Taller integrado de ciencias de datos
Proyecto 2 – Sebastián Farías

Identificar el problema:

El problema refiere a la aprobación de créditos mediante aplicaciones de tarjetas.

El desafío es representar y limpiar el dataset para posteriormente utilizar tres métodos de clasificación supervisada, entrenarlo y predecir en base a ello.

Recopilar datos:

Los datos se obtuvieron de UC Irvine Machine Learning Repository, en donde el dataset utilizado es el Credit Approval Data Set o de aprobación de créditos.

Problema a resolver:

Dataset:

Instancias: 690
Atributos: 16
Tipo de datos: numeric
factor
Valores perdidos: datos de diferentes atributos indicados con el símbolo '?'.
Outliers: datos atípicos encontrados en algunos atributos

Composición del dataset:

Los atributos son nombrados de la forma A_n, con n desde 1 a 16. Esto para resguardar la información de las cuentas y personas. Por ejemplo:

```
> head(datos)
  A1    A2    A3 A4 A5 A6 A7   A8 A9 A10 A11 A12 A13   A14 A15 A16
1  b 30.83 0.000  u  g  w  v 1.25  t   t   1   f   g 00202   0   +
2  a 58.67 4.460  u  g  q  h 3.04  t   t   6   f   g 00043 560   +
3  a 24.50 0.500  u  g  q  h 1.50  t   f   0   f   g 00280 824   +
4  b 27.83 1.540  u  g  w  v 3.75  t   t   5   t   g 00100   3   +
5  b 20.17 5.625  u  g  w  v 1.71  t   f   0   f   s 00120   0   +
6  b 32.08 4.000  u  g  m  v 2.50  t   f   0   t   g 00360   0   +
```

Limpieza de datos:

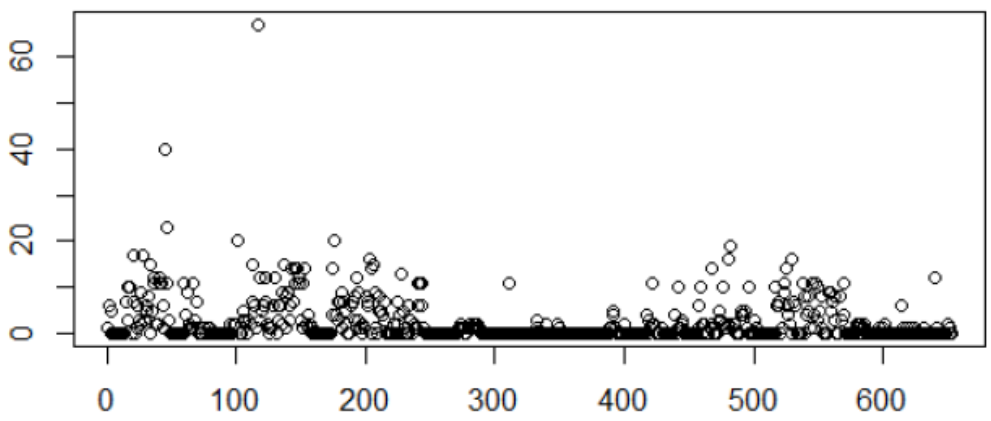
En primera instancia se identifican los datos perdidos, a través de la siguiente función en R:

```
> tabyl(datos$A1)
 datos$A1   n   percent
        ?  12 0.0173913
        a 210 0.3043478
        b 468 0.6782609
```

En donde, cuando se encontró un dato con el valor '?' fue eliminado del dataset. Esta acción se repitió para todos los atributos y se redujo el dataset a 653 elementos.

Posterior a ello, se identificaron los outliers, en donde se utilizaron los siguientes dos métodos:

```
> plot(datos$A11)
```




```
> tabyl(datos$A11)
 datos$A11   n     percent
         0 395 0.572463768
         1  71 0.102898551
         2  45 0.065217391
         3  28 0.040579710
         4  15 0.021739130
         5  18 0.026086957
         6  23 0.033333333
         7  16 0.023188406
         8  10 0.014492754
         9  10 0.014492754
        10   8 0.011594203
        11  19 0.027536232
        12   8 0.011594203
        13   1 0.001449275
        14   8 0.011594203
        15   4 0.005797101
        16   3 0.004347826
        17   2 0.002898551
        19   1 0.001449275
        20   2 0.002898551
        23   1 0.001449275
        40   1 0.001449275
        67   1 0.001449275
```

Donde se identifican dos, con valores de 67 y 40 para ese atributo respectivamente en este caso, los cuales fueron eliminados para entrenar correctamente los métodos de machine learning. Repitiéndose esto con los demás 16 atributos, lo cual redujo el dataset a 646 elementos.

Métodos de clasificación:

kNN:

Este método utiliza distancia euclidiana para dividir el espacio entre las distintas clases que contenga el dataset. Es por esto que se llama el k-vecino más cercano, y en donde, para realizar predicciones considera este aspecto para entregar una respuesta a la problemática.

Naive Bayes:

Se basa en el teorema de Bayes para resolver predicciones, es decir, la probabilidad de ocurrencia. Es un algoritmo simple, pero que tiene un nivel de éxito considerable, por lo que es altamente sofisticado en machine learning.

SVM:

Es un método de clasificación y regresión supervisado el cual posee la característica de agrupar las clases en diferentes kernel, ya sean lineal, polinomial, radial, entre otros. Su particularidad es que puede representar patrones de las formas antes mencionadas en dimensiones elevadas.

Librería e1071:

Esta librería se encuentra actualmente disponible en R, fue creada el 2 de febrero de 2017 y actualmente su versión es la 1.6-8. En ella se implementan una variedad de métodos de machine learning, de los cuales se escogerá para trabajar los siguientes:

- Naive Bayes
- SVM

Librería class:

Esta librería cuenta con una especialidad en kNN, el cual es el tercer método escogido. Fue creada el 30 de agosto de 2015 y se encuentra en la versión 7.3-14

Librería FNN:

Es una librería que al igual que class se creó para resolver problemas con el método de kNN y sus derivados. Fue creada el 19 de febrero de 2015 y se encuentra en la versión 1.1

Librería Naive Bayes:

Como lo dice su nombre, es una librería que se creó para resolver problemas con la metodología de Naive Bayes. Su creación fue el 3 de enero de 2018 y su versión actual es la 0.9.2

Librería kernlab:

Esta librería cuenta con una amplia variedad de métodos, y permite utilizar SVM. Fue creada el 30 de abril de 2018 y se encuentra en la versión 0.9-26

Entrenamiento y desempeño:

kNN:

Librería class:

Matriz de confusión:

```
> table(modelo,validacion)
      validacion
modelo  -  +
     0 19 13
     1 16 16
```

Accuracy: 54,68%

Precision: 54,28%

Recall: 59,37%

F-Score: 0,5671

Librería FNN:

Para aplicar el modelo, esta librería requiere que todos los atributos sean numéricos incluida la clase. Es por esto que el valor 1 representa los rechazos o '-' y 2 las aceptaciones o '+'

Matriz de confusión:

```
> table(modelo,validacion)
      validacion
modelo  1  2
     1 28 24
     2  7  5
```

Accuracy: 51,56%

Precision: 80%

Recall: 53,84%

F-Score: 0,6436

Naive Bayes:

Librería e1071:

Matriz de confusión:

```
> table(validacion, test$A16)

validacion  -  +
         - 26 10
         +  4 24
```

Accuracy: 78,12%

Precision: 86,66%

Recall: 72,22%

F-Score: 0,7878

Librería naivebayes:

Matriz de confusión:

```
> table(validacion, test$A16)

validacion  -  +
         - 25 29
         +  5  5
```

Accuracy: 46,87%

Precision: 83,33%

Recall: 46,29%

F-Score: 0,5952

SVM:

Librería e1071:

Matriz de confusión:

```
> table(validacion,test$A16)

validacion  -  +
         - 33 13
         +  3 15
```

Accuracy: 75%

Precision: 91,66%

Recall: 71,73%

F-Score: 0,8048

Librería kernlab:

Matriz de confusión:

```
> table(validacion,test$A16)

validacion  -  +
         - 29  7
         +  7 21
```

Accuracy: 78,12%

Precision: 80,55%

Recall: 80,55%

F-Score: 0,8055

Comparativa:

Al experimentar con el método knn, la respuesta más efectiva fue al utilizar la librería class, debido a que la exactitud de su respuesta fue 3,12% mejor. Pero si medimos la precisión, la librería fnn tuvo un desempeño superior.

Con el método de Naive Bayes los resultados fueron más claros, dado que la librería e1071 obtuvo mejores predicciones en los indicadores de exactitud y precisión que la naivebayes.

Finalmente, al predecir mediante el modelo support vector machine, los resultados fueron más efectivos en la librería kernlab en comparativa con la e1071, aunque esta última obtuvo mejores respuestas para la precisión de la predicción.