

```
clc; clear all; clf;
```

## Avance 2: Entrenamiento, adecuación y evaluación de modelos

Para este avance se tomarán los diferentes datasets obtenidos en el avance 1 y se procederá a buscar el mejor/mejores modelos de clasificación para predecir la posibilidad de intento de suicidio recurrente.

En un primer momento se probarán diferentes modelos, haciendo uso de la herramienta "Classification Learner" de Matlab, debido a su facilidad y rapidez para probar múltiples modelos simultáneamente. Se tomarán los pares dataset-modelo que mejores resultados den (preferiblemente por encima de 70% de acierto) para seguirlos desarrollando, en términos de selección de parámetros y optimización de hiperparámetros.

Para la selección de parámetros y ajuste de hiperparámetros, en donde sea posible se usarán herramientas las interactivas o automáticas que provee Matlab.

Como métodos de validación y calificación de los modelos se pretenden usar los datos a continuación (**To Do: añadir breve descripción de cada uno**)

- Score
- Matriz de confusión
- ROC curve
- F1

Al momento de realizar predicciones se generarán dos, una determinística y otra probabilística.

### Data sets de entrada.

En el avance 1 se obtuvieron 4 datasets después del proceso de limpieza, los cuales se mencionan a continuación:

- cds\_imputed : dataset con 33 características y 4146 registros
- cds : dataset con 28 características 4146 registros,
- cds\_few : dataset 33 características y 655 registros
- cds\_fem\_minus\_alcohol: dataset 32 características 1690 registros.

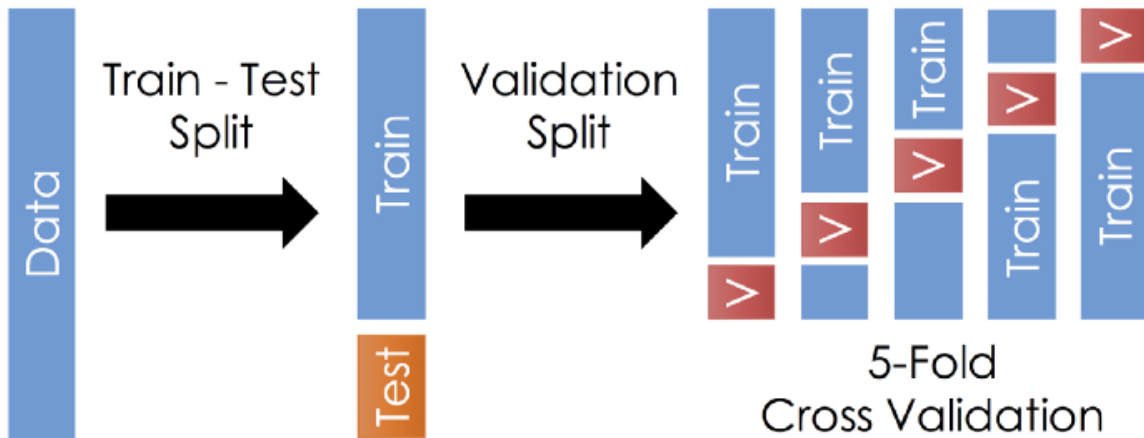
```
%cds = readtable('cds.csv'); size_cds = size(cds)
cds_imputed = readtable('cds_imputed.csv'); size_imputed = size(cds_imputed)
```

```
size_imputed = 1x2
              4146      34
```

```
cds_imputed = movevars(cds_imputed, 'inten_prev', 'after', 'tipo_ss_S');
%cds_few = readtable('cds_few.csv'); size_few = size (cds_few)
%cds_few_minus_alcohol = readtable('cds_few_minus_alcohol.csv');
%      size_few_minus_alcohol = size (cds_few_minus_alcohol)
```

Con estos dataset se procede realizar un enntrenamiento exploratorio de modelos , para continuar con los mas prometedores. Sin embargo, es necesarion definir el concepto de "mas prometedor". En este primer momento se tendra en cuenta la exactitud de los modelos

Es de utilidad tener en cuenta que para el entrenamiento de los modelos fue usada validacion cruzaada con "k-folds" ( k=5) ,asi, el valor de la exactitud presentado corresponde a la exactitud de validacion y esta sirve como un estimado del desempeño del modelo en nuevos datos comparados con el conjunto de entrenamiento.



**Resultados cds**

<b>1.1</b>	☆ Tree	Accuracy: 63.8%
Last change: Fine Tree		28/28 features
<b>1.2</b>	☆ Tree	Accuracy: 65.1%
Last change: Medium Tree		28/28 features
<b>1.3</b>	☆ Tree	Accuracy: 66.7%
Last change: Coarse Tree		28/28 features
<b>1.4</b>	☆ Linear Discriminant	Accuracy: 66.9%
Last change: Linear Discriminant		28/28 features
<b>1.5</b>	☆ Quadratic Discriminant	<b>Failed</b>
Last change: Quadratic Discriminant		28/28 features
<b>1.6</b>	☆ Logistic Regression	Accuracy: <b>67.4%</b>
Last change: Logistic Regression		28/28 features
<b>1.7</b>	☆ Naive Bayes	Accuracy: 62.2%
Last change: Gaussian Naive Bayes		28/28 features
<b>1.8</b>	☆ Naive Bayes	Accuracy: 62.0%
Last change: Kernel Naive Bayes		28/28 features
<b>1.9</b>	☆ SVM	Accuracy: 66.0%
Last change: Linear SVM		28/28 features
<b>1.10</b>	☆ SVM	Accuracy: 65.3%
Last change: Quadratic SVM		28/28 features
<b>1.11</b>	☆ SVM	Accuracy: 63.8%
Last change: Cubic SVM		28/28 features
<b>1.12</b>	☆ SVM	Accuracy: 61.8%
Last change: Fine Gaussian SVM		28/28 features
<b>1.13</b>	☆ SVM	Accuracy: 65.8%
Last change: Medium Gaussian SVM		28/28 features

<b>1.14</b>	☆ SVM	Accuracy: 66.8%
Last change: Coarse Gaussian SVM 28/28 features		
<b>1.15</b>	☆ KNN	Accuracy: 59.5%
Last change: Fine KNN 28/28 features		
<b>1.16</b>	☆ KNN	Accuracy: 63.8%
Last change: Medium KNN 28/28 features		
<b>1.17</b>	☆ KNN	Accuracy: 65.6%
Last change: Coarse KNN 28/28 features		
<b>1.18</b>	☆ KNN	Accuracy: 63.7%
Last change: Cosine KNN 28/28 features		
<b>1.19</b>	☆ KNN	Accuracy: 63.3%
Last change: Cubic KNN 28/28 features		
<b>1.20</b>	☆ KNN	Accuracy: 62.4%
Last change: Weighted KNN 28/28 features		
<b>1.21</b>	☆ Ensemble	Accuracy: 66.6%
Last change: Boosted Trees 28/28 features		
<b>1.22</b>	☆ Ensemble	Accuracy: 65.4%
Last change: Bagged Trees 28/28 features		
<b>1.23</b>	☆ Ensemble	Accuracy: 66.8%
Last change: Subspace Discriminant 28/28 features		
<b>1.24</b>	☆ Ensemble	Accuracy: 63.6%
Last change: Subspace KNN 28/28 features		
<b>1.25</b>	☆ Ensemble	Accuracy: 64.7%
Last change: RUSBoosted Trees 28/28 features		
<b>2</b>	☆ Quadratic Discriminant	Accuracy: 62.2%
Last change: 'Covariance structure' ... 28/28 features		

Resultados cds\_imputed

<b>1.1</b>	☆ Tree	Accuracy: 64.5%
Last change: Fine Tree		33/33 features
<b>1.2</b>	☆ Tree	Accuracy: 66.1%
Last change: Medium Tree		33/33 features
<b>1.3</b>	☆ Tree	Accuracy: 66.8%
Last change: Coarse Tree		33/33 features
<b>1.4</b>	☆ Linear Discriminant	Accuracy: <b>67.9%</b>
Last change: Linear Discriminant		33/33 features
<b>1.5</b>	☆ Quadratic Discriminant	<b>Failed</b>
Last change: Quadratic Discriminant		33/33 features
<b>1.6</b>	☆ Logistic Regression	Accuracy: 67.8%
Last change: Logistic Regression		33/33 features
<b>1.7</b>	☆ Naive Bayes	Accuracy: 64.2%
Last change: Gaussian Naive Bayes		33/33 features
<b>1.8</b>	☆ Naive Bayes	Accuracy: 62.5%
Last change: Kernel Naive Bayes		33/33 features
<b>1.9</b>	☆ SVM	Accuracy: 66.8%
Last change: Linear SVM		33/33 features
<b>1.10</b>	☆ SVM	Accuracy: 65.4%
Last change: Quadratic SVM		33/33 features
<b>1.11</b>	☆ SVM	Accuracy: 63.3%
Last change: Cubic SVM		33/33 features
<b>1.12</b>	☆ SVM	Accuracy: 61.8%
Last change: Fine Gaussian SVM		33/33 features
<b>1.13</b>	☆ SVM	Accuracy: 65.5%
Last change: Medium Gaussian SVM		33/33 features

<b>1.14</b>	☆ SVM	Accuracy: 67.6%
Last change: Coarse Gaussian SVM 33/33 features		
<b>1.15</b>	☆ KNN	Accuracy: 61.3%
Last change: Fine KNN 33/33 features		
<b>1.16</b>	☆ KNN	Accuracy: 64.4%
Last change: Medium KNN 33/33 features		
<b>1.17</b>	☆ KNN	Accuracy: 64.8%
Last change: Coarse KNN 33/33 features		
<b>1.18</b>	☆ KNN	Accuracy: 64.6%
Last change: Cosine KNN 33/33 features		
<b>1.19</b>	☆ KNN	Accuracy: 64.5%
Last change: Cubic KNN 33/33 features		
<b>1.20</b>	☆ KNN	Accuracy: 63.7%
Last change: Weighted KNN 33/33 features		
<b>1.21</b>	☆ Ensemble	Accuracy: 67.6%
Last change: Boosted Trees 33/33 features		
<b>1.22</b>	☆ Ensemble	Accuracy: 65.6%
Last change: Bagged Trees 33/33 features		
<b>1.23</b>	☆ Ensemble	Accuracy: 67.7%
Last change: Subspace Discriminant 33/33 features		
<b>1.24</b>	☆ Ensemble	Accuracy: 63.8%
Last change: Subspace KNN 33/33 features		
<b>1.25</b>	☆ Ensemble	Accuracy: 64.7%
Last change: RUSBoosted Trees 33/33 features		
<b>2</b>	☆ Quadratic Discriminant	Accuracy: 64.2%
Last change: 'Covariance structure' ... 33/33 features		

## Resultados cds\_few

Para este dataset algunos modelos se hicieron individualmente, porque presentaban problemas con las características 'antec\_tran', 'tipo\_ss\_l', 'suici\_fm\_a' y 'tipo\_SS\_P' ya que la mayoría o casi todos sus valores son iguales por lo que no aportan información o no presentan variación con respecto a una de las clases por hallar.

<b>1.1</b> ☆ Tree	Accuracy: 53.9%
Last change: Fine Tree	33/33 features
<b>1.2</b> ☆ Tree	Accuracy: 60.5%
Last change: Medium Tree	33/33 features
<b>1.3</b> ☆ Tree	Accuracy: <b>64.4%</b>
Last change: Coarse Tree	33/33 features
<b>1.4</b> ☆ Linear Discriminant	<b>Failed</b>
Last change: Linear Discriminant	33/33 features
<b>1.5</b> ☆ Quadratic Discriminant	<b>Failed</b>
Last change: Quadratic Discriminant	33/33 features
<b>1.6</b> ☆ Logistic Regression	Accuracy: 61.8%
Last change: Logistic Regression	33/33 features
<b>1.7</b> ☆ Naive Bayes	<b>Failed</b>
Last change: Gaussian Naive Bayes	33/33 features
<b>1.8</b> ☆ Naive Bayes	Accuracy: 61.1%
Last change: Kernel Naive Bayes	33/33 features
<b>1.9</b> ☆ SVM	Accuracy: 61.2%
Last change: Linear SVM	33/33 features
<b>1.10</b> ☆ SVM	Accuracy: 57.7%
Last change: Quadratic SVM	33/33 features
<b>1.11</b> ☆ SVM	Accuracy: 57.3%
Last change: Cubic SVM	33/33 features
<b>1.12</b> ☆ SVM	Accuracy: 58.9%
Last change: Fine Gaussian SVM	33/33 features
<b>1.13</b> ☆ SVM	Accuracy: 63.7%
Last change: Medium Gaussian SVM	33/33 features
<b>1.14</b> ☆ SVM	Accuracy: 61.1%
Last change: Coarse Gaussian SVM	33/33 features



<b>1.15</b>	☆ KNN	Accuracy: 53.6%
Last change: Fine KNN		33/33 features
<b>1.16</b>	☆ KNN	Accuracy: 60.8%
Last change: Medium KNN		33/33 features
<b>1.17</b>	☆ KNN	Accuracy: 60.8%
Last change: Coarse KNN		33/33 features
<b>1.18</b>	☆ KNN	Accuracy: 61.2%
Last change: Cosine KNN		33/33 features
<b>1.19</b>	☆ KNN	Accuracy: 60.2%
Last change: Cubic KNN		33/33 features
<b>1.20</b>	☆ KNN	Accuracy: 57.7%
Last change: Weighted KNN		33/33 features
<b>1.21</b>	☆ Ensemble	Accuracy: 59.1%
Last change: Boosted Trees		33/33 features
<b>1.22</b>	☆ Ensemble	Accuracy: 58.3%
Last change: Bagged Trees		33/33 features
<b>1.23</b>	☆ Ensemble	Accuracy: 63.4%
Last change: Subspace Discriminant		33/33 features
<b>1.24</b>	☆ Ensemble	Accuracy: 57.3%
Last change: Subspace KNN		33/33 features
<b>1.25</b>	☆ Ensemble	Accuracy: 58.3%
Last change: RUSBoosted Trees		33/33 features
<b>2</b>	☆ Linear Discriminant	Accuracy: 61.5%
Last change: 'Covariance structure' ...		33/33 features
<b>3</b>	☆ Quadratic Discriminant	Accuracy: 60.6%
Last change: 'Covariance structure' ...		33/33 features
<b>4</b>	☆ Naive Bayes	Accuracy: 53.6%
Last change: Removed 3 features		29/33 features

Resultados cds\_few\_minus\_alcohol



<b>1.1</b>	☆ Tree	Accuracy: 54.2%
Last change: Fine Tree		32/32 features
<b>1.2</b>	☆ Tree	Accuracy: 55.6%
Last change: Medium Tree		32/32 features
<b>1.3</b>	☆ Tree	Accuracy: 55.6%
Last change: Coarse Tree		32/32 features
<b>1.4</b>	☆ Linear Discriminant	<u>Failed</u>
Last change: Linear Discriminant		32/32 features
<b>1.5</b>	☆ Quadratic Discriminant	<u>Failed</u>
Last change: Quadratic Discriminant		32/32 features
<b>1.6</b>	☆ Logistic Regression	Accuracy: 59.0%
Last change: Logistic Regression		32/32 features
<b>1.7</b>	☆ Naive Bayes	<u>Failed</u>
Last change: Gaussian Naive Bayes		32/32 features
<b>1.8</b>	☆ Naive Bayes	Accuracy: 55.0%
Last change: Kernel Naive Bayes		32/32 features
<b>1.9</b>	☆ SVM	Accuracy: 58.5%
Last change: Linear SVM		32/32 features
<b>1.10</b>	☆ SVM	Accuracy: 55.3%
Last change: Quadratic SVM		32/32 features
<b>1.11</b>	☆ SVM	Accuracy: 53.3%
Last change: Cubic SVM		32/32 features
<b>1.12</b>	☆ SVM	Accuracy: 54.3%
Last change: Fine Gaussian SVM		32/32 features
<b>1.13</b>	☆ SVM	Accuracy: 56.5%
Last change: Medium Gaussian SVM		32/32 features
<b>1.14</b>	☆ SVM	Accuracy: 58.4%
Last change: Coarse Gaussian SVM		32/32 features

<b>1.15</b>	☆ KNN	Accuracy: 53.7%
Last change: Fine KNN		32/32 features
<b>1.16</b>	☆ KNN	Accuracy: 55.1%
Last change: Medium KNN		32/32 features
<b>1.17</b>	☆ KNN	Accuracy: 56.2%
Last change: Coarse KNN		32/32 features
<b>1.18</b>	☆ KNN	Accuracy: 55.4%
Last change: Cosine KNN		32/32 features
<b>1.19</b>	☆ KNN	Accuracy: 54.7%
Last change: Cubic KNN		32/32 features
<b>1.20</b>	☆ KNN	Accuracy: 55.0%
Last change: Weighted KNN		32/32 features
<b>1.21</b>	☆ Ensemble	Accuracy: 59.2%
Last change: Boosted Trees		32/32 features
<b>1.22</b>	☆ Ensemble	Accuracy: 56.6%
Last change: Bagged Trees		32/32 features
<b>1.23</b>	☆ Ensemble	Accuracy: 58.1%
Last change: Subspace Discriminant		32/32 features
<b>1.24</b>	☆ Ensemble	Accuracy: 54.0%
Last change: Subspace KNN		32/32 features
<b>1.25</b>	☆ Ensemble	Accuracy: 57.3%
Last change: RUSBoosted Trees		32/32 features
<b>2</b>	☆ Linear Discriminant	Accuracy: <b>60.2%</b>
Last change: 'Covariance structure' ...		32/32 features
<b>3</b>	☆ Quadratic Discriminant	Accuracy: 55.0%
Last change: 'Covariance structure' ...		32/32 features
<b>4</b>	☆ Naive Bayes	Accuracy: 54.4%
Last change: Removed 3 features		28/32 features

Por motivos exploratorios se realizaron pruebas aplicandole PCA a los datos, pero los resultados en general fueron inferiores a los obtenidos sin esta transformacion, por lo que esta transformacion de los datos no sera utilizada. (*¿Uno si deberia hacer PCA en datos categoricos?*)

Como se puede notar, ningún par dataset-modelo obtuvo una precisión mayor al 70% tal y como se había definido inicialmente para su aceptación. Por este motivo se tomará aquel dataset que produjo el modelo con la mayor precisión (cds\_imputed) y los mejores modelos obtenidos a partir de este -Coarse Tree, Linear discriminant, Logistic regresión, SVM (lineal y coarse) y Ensemble(BoostTrees, SubsD)-

## Feature selection

Buscando reducir la dimensionalidad y explorar diferentes opciones se pretende realizar un proceso de selección de características. Esto se hará filtrando aquellas características menos importantes para la respuesta 'inten\_prev' mediante el algoritmo MRMR (Minimum Redundancy Maximum Relevance), del cual se puede obtener el "ranking" de importancia de los predictores teniendo en cuenta la respuesta.

Se entrenarán 2 modelos, uno con todas las características y adicionalmente otro con el conjunto de las 7 más importantes

```
idx = fscmr(r(cds_imputed, 'inten_prev'));  
most_signif_features = cds_imputed.Properties.VariableNames(idx(1:7)).'
```

```
most_signif_features = 7x1 cell  
'antec_tran'  
'hist_famil'  
'muerte_fam'  
'antec_v_a'  
'prob_consu'  
'plan_suici'  
'gp_psiquia'
```

```
less_signif_features = cds_imputed.Properties.VariableNames(idx(end-4:end)).'
```

```
less_signif_features = 5x1 cell  
'escolarid'  
'esco_educ'  
'tipo_ss_C'  
'trab_socia'  
'sexo_'
```

## Optimización de hiperparámetros

Para la optimización de hiperparámetros serán utilizados dos enfoques:

Para los modelos simples (e.g. árboles de decisión) se realizará mediante GridSearch, mientras que para los más complejos (Ensamble, SVM) será utilizado un método de optimización bayesiana, el cual, a través de 30 iteraciones se va redirigiendo hacia aquellos hiperparámetros del espacio de búsqueda que proveen mejores resultados para el modelo. Esta elección se hace debido a los costos computacionales elevados de realizar Grid o Random Search en modelos complejos.

Como parte del proceso de optimización se obtiene el gráfico de error de clasificación mínimo en el cual se encuentran principalmente :

- Los resultados de la optimización
- El mínimo error de clasificación observado (puntos azules) hasta la iteración actual
- Bestpoint hyperparameters (cuadrado rojo), indica la iteración que corresponde a los valores de los hiperparámetros optimizados

En general hubo un desempeño un poco más alto usando todas las características del dataset que usando las 7 más importantes

Mínimo error de clasificación

Iteración

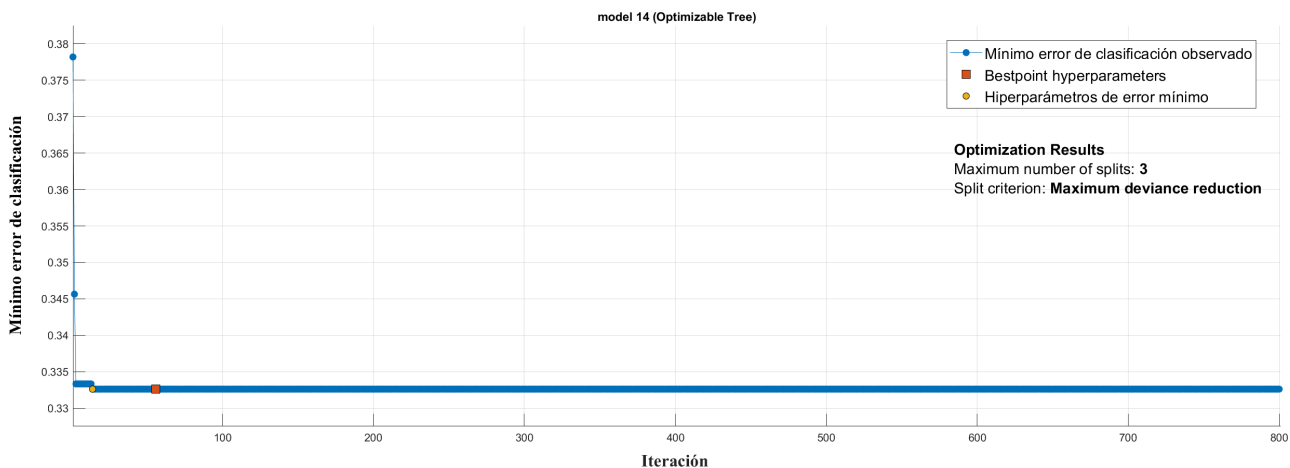
Mínimo error de clasificación observado

Hiperparámetros de error mínimo

Los resultados se presentarán a continuación para cada modelo:

Árboles de decisión:

En la anterior etapa mejores resultados fueron obtenidos con árboles de decisión gruesos (con poco número de splits), lo cual es confirmado con los resultados de este proceso



**Árboles de decisión (7 características):**

Para este caso los mejores resultados fueron con árboles finos

Discriminante:

Anteriormente se había hallado que el lineal era el que mejores resultados presentaba, esto se comprueba/ reafirma al realizar este paso.

Las combinaciones disponibles para este tipo de modelos son pocas por lo que con pocas iteraciones es suficiente

Discriminante(7 características):

Logistic regression:

Este modelo no presenta hiperparametros para ajustar.

SVM:

Ensamble:

## **Presentacion de resultados de los modelos entrenados con el data set completo y el de características reducidas**

*Hacerlo en terminos de matriz de confusion. ROC,...*

## **Comparacion de modelos**

Para este punto se tendran en cuenta varias cosas:

- La precision y exactitud del modelo, mientras mayor mejor, sin llegar a un caso de sobreajuste.
- El numero de parametros, en general es de interes obtener modelos que con un bajo numero de parametros sean capaces de cumplir con su objetivo a cabalidad, esto debido a que en un caso real es mas dificil y costoso, en terminos de dinero y tiempo obtener una cantidad grande de informacion. En este caso no se le dara mayor importancia a unos parametros sobre otros, solo sera de interes el numero de ellos.
- Un ultimo factor que se tendra en cuenta para preferir un modelo sobre otro, es la distribucion de falsos negativos hacia cierta clase particular, i.e. en este contexto no seria nada bueno identificar erroneamente a aquellas personas con tendencia repetitiva al intento de suicidio, mientras que identificar erroneamente a aquellos que en realidad no(falso positivo), seria mas aceptable.

When you open the plot, the rows show the true class, and the columns show the predicted class. If you are using holdout or cross-validation, then the confusion matrix is calculated using the predictions on the held-out observations. The diagonal cells show where the true class and predicted class match. If these diagonal cells are blue, the classifier has classified observations of this true class are classified correctly.

## **Conclusiones**

*El por que nuestros modelos son tan malos(posibles razones:) y como seria excelente de acuerdo a la propuesta inicial poder conseguir datasets con informacion de personas que han intentando previamente el suicidio como de aquella que no*

*Sugeridos por el profesor: Regresion Logistica, SVM, Arboles de decision, Redes neuronales, LMP, Random Fores*

*Intent\_prev{1 = SI; 2 = NO}*

## **Referencias**

<https://www.mathworks.com/help/stats/feature-selection-and-feature-transformation.html>

<https://www.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html>

<https://www.mathworks.com/help/stats/assess-classifier-performance.html>

<https://towardsdatascience.com/intuitive-hyperparameter-optimization-grid-search-random-search-and-bayesian-search-2102dbfaf5b>

<https://towardsdatascience.com/automated-machine-learning-hyperparameter-tuning-in-python-dfda59b72f8a>

<https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94>

<https://www.mathworks.com/help/stats/feature-selection.html>

<https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

<https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>