```
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 data set-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 dados a continuación ***(To Do: añadir breve descripción de cada uno)***

- ¿Score?
- Matriz de confusión
- ROC curve

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

Data sets de entrada.

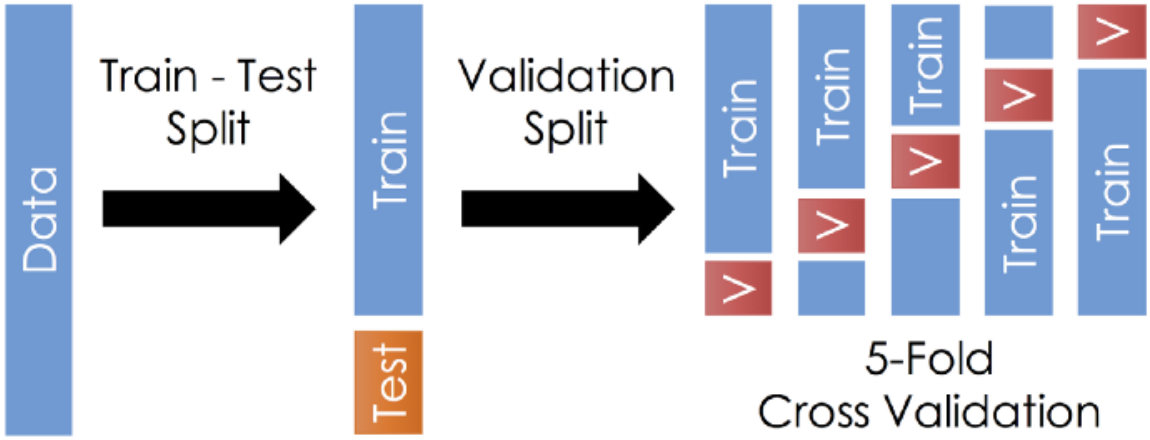
En el avance 1 se obtuvieron 4 datasets después del proceso de limpieza, los cuales se mencionan a continuación:

- 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('clean_datasets\cds.csv'); size_cds = size(cds)
cds_imputed = readtable('clean_datasets\cds_imputed.csv'); size_imputed = size(cds_imputed)
cds_imputed = movevars(cds_imputed,'inten_prev','after','tipo_ss_S');
%cds_few = readtable('clean_datasets\cds_few.csv'); size_few = size (cds_few)
%cds_few_minus_alcohol = readtable('clean_datasets\cds_few_minus_alcohol.csv');
%       size_few_minus_alcohol = size (cds_few_minus_alcohol)
```

Con estos dataset se procede realizar un entrenamiento exploratorio de modelos, para continuar con los más prometedores. Sin embargo, es necesario definir el concepto de "más prometedor". En este primer momento se tendrá en cuanta la exactitud de los modelos

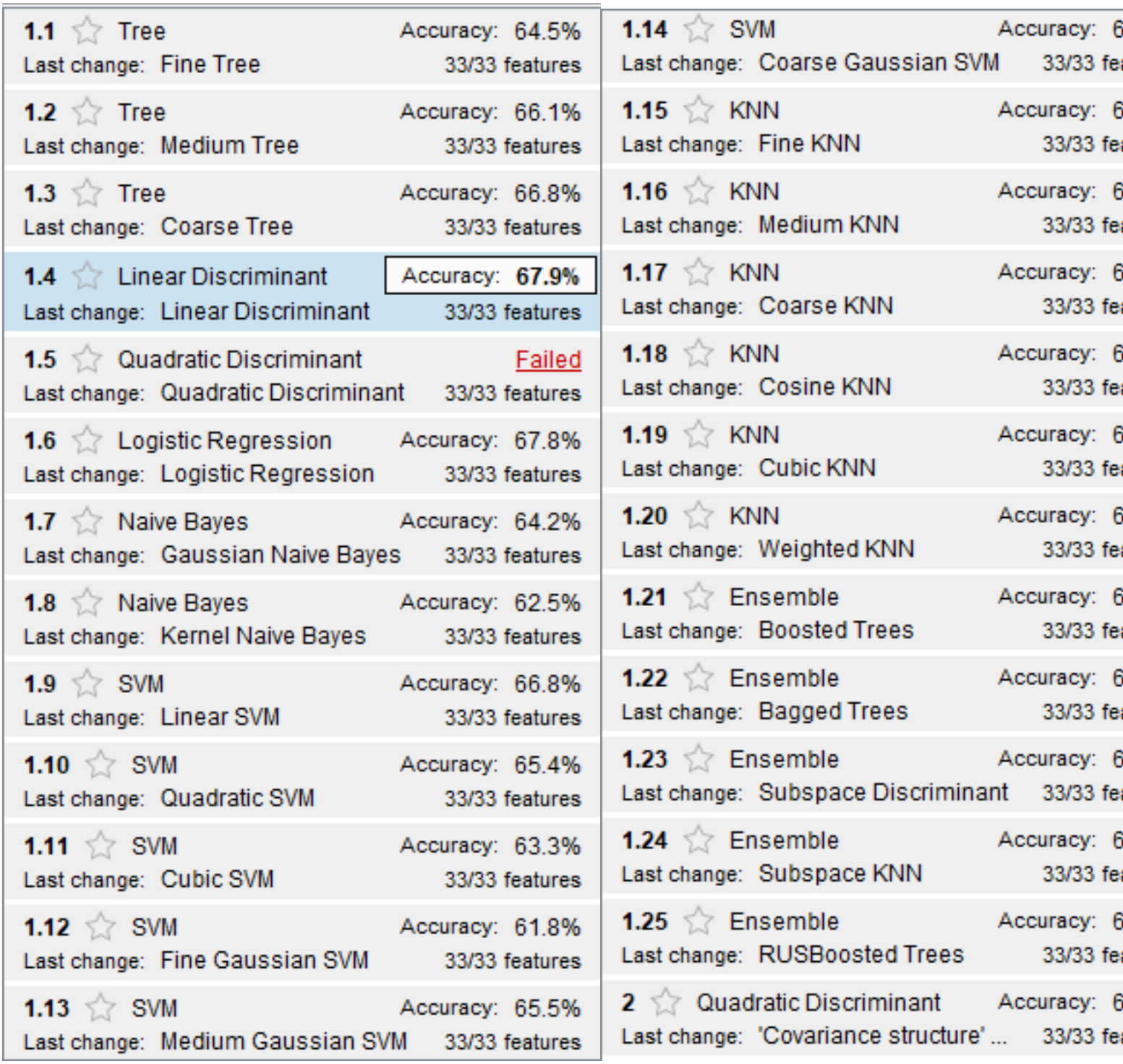
Es de utilidad tener en cuenta que para el entrenamiento de los modelos fue usada validación cruzada con "k-folds" (k=5), así, el valor de la exactitud presentado corresponde a la exactitud de validación y esta sirve como un estimado del desempeño del modelo en nuevos datos comparados con el conjunto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



Resultados cds

1.1 ☆ Tree	Accuracy: 63.8%	**1.14** ☆ SVM	Accuracy: 6
Last change: Fine Tree	28/28 features	Last change: Coarse Gaussian SVM	28/28 fea
1.2 ☆ Tree	Accuracy: 65.1%	**1.15** ☆ KNN	Accuracy: 5
Last change: Medium Tree	28/28 features	Last change: Fine KNN	28/28 fea
1.3 ☆ Tree	Accuracy: 66.7%	**1.16** ☆ KNN	Accuracy: 6
Last change: Coarse Tree	28/28 features	Last change: Medium KNN	28/28 fea
1.4 ☆ Linear Discriminant	Accuracy: 66.9%	**1.17** ☆ KNN	Accuracy: 6
Last change: Linear Discriminant	28/28 features	Last change: Coarse KNN	28/28 fea
1.5 ☆ Quadratic Discriminant	Failed	**1.18** ☆ KNN	Accuracy: 6
Last change: Quadratic Discriminant	28/28 features	Last change: Cosine KNN	28/28 fea
1.6 ☆ Logistic Regression	Accuracy: **67.4%**	**1.19** ☆ KNN	Accuracy: 6
Last change: Logistic Regression	28/28 features	Last change: Cubic KNN	28/28 fea
1.7 ☆ Naive Bayes	Accuracy: 62.2%	**1.20** ☆ KNN	Accuracy: 6
Last change: Gaussian Naive Bayes	28/28 features	Last change: Weighted KNN	28/28 fea
1.8 ☆ Naive Bayes	Accuracy: 62.0%	**1.21** ☆ Ensemble	Accuracy: 6
Last change: Kernel Naive Bayes	28/28 features	Last change: Boosted Trees	28/28 fea
1.9 ☆ SVM	Accuracy: 66.0%	**1.22** ☆ Ensemble	Accuracy: 6
Last change: Linear SVM	28/28 features	Last change: Bagged Trees	28/28 fea
1.10 ☆ SVM	Accuracy: 65.3%	**1.23** ☆ Ensemble	Accuracy: 6
Last change: Quadratic SVM	28/28 features	Last change: Subspace Discriminant	28/28 fea
1.11 ☆ SVM	Accuracy: 63.8%	**1.24** ☆ Ensemble	Accuracy: 6
Last change: Cubic SVM	28/28 features	Last change: Subspace KNN	28/28 fea
1.12 ☆ SVM	Accuracy: 61.8%	**1.25** ☆ Ensemble	Accuracy: 6
Last change: Fine Gaussian SVM	28/28 features	Last change: RUSBoosted Trees	28/28 fea
1.13 ☆ SVM	Accuracy: 65.8%	**2** ☆ Quadratic Discriminant	Accuracy: 6
Last change: Medium Gaussian SVM	28/28 features	Last change: 'Covariance structure' ...	28/28 fea

Resultados cds_imputed



Resultados cds_few

Para este dataset algunos modelos se hicieron individualmente, porque presentaban problemas con las características 'antec_tran', 'tipo_ss_I', '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..

1.1	Tree	Accuracy: 53.9%	**1.15**	KNN	Accuracy: 53.6%
Last change:	Fine Tree	33/33 features	Last change:	Fine KNN	33/33 features
1.2	Tree	Accuracy: 60.5%	**1.16**	KNN	Accuracy: 60.8%
Last change:	Medium Tree	33/33 features	Last change:	Medium KNN	33/33 features
1.3	Tree	Accuracy: **64.4%**	**1.17**	KNN	Accuracy: 60.8%
Last change:	Coarse Tree	33/33 features	Last change:	Coarse KNN	33/33 features
1.4	Linear Discriminant	Failed	**1.18**	KNN	Accuracy: 61.2%
Last change:	Linear Discriminant	33/33 features	Last change:	Cosine KNN	33/33 features
1.5	Quadratic Discriminant	Failed	**1.19**	KNN	Accuracy: 60.2%
Last change:	Quadratic Discriminant	33/33 features	Last change:	Cubic KNN	33/33 features
1.6	Logistic Regression	Accuracy: 61.8%	**1.20**	KNN	Accuracy: 57.7%
Last change:	Logistic Regression	33/33 features	Last change:	Weighted KNN	33/33 features
1.7	Naive Bayes	Failed	**1.21**	Ensemble	Accuracy: 59.1%
Last change:	Gaussian Naive Bayes	33/33 features	Last change:	Boosted Trees	33/33 features
1.8	Naive Bayes	Accuracy: 61.1%	**1.22**	Ensemble	Accuracy: 58.3%
Last change:	Kernel Naive Bayes	33/33 features	Last change:	Bagged Trees	33/33 features
1.9	SVM	Accuracy: 61.2%	**1.23**	Ensemble	Accuracy: 63.4%
Last change:	Linear SVM	33/33 features	Last change:	Subspace Discriminant	33/33 features
1.10	SVM	Accuracy: 57.7%	**1.24**	Ensemble	Accuracy: 57.3%
Last change:	Quadratic SVM	33/33 features	Last change:	Subspace KNN	33/33 features
1.11	SVM	Accuracy: 57.3%	**1.25**	Ensemble	Accuracy: 58.3%
Last change:	Cubic SVM	33/33 features	Last change:	RUSBoosted Trees	33/33 features
1.12	SVM	Accuracy: 58.9%	**2**	Linear Discriminant	Accuracy: 61.5%
Last change:	Fine Gaussian SVM	33/33 features	Last change:	'Covariance structure' ...	33/33 features
1.13	SVM	Accuracy: 63.7%	**3**	Quadratic Discriminant	Accuracy: 60.6%
Last change:	Medium Gaussian SVM	33/33 features	Last change:	'Covariance structure' ...	33/33 features
1.14	SVM	Accuracy: 61.1%	**4**	Naive Bayes	Accuracy: 53.6%
Last change:	Coarse Gaussian SVM	33/33 features	Last change:	Removed 3 features	29/33 features

Resultados cds_few_minus_alcohol

Por motivos exploratorios se realizaron pruebas aplicándole PCA a los datos, pero los resultados en general fueron inferiores a los obtenidos sin esta transformación, por lo que esta transformación de los datos no será utilizada. ***(¿Uno si debería hacer PCA en datos categóricos?)***

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 regresion, SVM (linear y coarse) y Ensamble (BoostTrees)-

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 Maximun Relevance), del cual se puede obtener el "ranking" de importancia de los predictores teniendo en cuentas la respuesta.

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

```
idx = fscmrmr(cds_imputed,'inten_prev');
most_signif_features = cds_imputed.Properties.VariableNames(idx(1:7)).'
less_signif_features =cds_imputed.Properties.VariableNames(idx(end-4:end)).'
```

Optimización de hiperparámetros

Para la optimización de hiperparámetros serán utilizados dos enfoques:

Para los modelos simples (arboles 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.

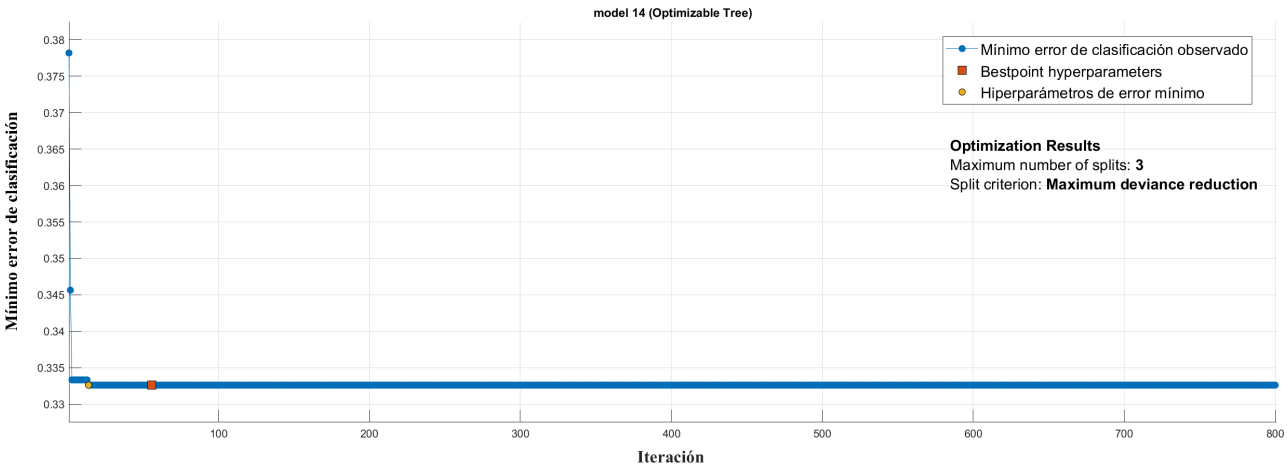
Como parte del proceso de optimización se obtiene el gráfico de error de clasificación mínimo en el cual se encuentran principalmente:

- 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

Los resultados se presentarán a continuación para cada modelo:

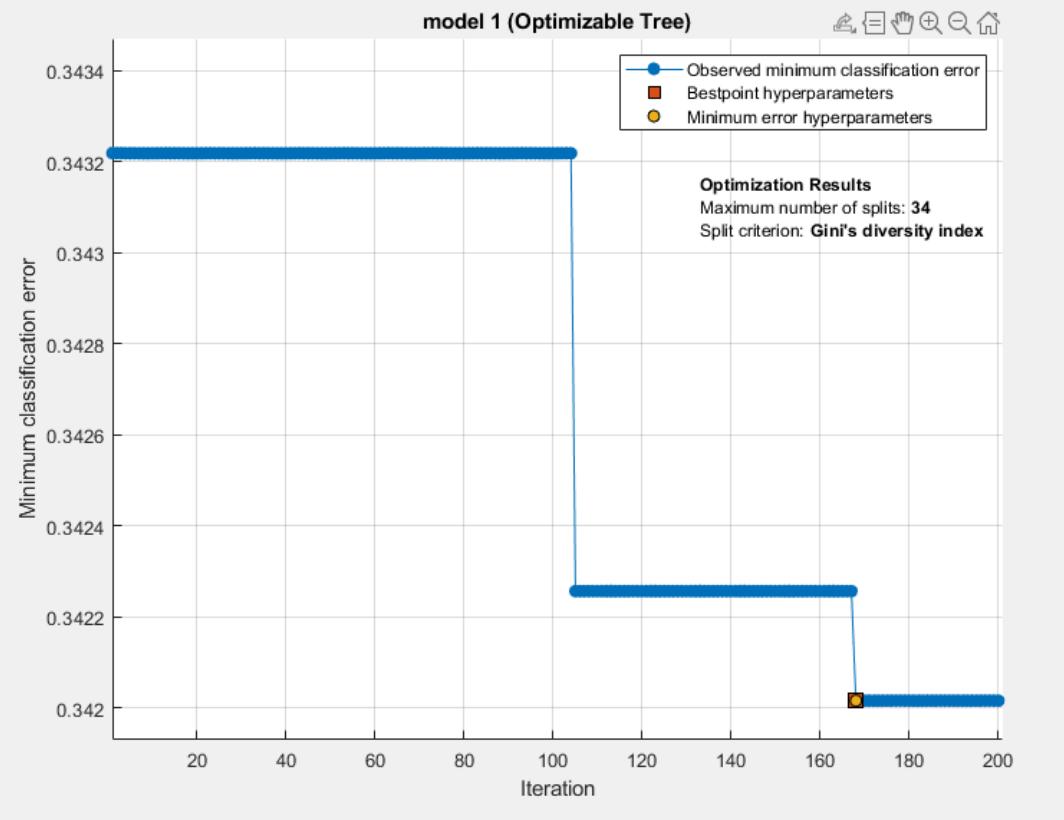
Arboles de decisión:

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



Arboles de decisión (7 características):

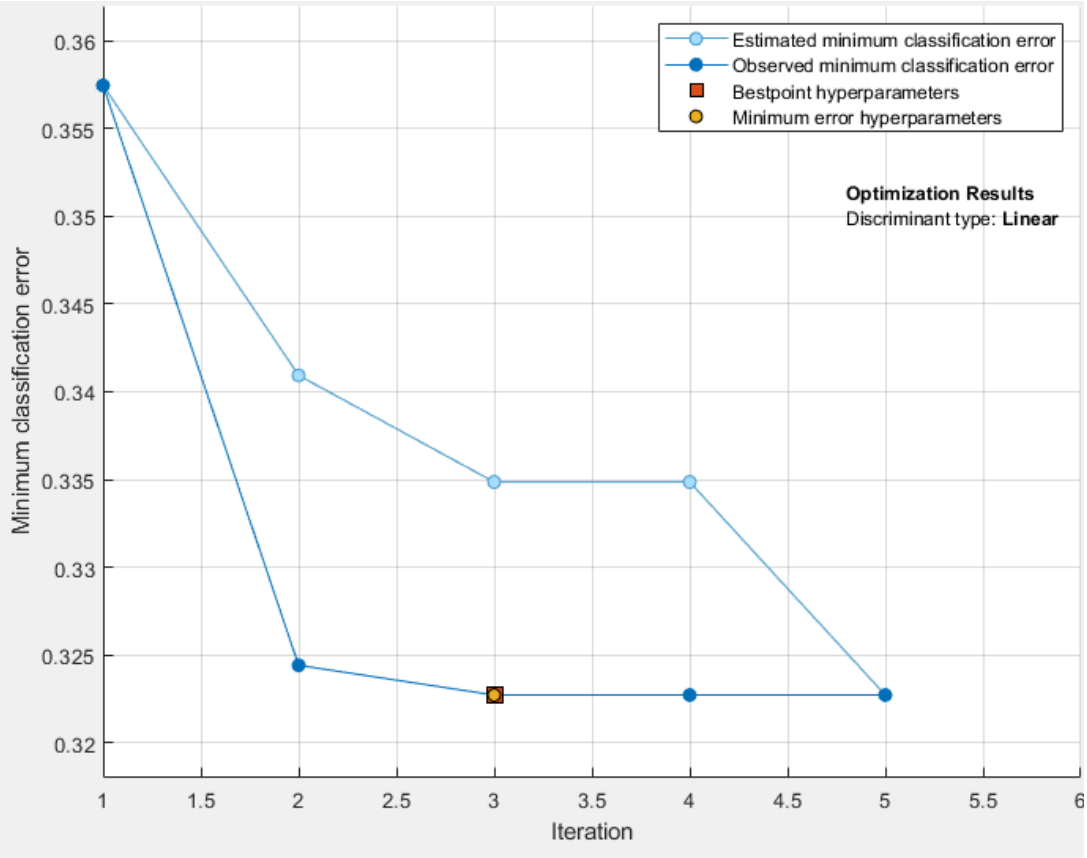
Para este caso los mejore resultados fueron con árboles finos



Discriminante:

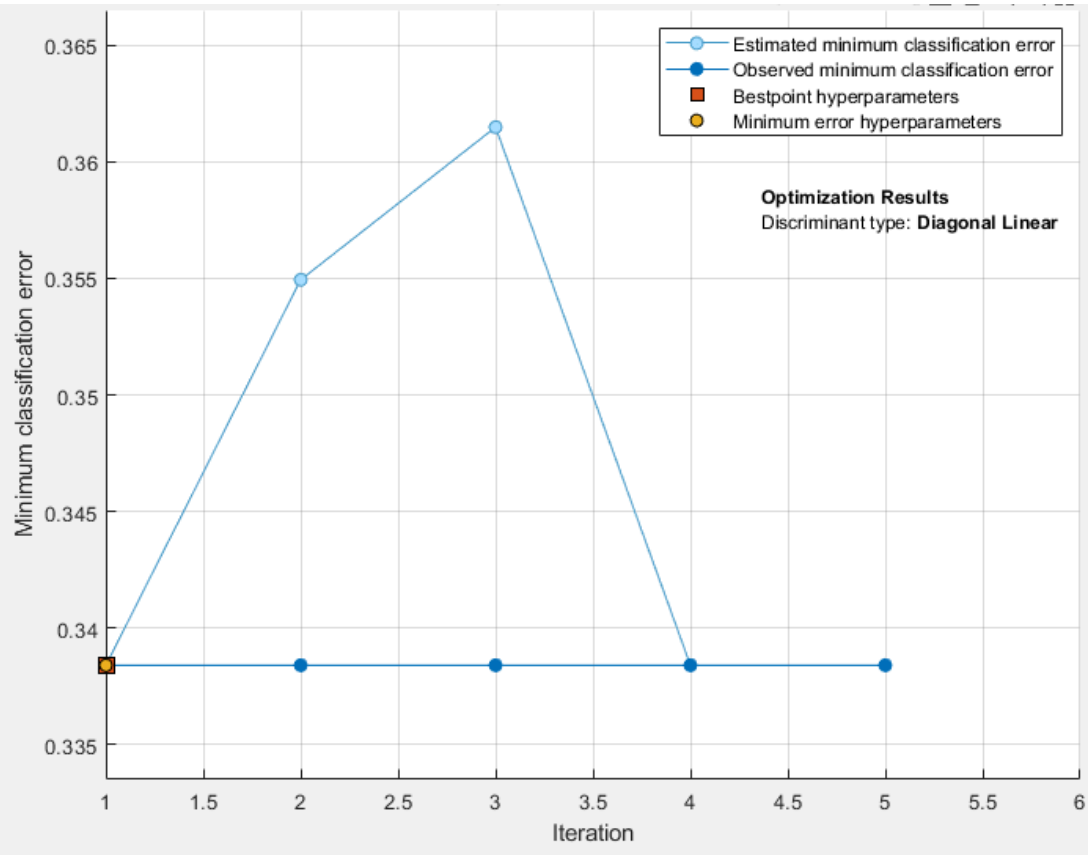
Anteriormente se había hallado que el lineal era el que mejores resultados presentaba, esto se comprueba/ reafirma al realizar este paso.

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



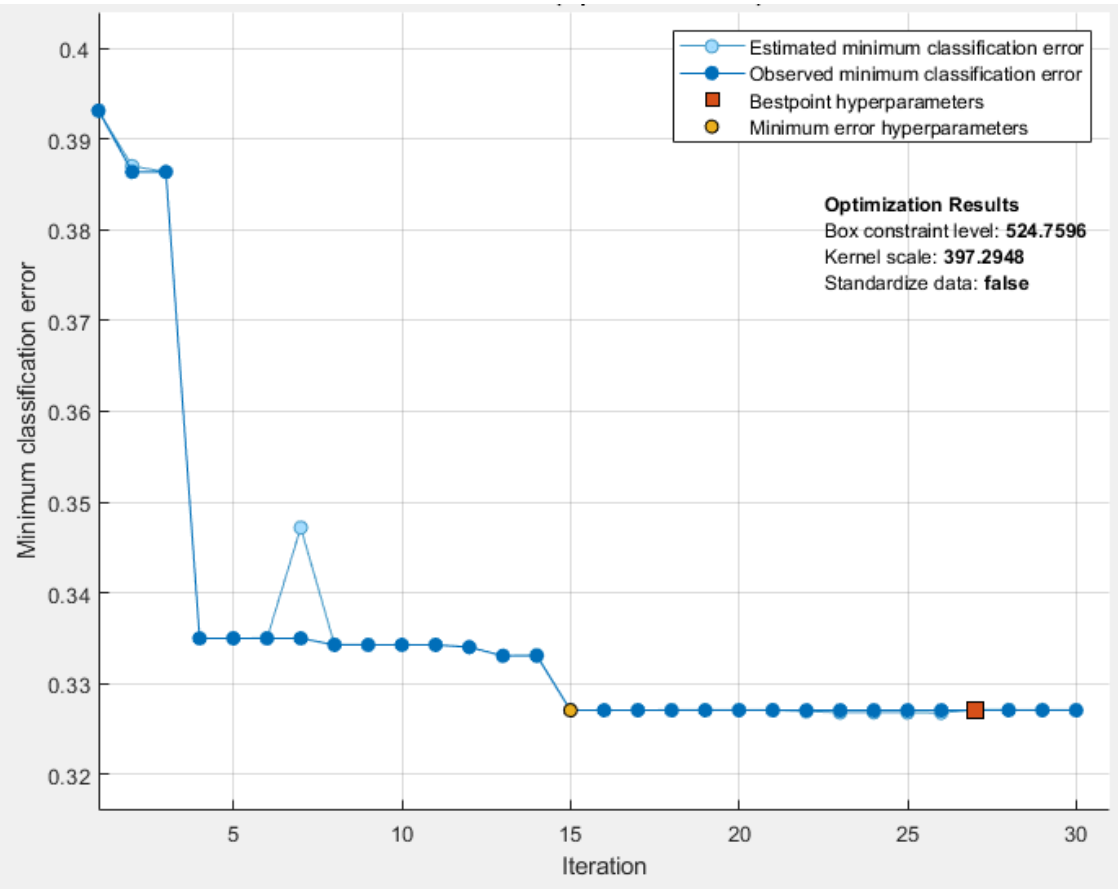
Discriminante (7 características):

Mejores resultados con diagonal linear

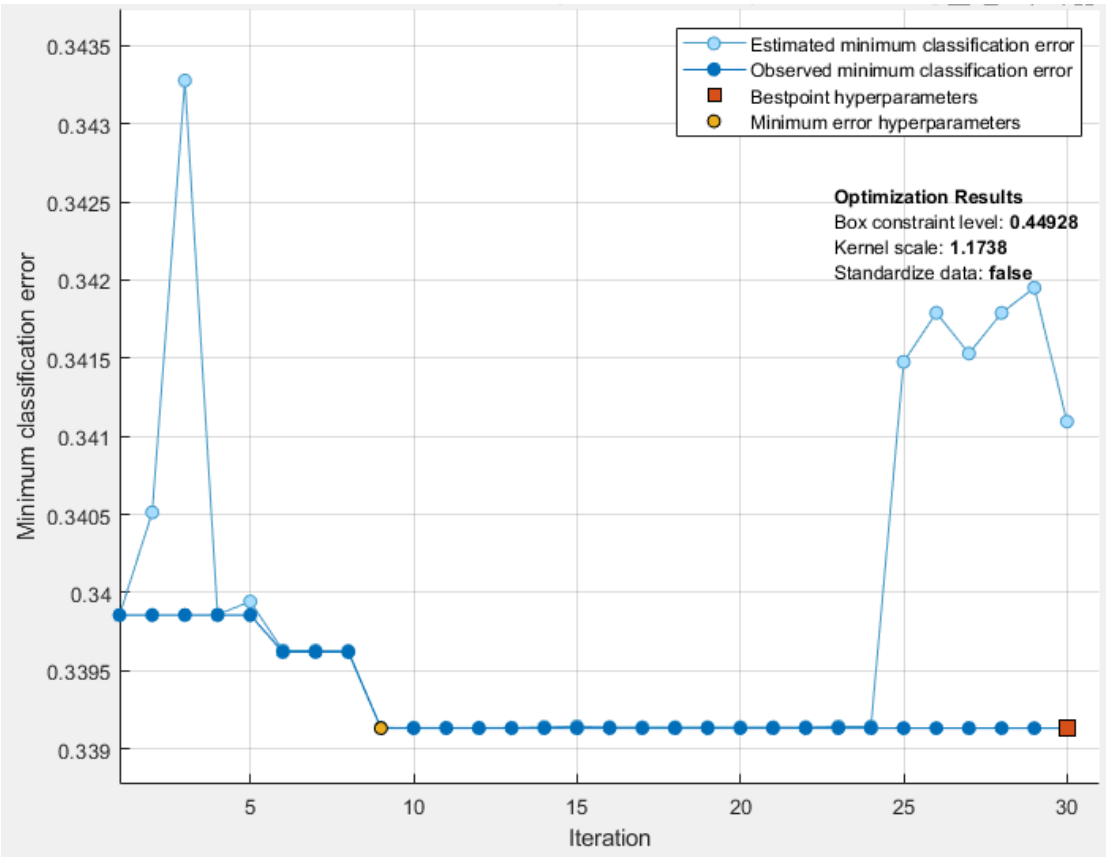


SVM:

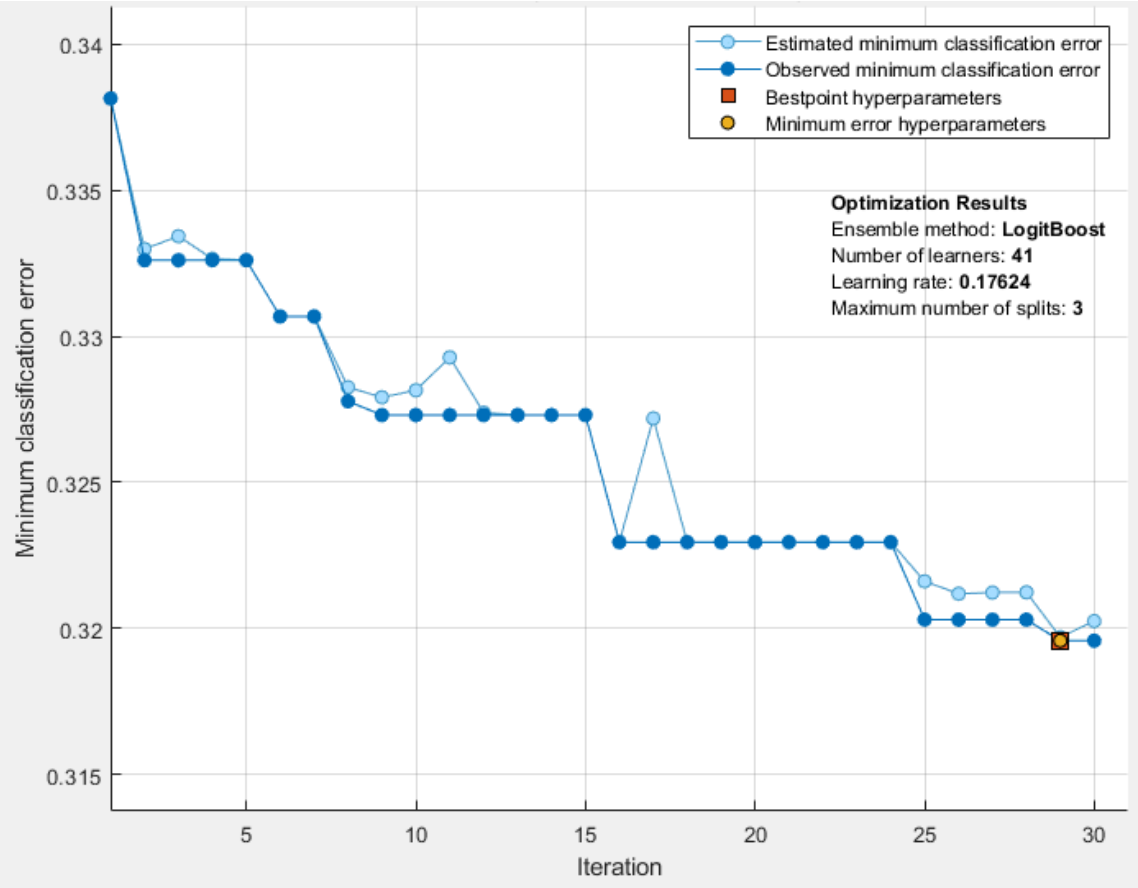
Para SVM los mejores resultados se obtuvieron para gaussiano "course" por lo que este será optimizado



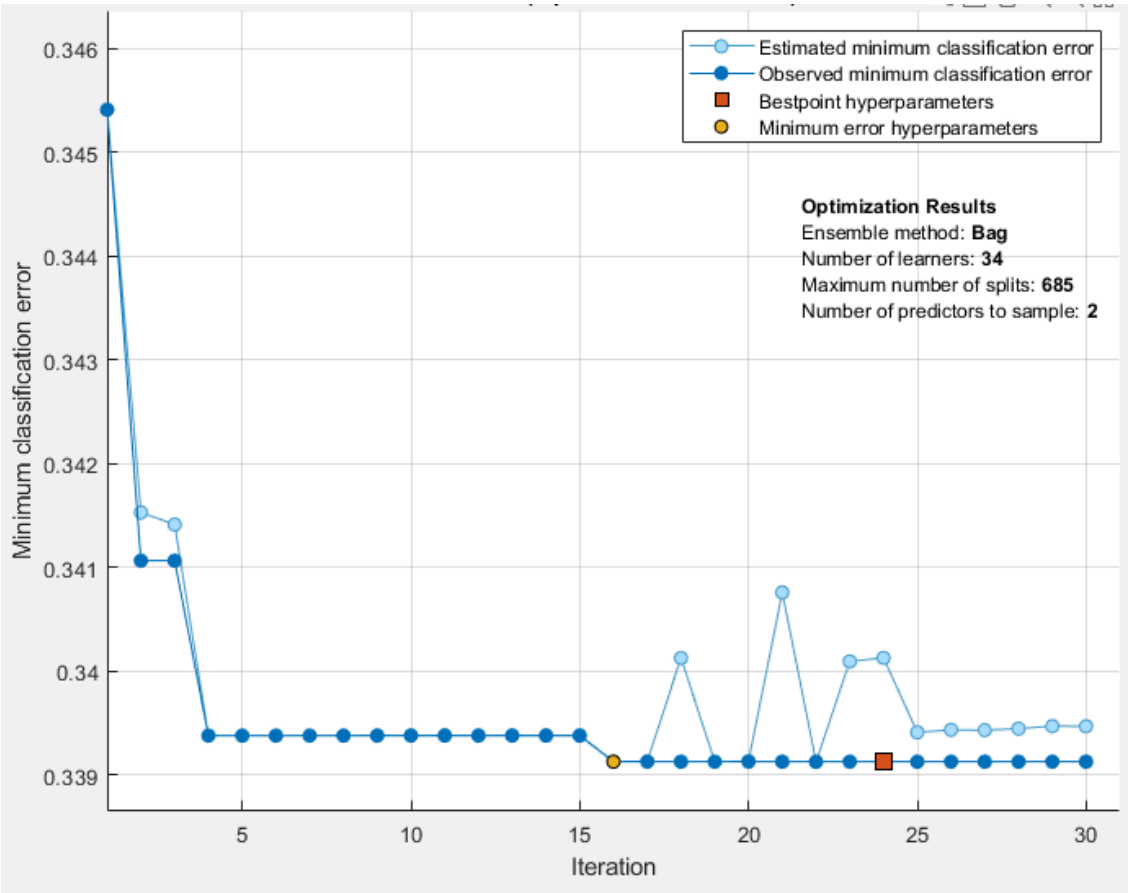
SVM (7 características):


Estimated minimum classification error
Observed minimum classification error
Bestpoint hyperparameters
Minimum error hyperparameters
Optimization Results
Box constraint level: 0.44928
Kernel scale: 1.1738
Standardize data: false
Minimum classification error
Iteration
0.3435
0.343
0.3425
0.342
0.3415
0.341
0.3405
0.34
0.3395
0.339
5
10
15
20
25
30

Ensemble:



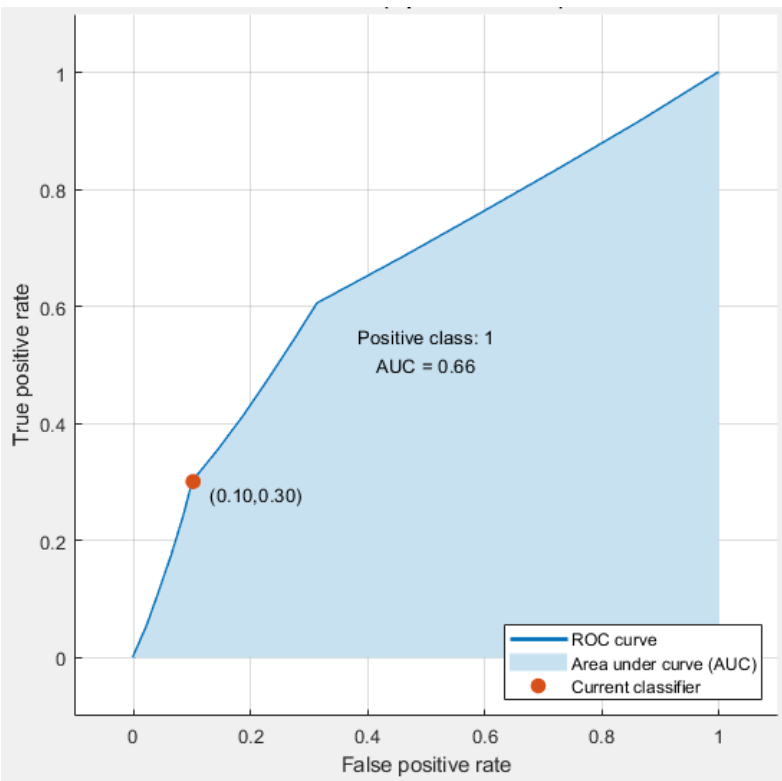
Ensemble (7 características):

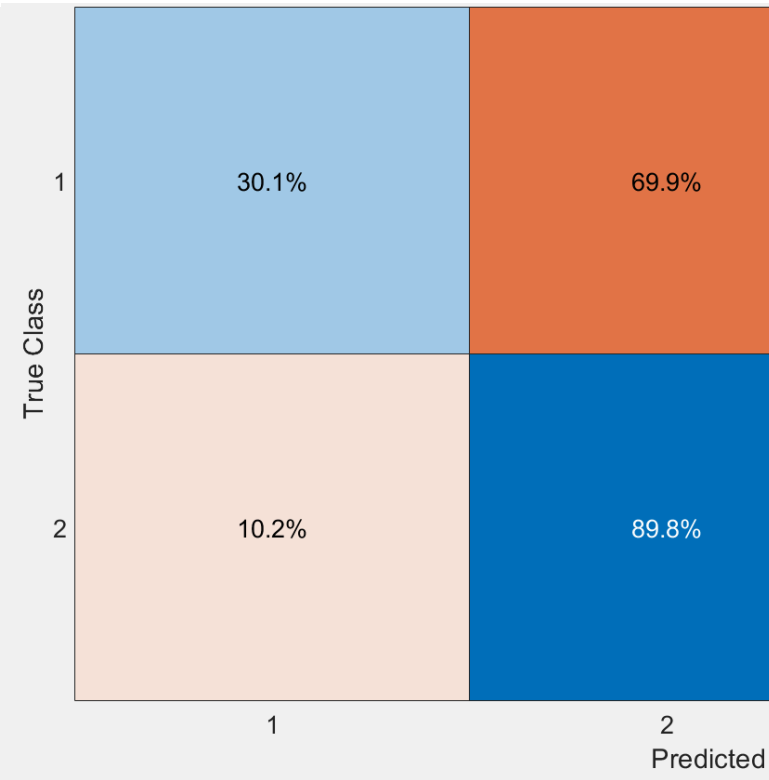


Presentación de resultados de los modelos entrenados con el data set completo y el de características reducidas

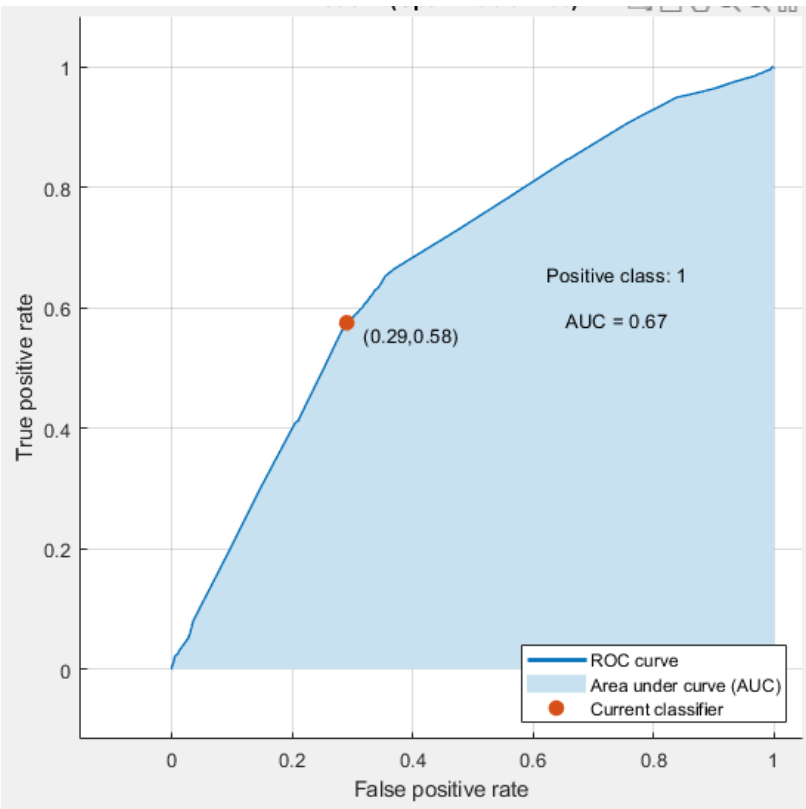
El código para entrenar los modelos se encuentra en la carpeta "models".

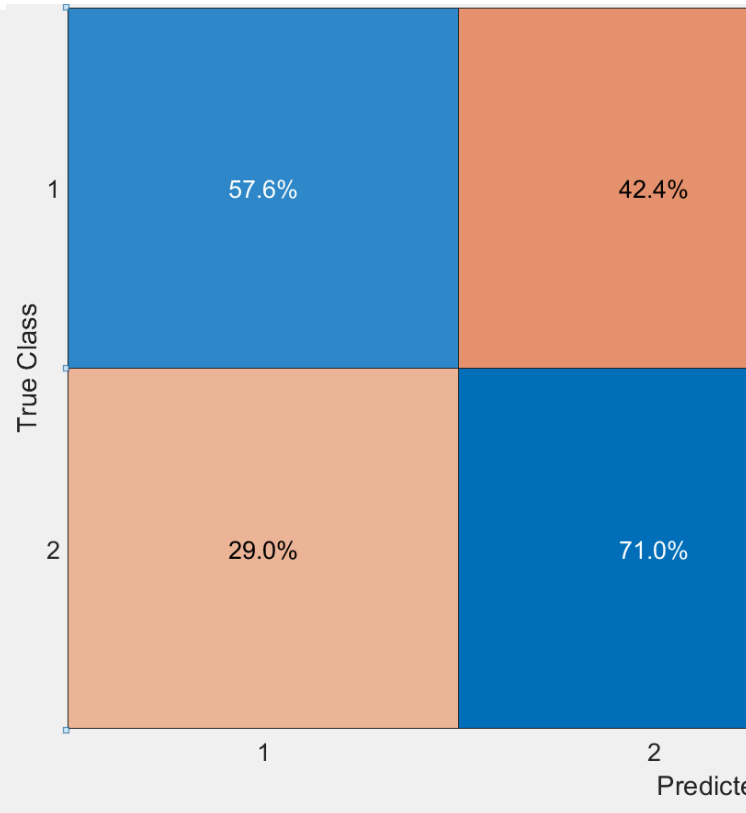
Arboles de decisión:



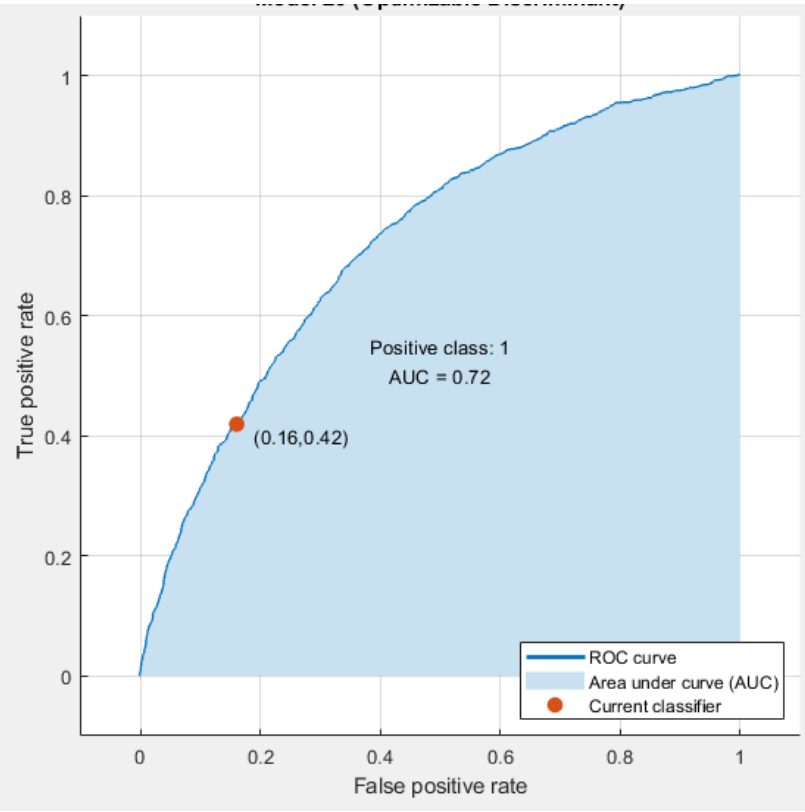


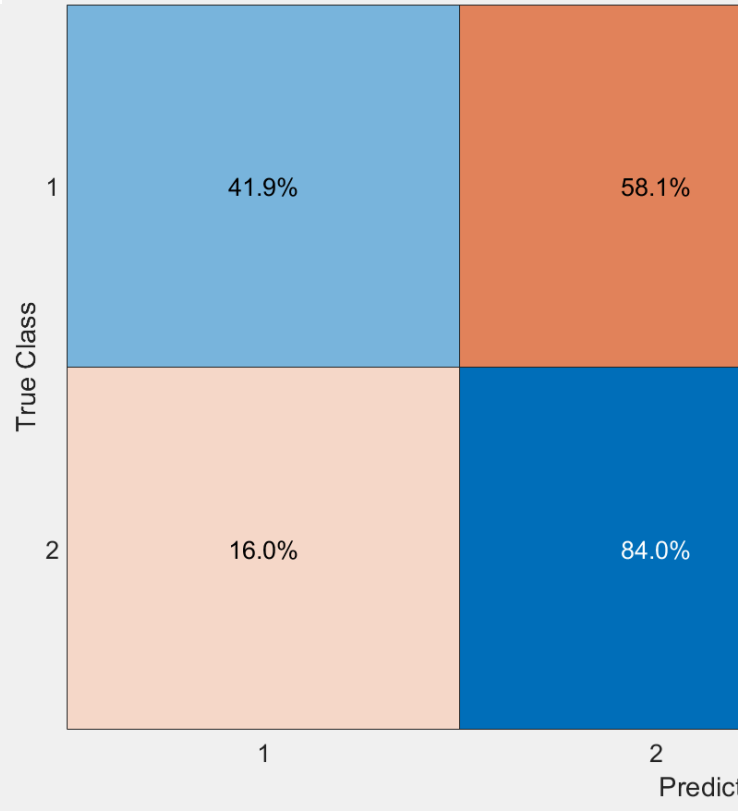
Arboles de decisión (7 características):



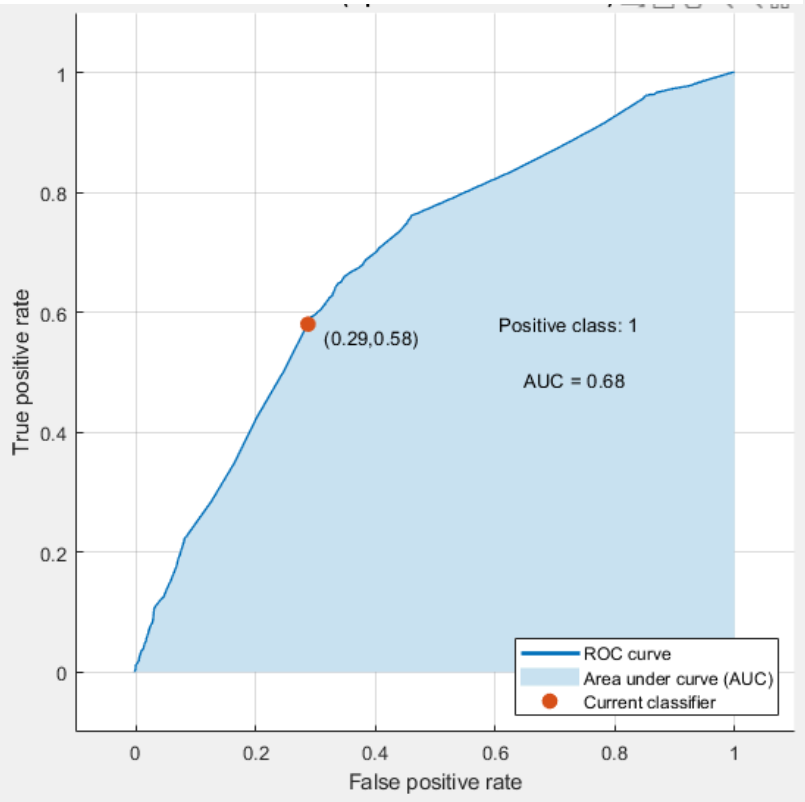


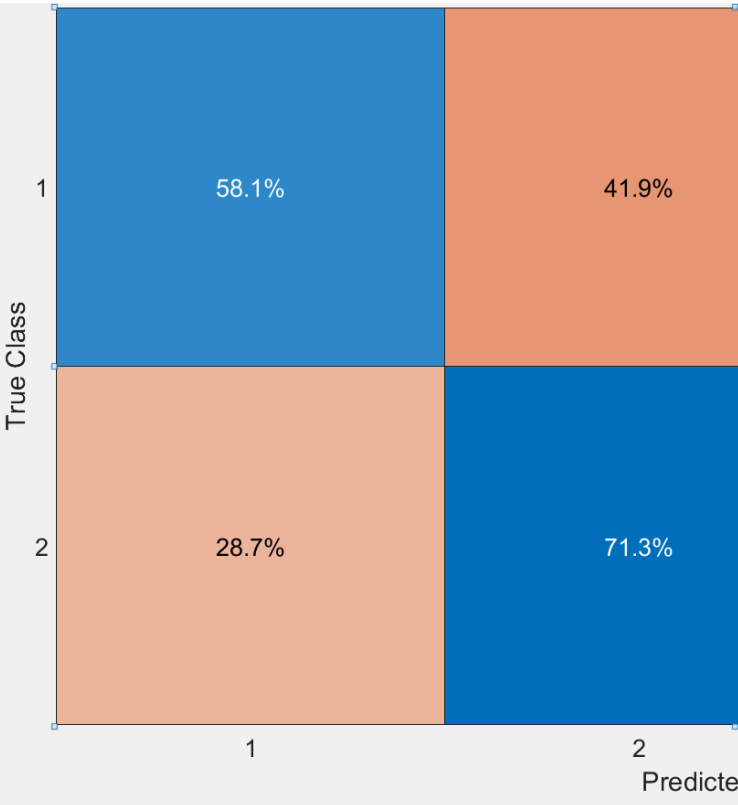
Discriminante:



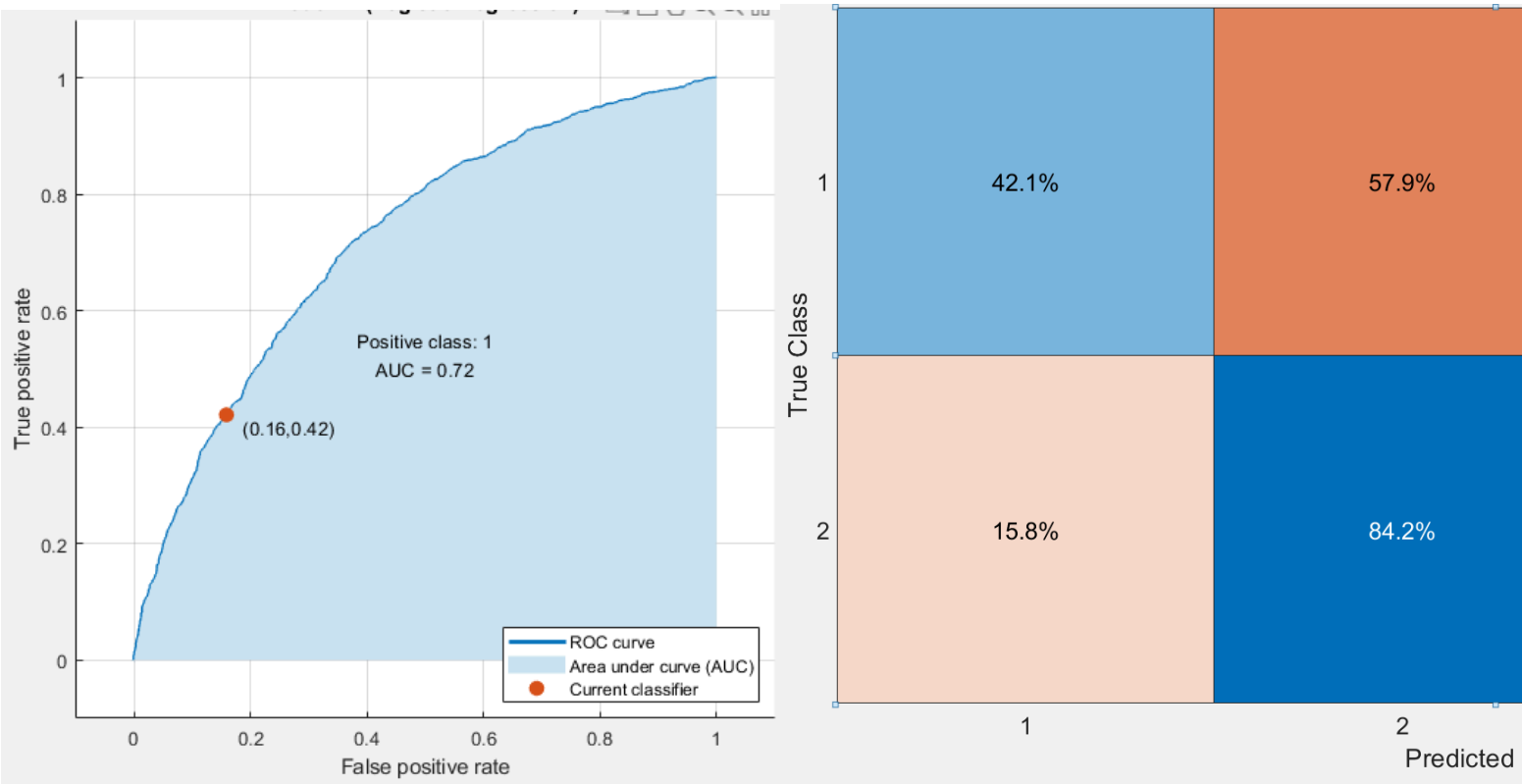


Discriminante (7 características):

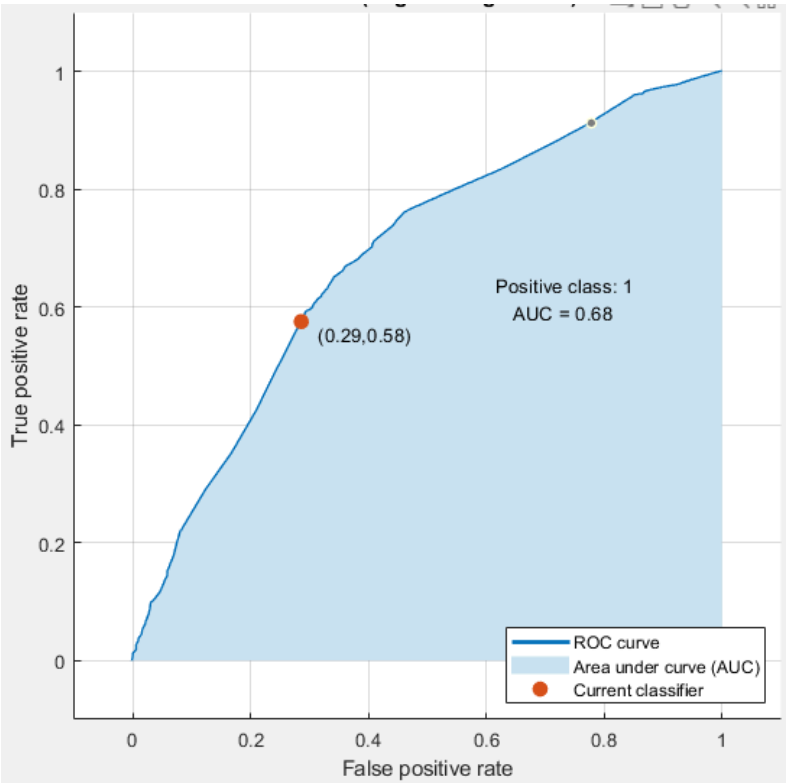


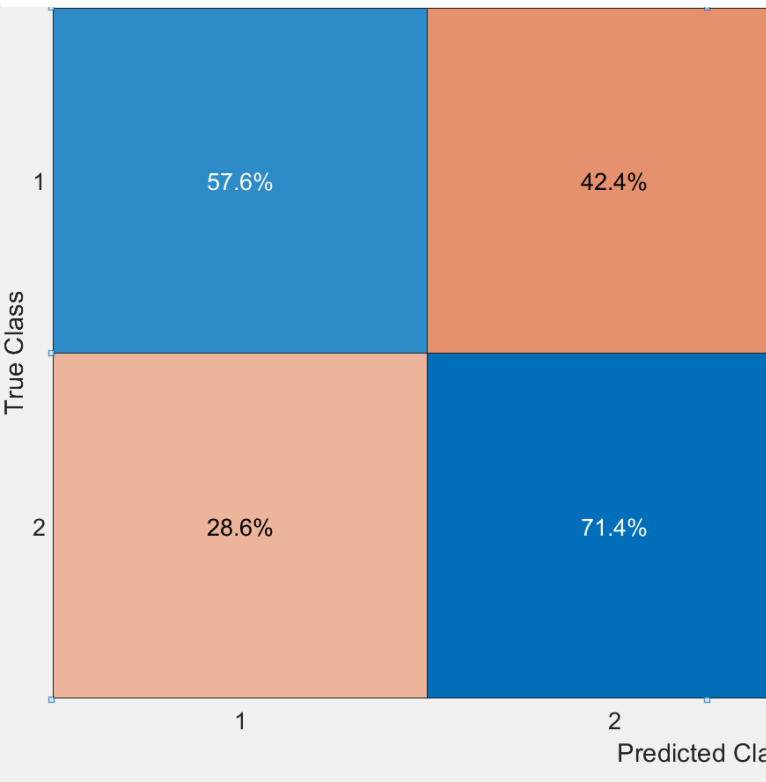


Logistic regression:

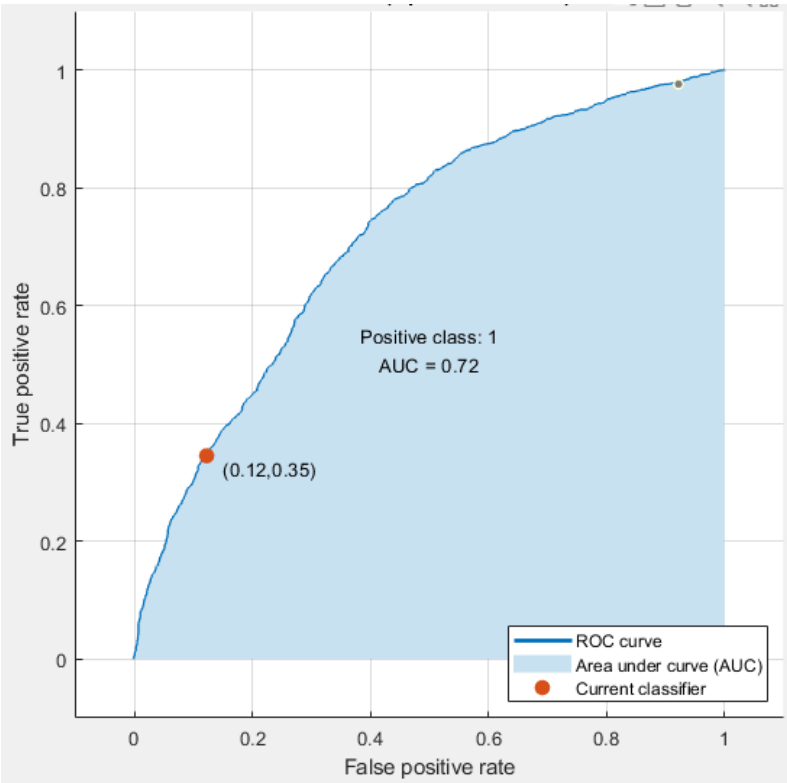


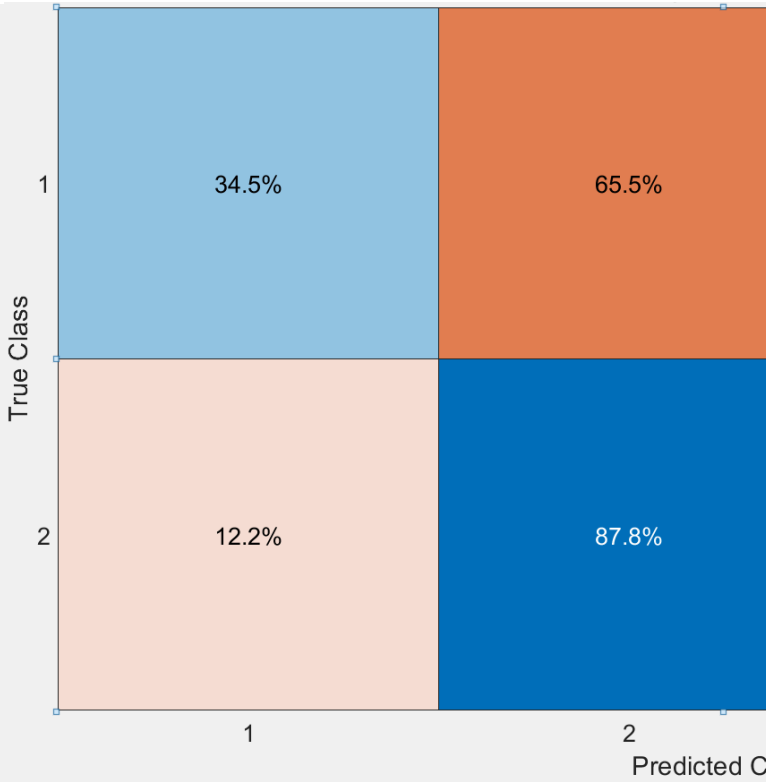
Logistic regression (7 características):


True positive rate
False positive rate
Positive class: 1
AUC = 0.68
(0.29,0.58)
ROC curve
Area under curve (AUC)
Current classifier

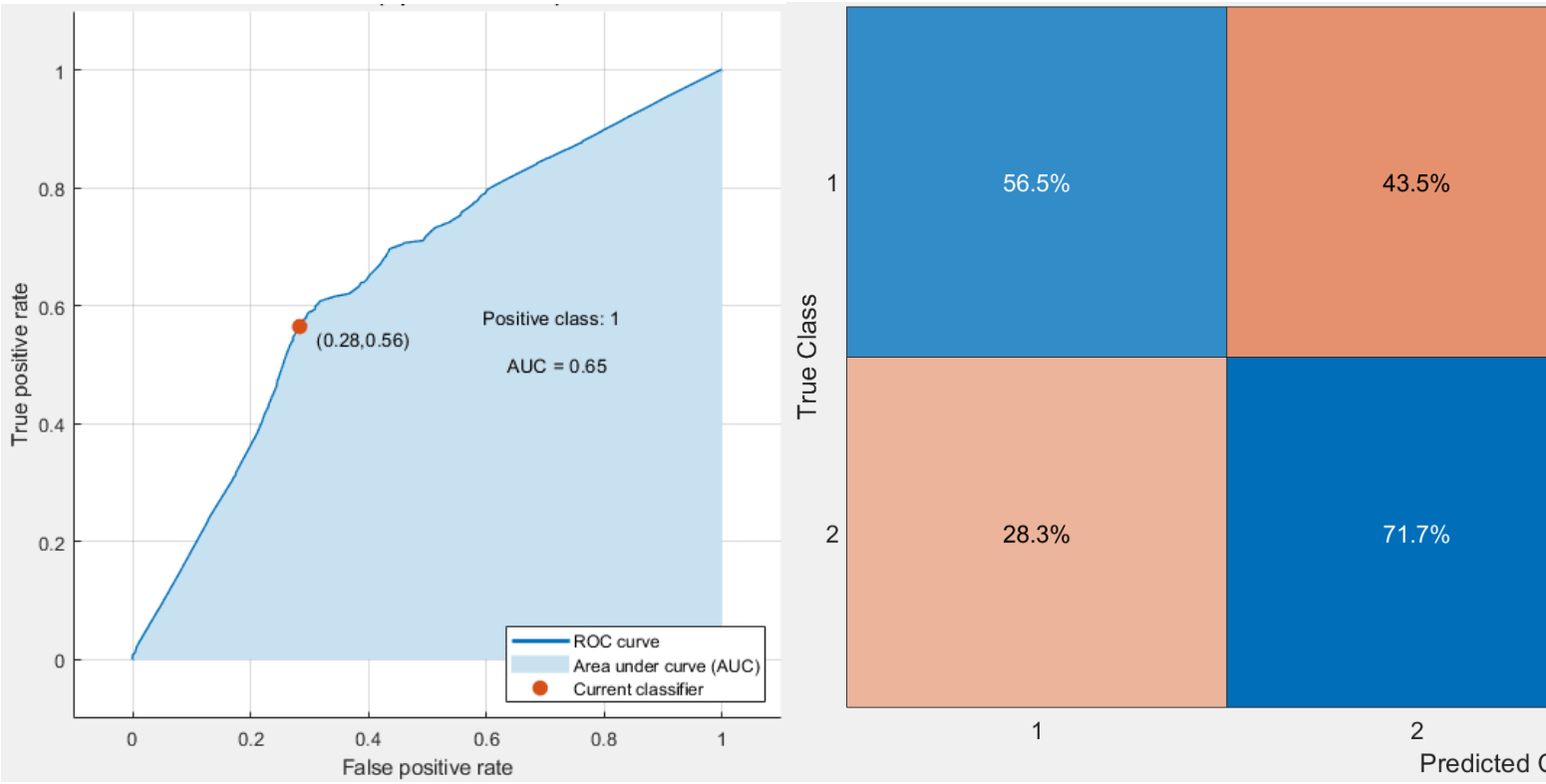

True Class
57.6%
42.4%
28.6%
71.4%
1
2
Predicted Cla

SVM:

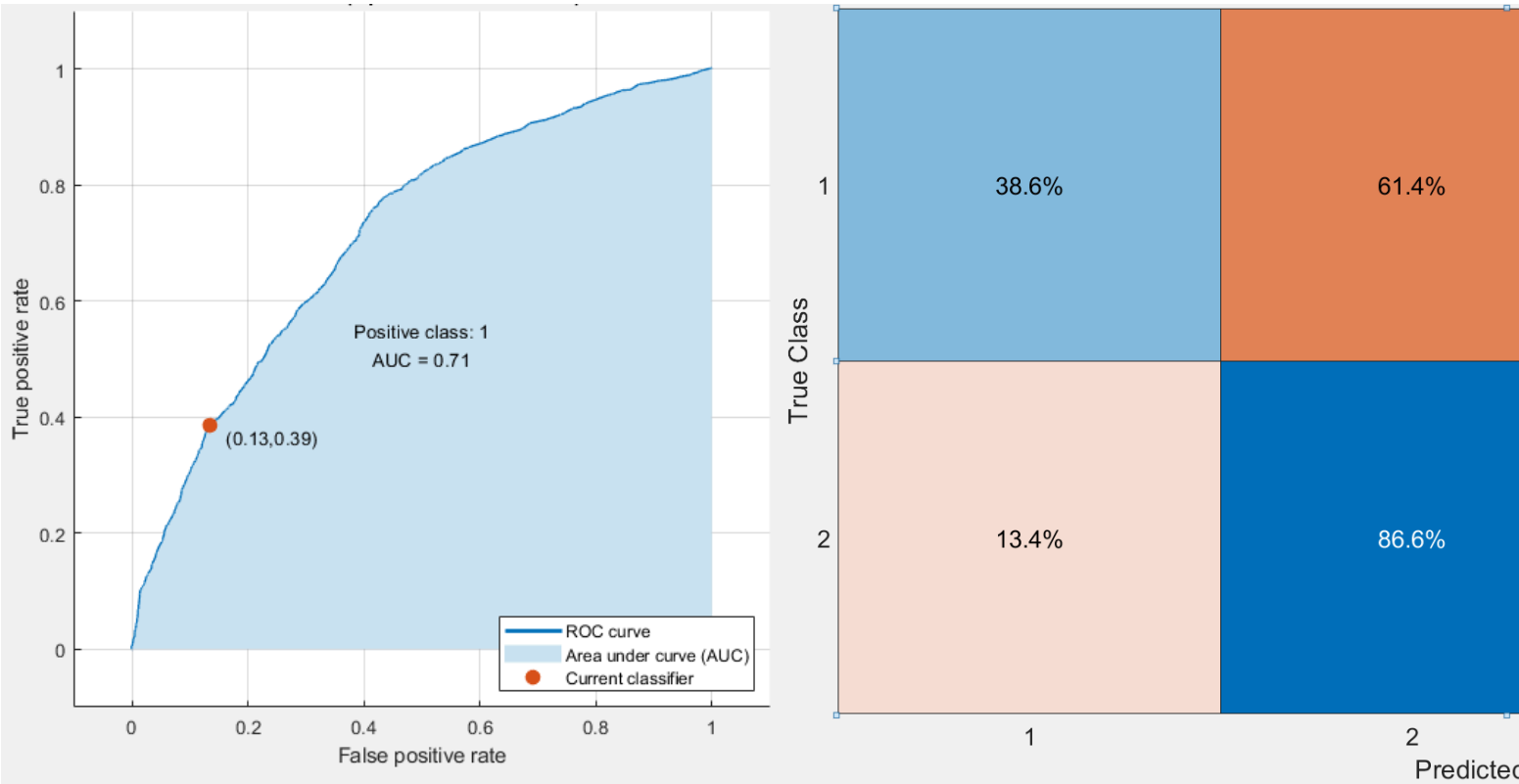

True positive rate
False positive rate
Positive class: 1
AUC = 0.72
(0.12,0.35)
ROC curve
Area under curve (AUC)
Current classifier


True Class
34.5%
65.5%
12.2%
87.8%
1
2
Predicted C

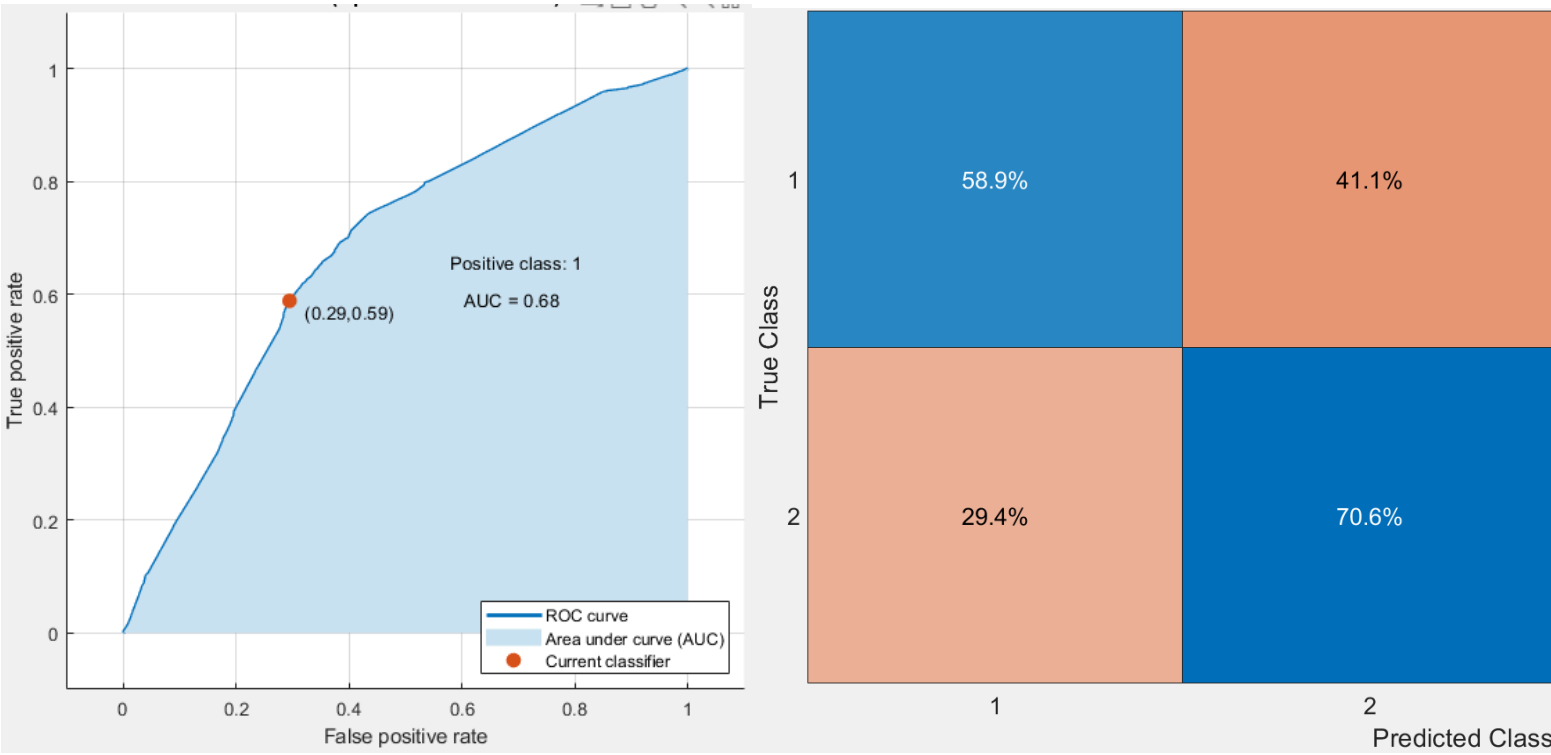
SVM (7 características):



Ensemble:



Ensemble (7 características):



Comparación de modelos

Para este punto se tendrán en cuenta varias cosas:

- La precisión y exactitud del modelo, mientras mayor mejor, sin llegar a un caso de sobreajuste.
- El número de parámetros en general es de interés obtener modelos que con un bajo número de parámetros sean capaces de cumplir con su objetivo a cabalidad, esto debido a que en un caso real es más difícil y costoso, en términos de dinero y tiempo obtener una cantidad grande de informaciones este caso no se les dará mayor importancia a unos parámetros sobre otros, solo será de interés el número de ellos.
- La complejidad del modelo, se preferirán modelos más simples
- Un último factor que se tendrá en cuenta para preferir un modelo sobre otro es la distribución de falsos negativos hacia cierta clase particular, i.e. en este contexto no sería nada bueno identificar erróneamente a aquellas personas con tendencia repetitiva al intento de suicidio, mientras que identificar erróneamente a aquellos que en realidad no (falso positivo), seria más aceptable.

`Provisional:`

`Teniendo en cuenta lo anterior lo que se busca es maximizar la predicción correcta de la etiqueta 1 {si a intentos previos de suicidio}, en este sentido la mayoría de los modelos son deficientes. Pero en general los de 7 características se comportan mejor que los completos. Así, uno de los mejores sería el de discriminante linear con 7 características`

ya que es el que más se acerca a lo requerido, tienen poco número de características y no hace parte de los modelos demasiado complejos

Pendiente Escobar:

- **calcular score del modelo,**
- **poner análisis general de ROC y CM**
- **Hacer eso de la predicción probabilística y determinista**

Conclusiones

El chorro:

- *Hablar de porque nuestros modelos son tan malos (posibles razones: ¿se hicieron las cosas mal? xd. ¿Características comunes para aquellos que son reincidentes por lo que es difícil separarlos de aquello que no? Preferiblemente una mayor cantidad de datos (y también mejor calidad ya que había muchos datos faltantes))*
- *Hay que comentar que podría significar según el contexto esas 7 características más importantes*
- *Como seria excelente de acuerdo con la propuesta inicial poder conseguir datasets con información de personas que han intentado previamente el suicidio como de aquella que no*

Referencias (arreglar)

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


```
%test = readtable('suicidio_null.csv'); size_test = size(test) %Tes con el
%data set original sin procesar
```

Notas

Modelos sugeridos por el profesor: Regresion logistica, SVM, Arboles de decision, Redes neuronales, LMP, Random Forest

Intent_prev{1 = SI; 2 = NO}

*Para traducir graficas{**Mínimo error de clasificación,Iteración,Mínimo error de clasificación observado,Hiperparámetros de error mínimo**}*