100514 rows of data and columns are described in Table 1.

Table 1 Data description

Column name	**Column description**	**Missing variables**
Loan ID	GUID	514
Customer ID	GUID	514
Loan Status	Target class that will be predicted. Fully paid or Charged off	514
Current Loan Amount	Numerical	514
Term	Categorical. Short term or long term	514
Credit Score	Numerical	19668
Annual Income	Numerical	19668
Years in Current Job	Categorical. 1 Year, 2 Years, 3 Years etc	514
Home Ownership	Categorical. Home Mortgage, Own Home or Rent	514
Purpose	Categorical. Home Improvements, Debt Consolidation etc	514
Monthly Dept	Numerical	514
Years of Credit History	Numerical	514
Months since last delinquent	Numerical	53655
Number of Open Accounts	Numerical	514
Number of Credit Problems	Numerical	514
Current Credit Balance	Numerical	514
Maximum Open Credit	Numerical	516
Bankruptcies	Numerical	718
Tax Liens	Numerical	524

Cleaning data

For classification Loan ID and Customer ID are not necessary, so they were removed. Months since last delinquent had a lot of missing values, removed column instead of replacing with mean values, because it would have significant impact on column distribution. Removed rows with missing values. Removed duplicated rows. Removed 99999999 numbers from current loan amount because it looked wrong.

Categorical columns replaced with integers, so they could be added to training. Columns were not normalized, because when playing with data, it had no significant impact on results.

Prior bias

Table 2 shows prior probabilities before and after removing bias. Since we will try to predict loan status, then considering bias of this column is important. Prior probability was modified by removing excessive rows.

Table 2 Prior probabilities

Loan status	Original prior probability	Modified prior probability
Paid	71.3%	50%
Charged off	28.7%	50%
Rows count	56461	32428

Correlation plot

Figure 1 describes correlation matrix between columns. This matrix tells us that Bankruptcies, Credit problems and Tax liens are well correlated, which makes sense and they could be used to determine problematic customers. Also, strong correlation between Credit score, Loan amount and Term, which also makes sense and those columns could help to identify good customers. Out of correlated columns, we could leave just one of each column for training if we were to optimize for efficiency, but in current work decided to leave them in.

There are a lot of columns with weak or no correlation at all, which is good, since it means that there might be opportunity to learn something from data.

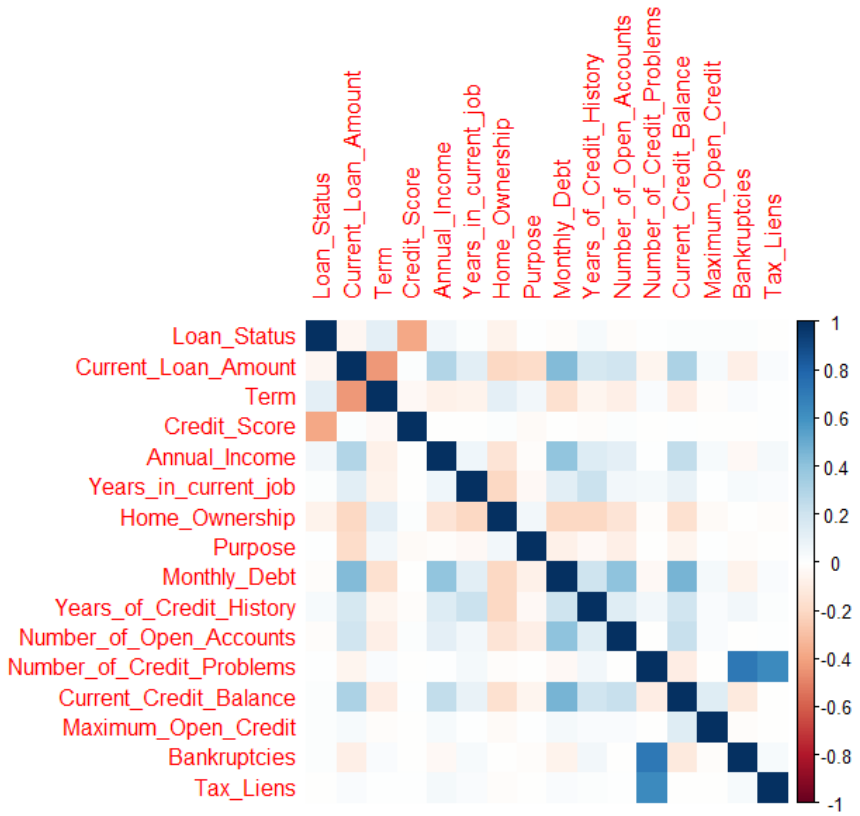

Figure 1 Correlation matrix

Principal Component Analysis (PCA)

Figure 2 describes importance of data using principal component analysis. We can see that relationship is almost linear, which means all columns might be important. We can see that 12 principal components explain almost 90% of variance.

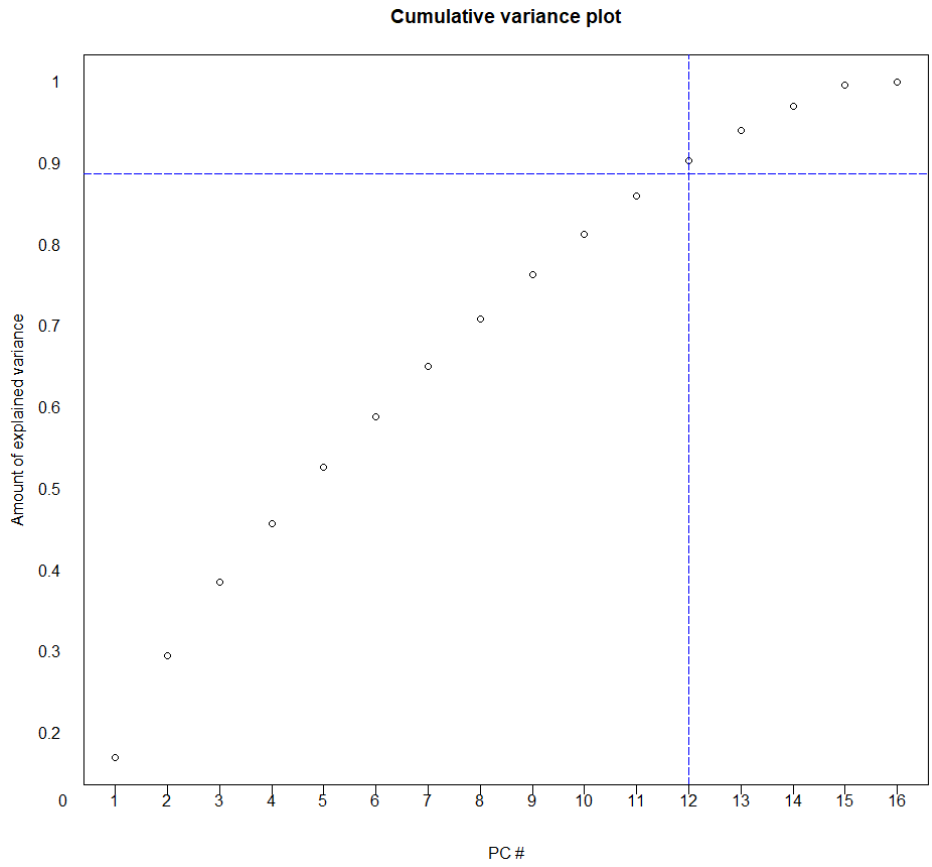

Figure 2 PCA

Decision tree

Table 3 describes feature selection using decision tree. It appears that Credit score has biggest impact. Total of 5 features are significant and other features can be ignored. For training purpose, we keep all features in. When we remove prior bias, then we see that credit score becomes less important, but still dominant factor. Features in table 3 will have most attention when training models.

Table 3 Feature selection according to tree splitting

Feature	**Importance with original prior probability**	**Importance with modified prior probability**
Credit Score	86.82%	69.36%
Term	7.55%	12.08%
Annual Income	3.02%	10.58%
Home Ownership	1.37%	4.15%
Current Loan Amount	1.23%	3.16%

Methodology

Linear Discriminant Analysis (LDA)

Dataset was split into training and testing sets, 75% for training and 25% for testing. K-fold cross-validation was used to calculate accuracy of model.

Chosen classification algorithm is LDA. Table 4 describes results trained using LDA.

Table 4 LDA results

Description	Original prior probability	Modified prior probability
All columns	78.90%	64.56%
All columns except Credit score	71.44%	56.98%
Only credit score column	78.88%	63.29%

Credit score has big impact on accuracy, and it is most important component when deciding to give a loan. Also, it is important to modify prior probabilities before classification, because otherwise result is unreliable. We should also consider that LDA does not work well with categorical data, because numbers are not continuous.

K-means

Chosen clustering algorithm is K-means. For K-means dataset was not split into train/test. Accuracy is calculated by comparing assigned classes from K-means to loan status classes of original dataset. Goal of K-means is to find 2 clusters (like paid and charged off) and see if algorithm manages to cluster the same way as humans do. Table 5 describes results trained using K-means.

Table 5 K-means results

Description	Original prior probability	Modified prior probability
All columns	71.28%	50.00%
All columns except Credit score	71.29%	50.00%
Only Credit score column	78.89%	63.24%

Table 5 shows how important it is to modify prior probabilities before using K-means. Results with over 70% accuracy are not reliable at all. Correct K-means result is 50% and it shows that algorithm accuracy is just as accurate as a coin flip. Overall, K-means fails to cluster customer data like humans do in order to predict loan status.

However, one edge case exists, K-means did work as expected on a single column for Credit score, but we do not need K-means algorithm for that. It can be simply calculated using information gain formulas used in decision tree splitting.

Results and conclusion

Training data contained 16 columns and 56461 rows (32428 rows with modified prior probabilities). While exploring data, it was cleaned, irrelevant columns removed, rows with missing variables removed. Data was described using correlation matrix, PCA cumulative variance plot and decision tree. Most relevant columns were provided by decision tree, it helped to determine that most relevant column is Credit score. Minor relevance provided by Term, Annual income, Home ownership and Current loan amount columns.

Classification was done using LDA algorithm. With modified prior probabilities result is not spectacular, accuracy is only 64.56% with all columns included. However, we need to consider here that it was difficult dataset about real customers and institution did give a loan to a customer. Algorithm still could be used in a risk management – for example, calculating higher interest rate for more risky customers.

Clustering was done using K-means algorithm. Goal was to see if K-means manages to separate customers from those who pay and don't pay back. K-means have failed in this task and result was no better than a coin flip.

As a result of this research, it was determined that credit score has biggest impact on determining if customer pays back a loan. Also, Annual income, Home ownership, Current loan amount and Term have minimal effect. It was shown that it is essential to modify prior probabilities before using classification or clustering algorithms, otherwise results are unreliable.

Appendix

Sources

1. Bank Loan Status Dataset, Zaur Begiev

   https://www.kaggle.com/zaurbegiev/my-dataset

2. Principal Component Analysis (PCA) 101, using R, Peter Nistrup

   https://towardsdatascience.com/principal-component-analysis-pca-101-using-r-361f4c53a9ff

Scripts

Cleaning data

```
clean_data <- function(data) {
  data <- data[!duplicated(data), ]

  data$Customer_ID <- NULL
  data$Loan_ID <- NULL
  data$Months_since_last_delinquent <- NULL

  data <- data[data$Current_Loan_Amount != 99999999,]

  data <- data[!is.na(data$Current_Loan_Amount),]
  data <- data[!is.na(data$Bankruptcies),]
  data <- data[!is.na(data$Tax_Liens),]
  data <- data[!is.na(data$Number_of_Credit_Problems),]
  data <- data[!is.na(data$Maximum_Open_Credit),]
  data <- data[!is.na(data$Credit_Score),]
  data <- data[!is.na(data$Annual_Income),]
  data <- data[!is.na(data$Monthly_Debt),]
  data <- data[!is.na(data$Years_of_Credit_History),]
  data <- data[!is.na(data$Number_of_Open_Accounts),]
  data <- data[!is.na(data$Current_Credit_Balance),]

  data <- data[!is.na(data$Term),]
  data$Term[data$Term == "Long Term"] <- 0
  data$Term[data$Term == "Short Term"] <- 1
  data$Term <- as.numeric(data$Term)

  data <- data[!is.na(data$Years_in_current_job),]
  data$Years_in_current_job[data$Years_in_current_job == "< 1 year"] <- 0
  data$Years_in_current_job[data$Years_in_current_job == "1 year"] <- 1
  data$Years_in_current_job[data$Years_in_current_job == "2 years"] <- 2
  data$Years_in_current_job[data$Years_in_current_job == "3 years"] <- 3
  data$Years_in_current_job[data$Years_in_current_job == "4 years"] <- 4
  data$Years_in_current_job[data$Years_in_current_job == "5 years"] <- 5
  data$Years_in_current_job[data$Years_in_current_job == "6 years"] <- 6
  data$Years_in_current_job[data$Years_in_current_job == "7 years"] <- 7
  data$Years_in_current_job[data$Years_in_current_job == "8 years"] <- 8
  data$Years_in_current_job[data$Years_in_current_job == "9 years"] <- 9
  data$Years_in_current_job[data$Years_in_current_job == "10+ years"] <- 10
  data = data[data$Years_in_current_job != "n/a",]
  data$Years_in_current_job <- as.numeric(data$Years_in_current_job)

  data <- data[!is.na(data$Home_Ownership),]
  data$Home_Ownership[data$Home_Ownership == "HaveMortgage"] <- 0
  data$Home_Ownership[data$Home_Ownership == "Home Mortgage"] <- 1
  data$Home_Ownership[data$Home_Ownership == "Own Home"] <- 2
  data$Home_Ownership[data$Home_Ownership == "Rent"] <- 3
  data$Home_Ownership <- as.numeric(data$Home_Ownership)
```

```
  data <- data[!is.na(data$Purpose),]
  data$Purpose[data$Purpose == "Business Loan"] <- 0
  data$Purpose[data$Purpose == "Buy a Car"] <- 1
  data$Purpose[data$Purpose == "Buy House"] <- 2
  data$Purpose[data$Purpose == "Debt Consolidation"] <- 3
  data$Purpose[data$Purpose == "Educational Expenses"] <- 4
  data$Purpose[data$Purpose == "Home Improvements"] <- 5
  data$Purpose[data$Purpose == "major_purchase"] <- 6
  data$Purpose[data$Purpose == "Medical Bills"] <- 7
  data$Purpose[data$Purpose == "moving"] <- 8
  data$Purpose[data$Purpose == "renewable_energy"] <- 9
  data$Purpose[data$Purpose == "small_business"] <- 10
  data$Purpose[data$Purpose == "Take a Trip"] <- 11
  data$Purpose[data$Purpose == "vacation"] <- 12
  data$Purpose[data$Purpose == "wedding"] <- 13
  data$Purpose[data$Purpose == "other"] <- 14
  data$Purpose[data$Purpose == "Other"] <- 14
  data$Purpose <- as.numeric(data$Purpose)

  data <- data[!is.na(data$Loan_Status),]
  data$Loan_Status[data$Loan_Status == "Charged Off"] <- 0
  data$Loan_Status[data$Loan_Status == "Fully Paid"] <- 1
  data$Loan_Status <- factor(data$Loan_Status)

  data <- modify_prior_bias(data)

  return(data)
}

modify_prior_bias <- function(data) {
  rows <- sample(nrow(data))
  data <- data[rows, ]
  notpaid <- data[data$Loan_Status == 0,]
  paid <- data[data$Loan_Status == 1,]
  paid <- paid[1:dim(notpaid)[1],]
  data <- rbind(notpaid, paid)
  return(data)
}
```

Feature selection

```
feature_selection <- function(data) {
  fit_tree <- rpart(Loan_Status~.,data)
  feature_selection = varImp(fit_tree)
  feature_selection$Percent <- feature_selection$Overall / sum(feature_selection$Overall) * 100
  return(feature_selection)
}
```

Drawing plots

```
library(corrplot)

draw_plot <- function(data) {
  data$Loan_Status <- as.numeric(data$Loan_Status)
  M <- cor(data)
  corrplot(M, method='color')

  data$Loan_Status <- as.numeric(data$Loan_Status)
  result <- prcomp(data, center=TRUE, scale=TRUE)
  print(summary(result))
  screeplot(result, main="PCA", type="lines", ylim=c(0,3), npcs = 16)
  cumpro <- cumsum(result$sdev^2 / sum(result$sdev^2))
  plot(cumpro[0:16], xaxt="n", yaxt="n", xlab = "PC #", ylab = "Amount of explained variance", main =
"Cumulative variance plot")
  abline(v = 12, col="blue", lty=5)
  abline(h = 0.88759, col="blue", lty=5)
  xtick<-seq(0, 16, by=1)
  text(x=xtick,  par("usr")[3], labels = xtick, pos = 1, xpd = TRUE)
  ytick<-seq(0.2, 1, by=0.1)
  text(y=ytick,  par("usr")[3], labels = ytick, pos = 2, xpd = TRUE)
  axis(side=1, at=xtick, labels = FALSE)
}
```

Training models

```
run_lda <- function(data) {
  lda_results <- 0
  for (i in 1:10) {
    sample <- sample.int(n = nrow(data), size = floor(.75*nrow(data)), replace = F)
    train <- data[sample, ]
    test  <- data[-sample, ]
    lda_fit <- lda(Loan_Status~., data=train)
    lda_predict <- predict(lda_fit, newdata=test)$class
    lda_result <- sum(test$Loan_Status == lda_predict) / length(test$Loan_Status)
    lda_results[i] <- lda_result
  }
  return(mean(lda_results))
}

run_kmeans <- function(data) {
  data$Loan_Status <- NULL
  target <- data$Loan_Status
  kmeans_fit <- kmeans(data, 2, nstart = 1000)
  kmeans_predict <- kmeans_fit$cluster - 1
  kmeans_result <- sum(kmeans_predict == target) / length(kmeans_predict)
  return(kmeans_result)
}
```