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

Avance 2: Entrenamiento, adecuacion y evaluacion de modelos

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

En un primer momento se probaran diferentes modelos , haciendo uso de la herramienta "Classification Learner" de Matlab, debido a su facilidad y rapidez para probar multiples modelos simultaneamente. Se tomaran los pares dataset-modelo que mejores resultados den (preferiblemente por encima de 70% de acierto) para seguirlos desarrollando, en terminos de seleccion de parametros y optimizacion de hiperparametros.

Para la seleccion de parametros y ajuste de hiperparametros, en donde sea posible se usaran herramientas las interactivas o automaticas que provee Matlab.

Como metodos de validacion y calificacion de los modelos se prentenden usar los dados a continuacion ***(To Do: añadir breve descripcion de cada uno)***

- Score?
- Matriz de confusion
- ROC curve

Al momento de realizar predicciones se generaran dos, una deterrministica y otra probabilistica.

Data sets de entrada.

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

- cds_imputed : dataset con 33 carateristicas y 4146 registros
- cds : dataset con 28 caracteristicas 4146 registros,
- cds_few : dataset 33 caracteristicas y 655 registros
- cds_fem_minus_alcohol: dataset 32 caracteristicas 1690 registros.

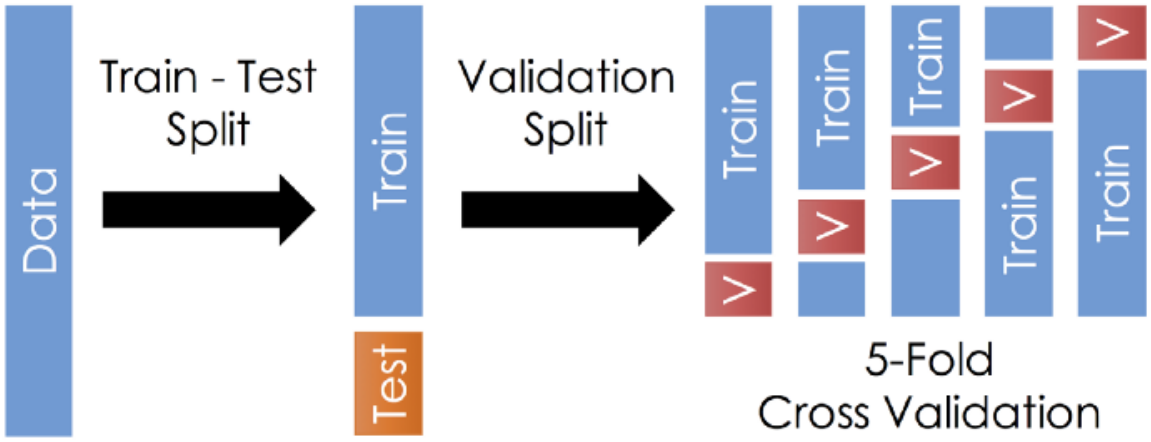
```
%cds = readtable('clean_datasets\cds.csv'); size_cds = size(cds)
cds_imputed = readtable('clean_datasets\cds_imputed.csv'); size_imputed = size(cds_imputed)
```

```
size_imputed = 1×2
        4146          34
```

```
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 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 cuanta 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

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

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

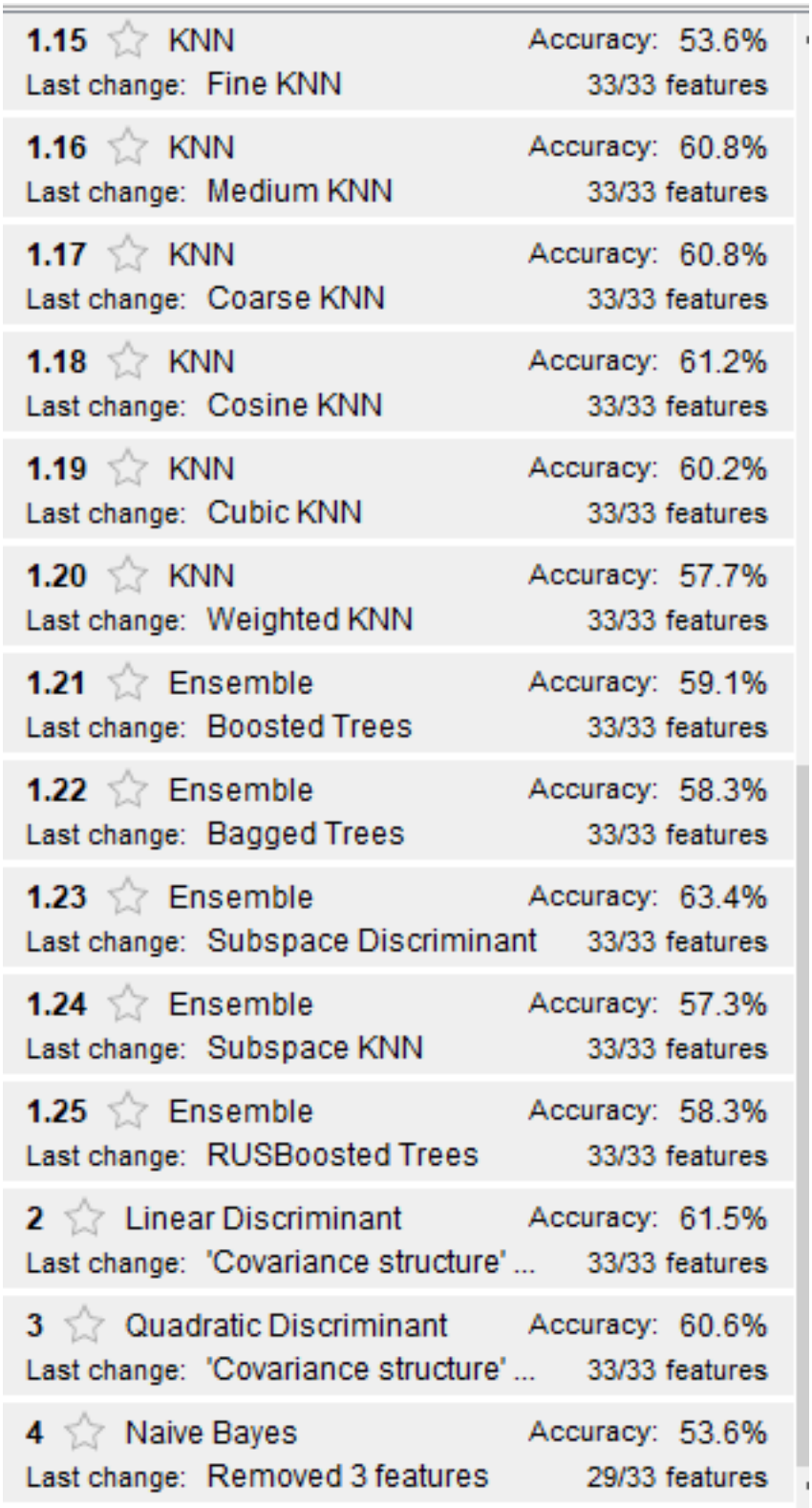
Resultados cds_imputed

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

Resultados cds_few

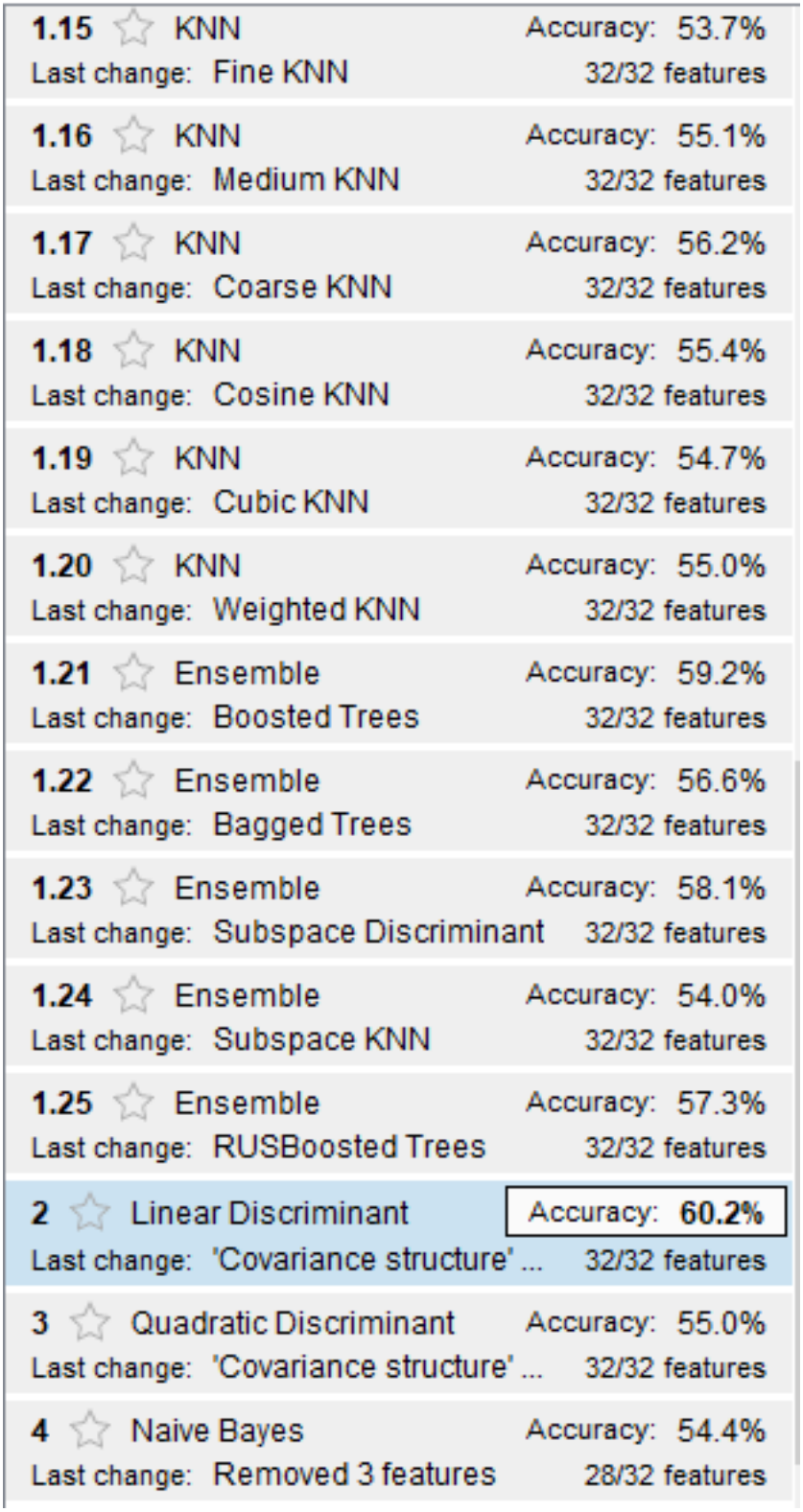
Para este dataset algunos modelos se hicieron individualmente, porque presentaban problemas con las caracteristicas 'antec_tran', 'tipo_ss_I', 'suici_fm_a' y 'tipo_SS_P ya que la mayoria o casi todos sus valores son iguales por lo que no aportan informacion o no presentan variacion con respecto a una de las clases por hallar.

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



Resultados cds_few_minus_alcohol

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



Por motivos exploratorios se realizaron pruebas aplicandole PCA a los datos, pero los resultadon 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, ningun par dataset-modelo obtuvo una precision mayor al 70% tal y como se habia definido inicialmente para su aceptacion. Por este motivo se tomara aquel dataset que produjo el modelo con la mayor precision(cds_imputed) y los mejores modelos obtenidos a partir de este -Coarse Tree, Linear discriminant, Logistic regresion , SVM (linear y coarse) y Ensamble(BoostTrees, SubsD)-

Feature selection

Bucando reducir la dimensionalidad y explorar direferente opciones se pretende realizar un proceso de seleccion de caracteriscas. Esto se hara filtrando aquellas caracteristicas 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 entrenaran 2 modelos, uno con todas las caracteristicas y adicionalmente otro con el conjunto de las 7 mas importantes

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

```
most_signif_features = 7×1 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 = 5×1 cell
'escolarid'
'esco_educ'
'tipo_ss_C'
'trab_socia'
'sexo_'
```

Optimizacion de hiperparametros

Para la optimizacion de hiperparametros seran utilizados dos enfoques:

Para los modelos simples(e.g. arboles de decision) se realizara mediante GridSearch, mientras que para los mas complejos(Ensamble, SVM) sera utilizado un metodo de optimizacion bayesiana, el cual, a travez de 30 iteraciones se va redirigiendo hacia aquellos hiperparametros del espacio de busqueda que proveen mejores resultados para el modelo. Este eleccion se hace debido a los costos computacionales elevados de realizar Grid o Random Serch en modelos complejos.

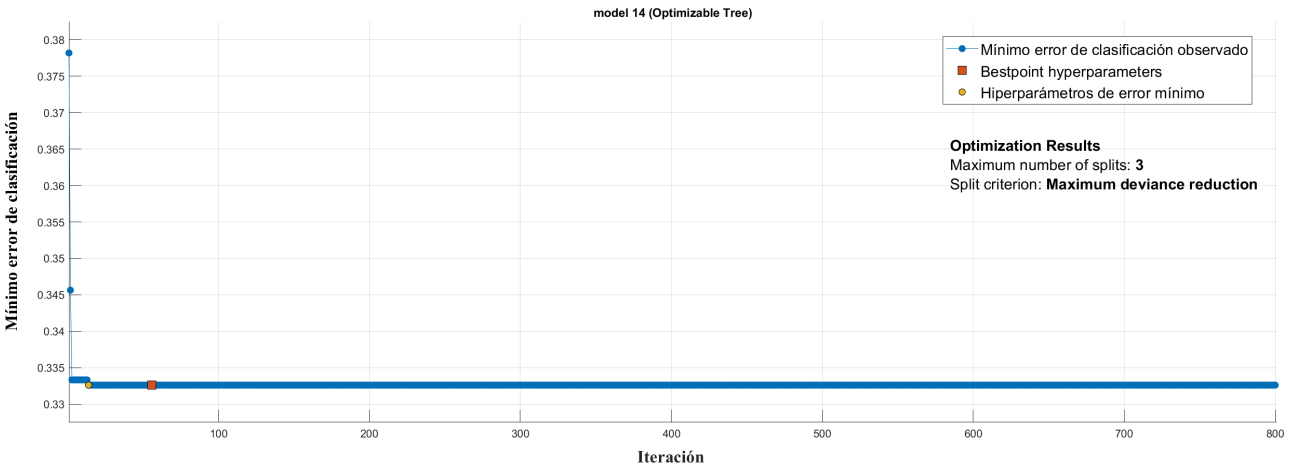
Como parte del proceso de optimizacion se obtiene el gráfico de error de clasificación mínimo en el cual se encuentran principalmente :

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

Los resultados se presentaran a continuacion para cada modelo:

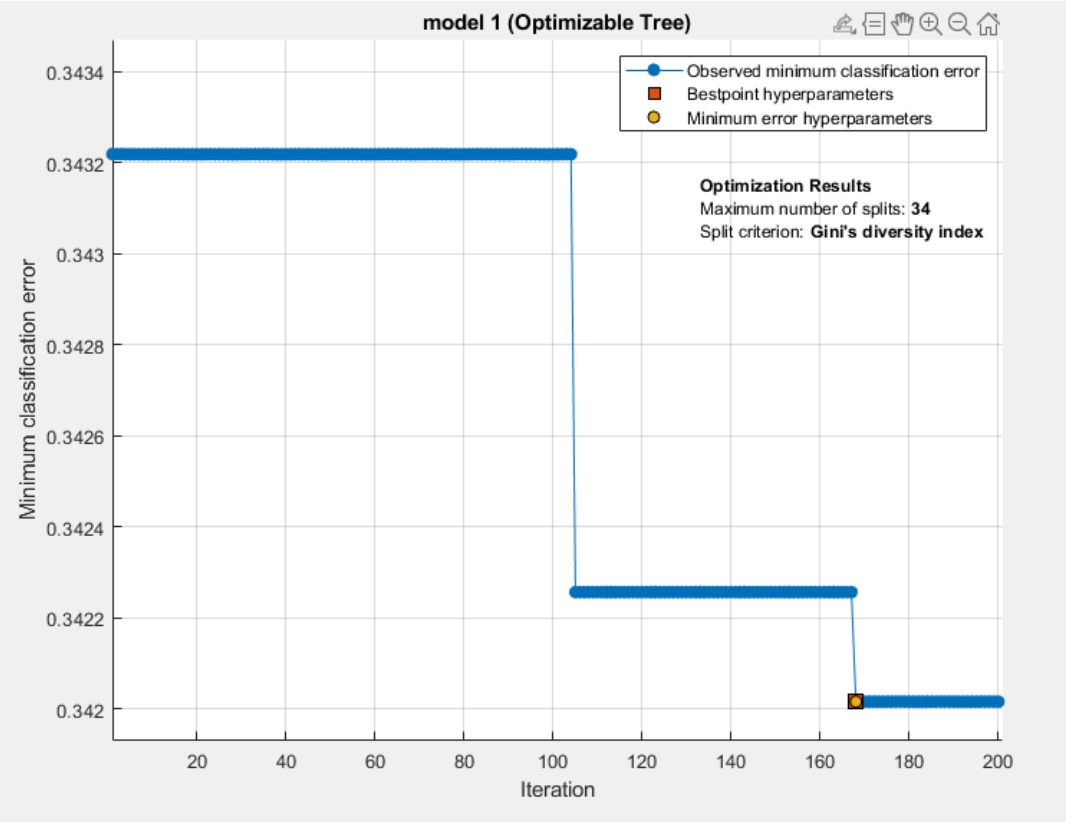
Arboles de decision:

En la anterior etapa mejores resultados fueron obtenidos con arboles de decision gruesos(con poco numero de splits), lo cual es confirmado con los resultados de este proceso



Arboles de decision(7 caracteristicas):

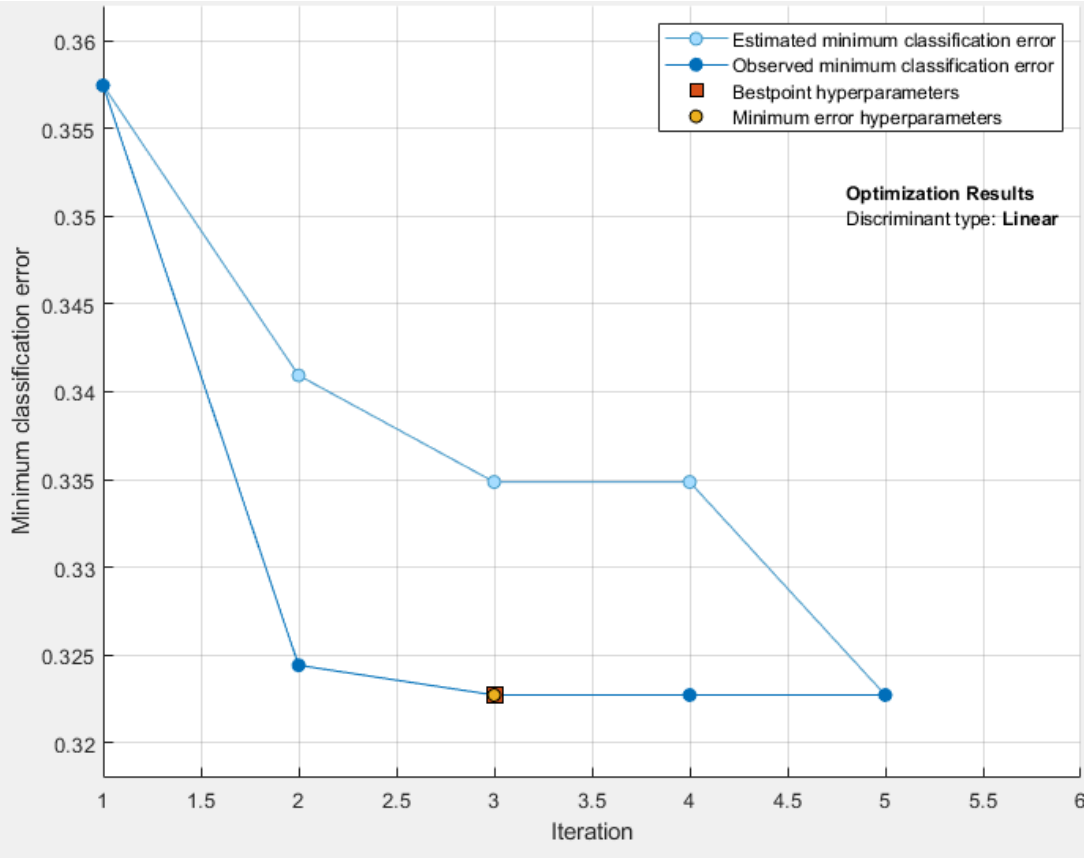
Para este caso los mejore resultados fueron con arboles finos



Discriminante:

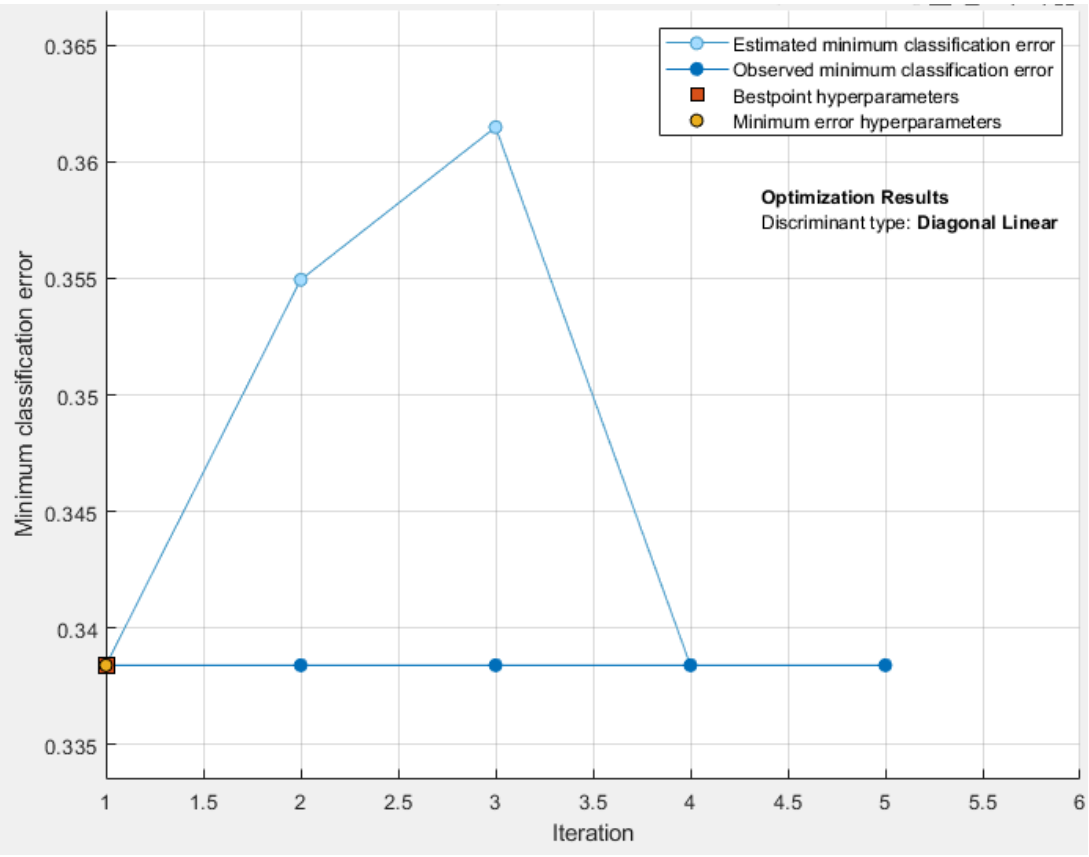
Anteriormente se habia hallado que el lineal era el que mejores resultados presentaba, esto se comprueba/ reafirma al raelizar este paso.

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



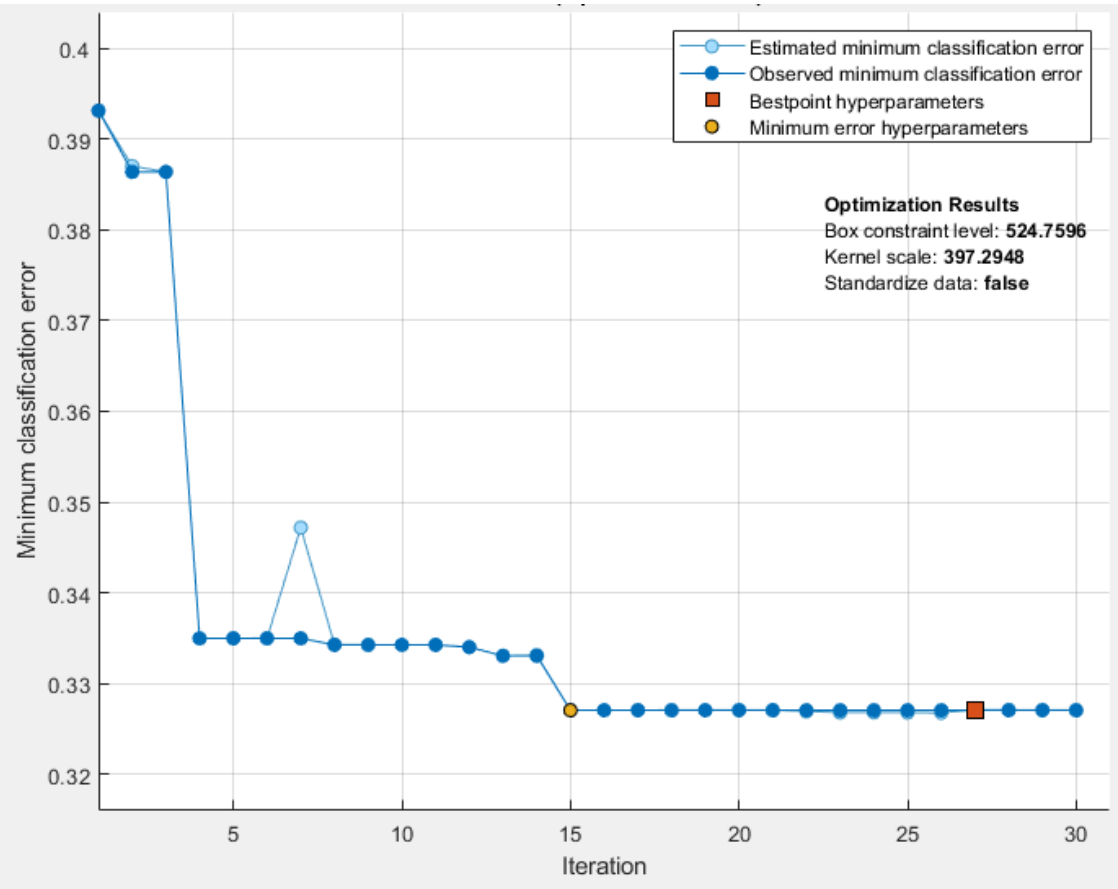
Discriminante(7 caracteristicas):

Mejores resultados con diagonal linear

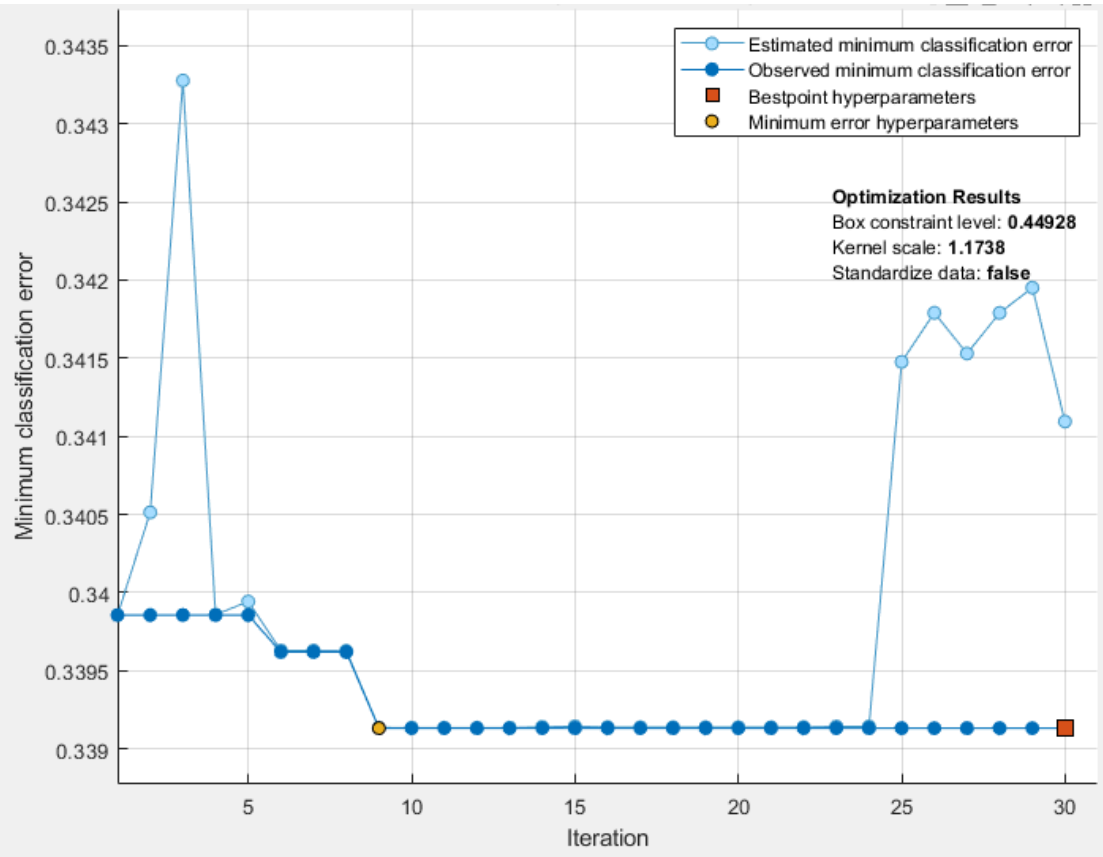


SVM:

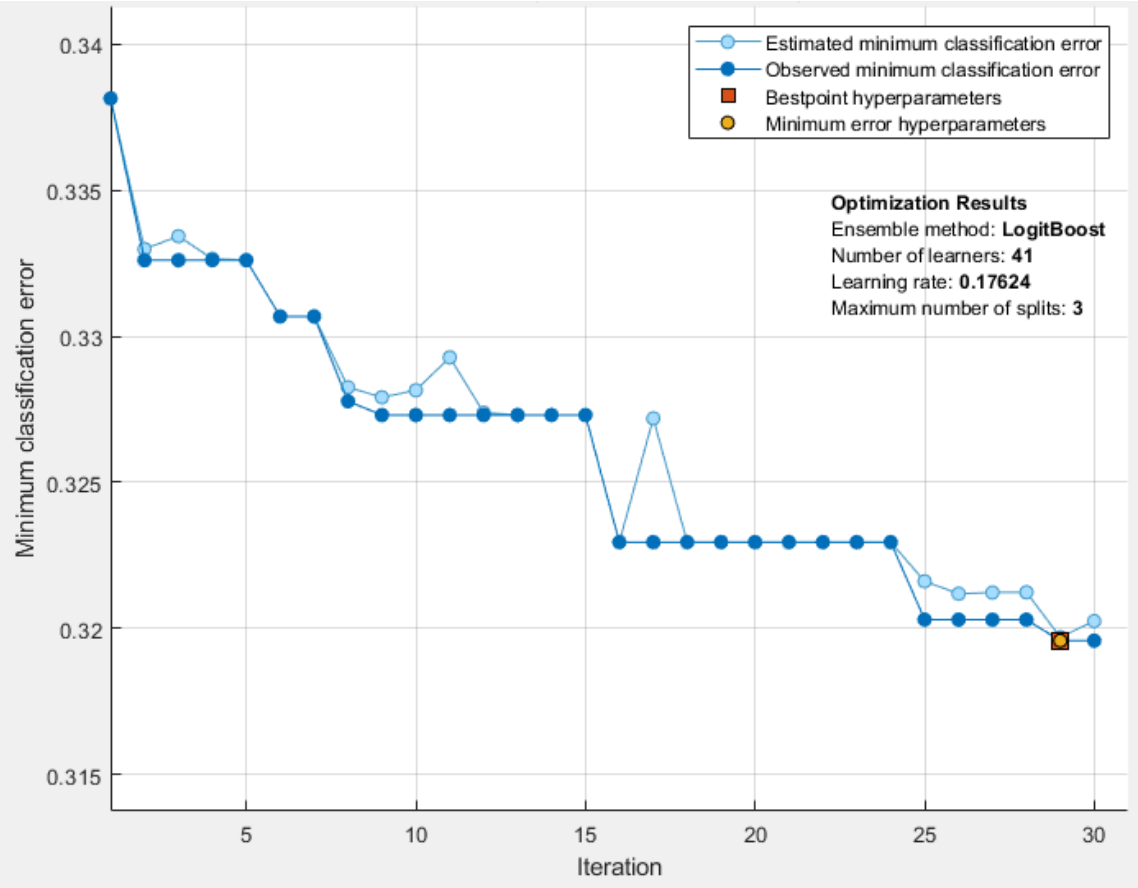
Para SVM los mejores resultados se obtuvieron para gaussiano "course" por lo que este sera optimizado



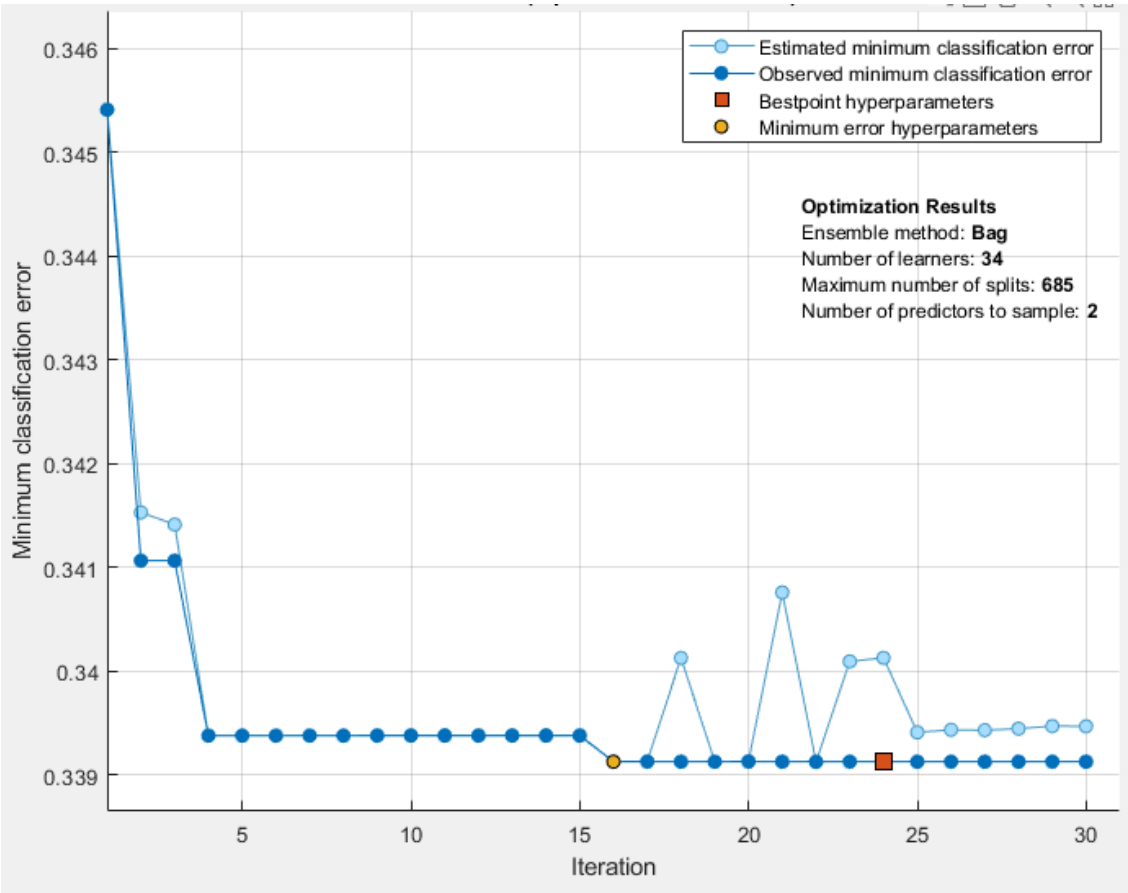
SVM(7 caracteristicas):


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:



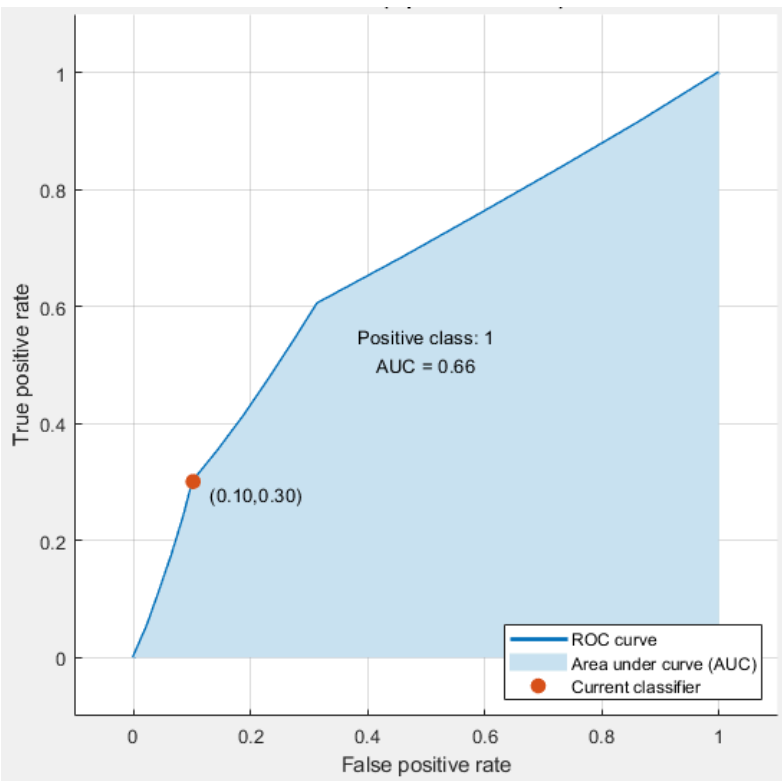
Ensemble (7 caracteristicas):

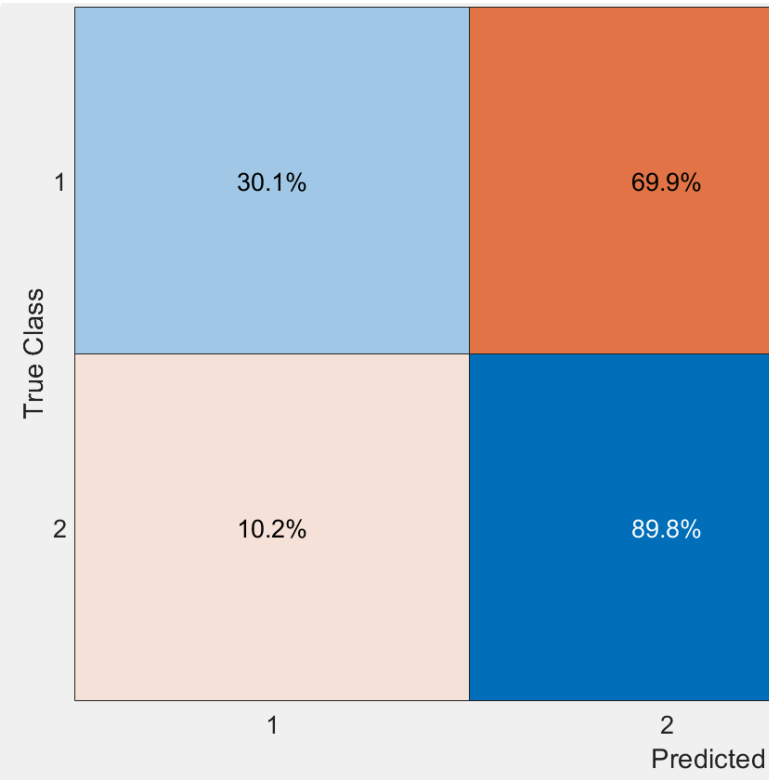


Presentacion de resultados de los modelos entrenados con el data set completo y el de caracteristicas reducidas

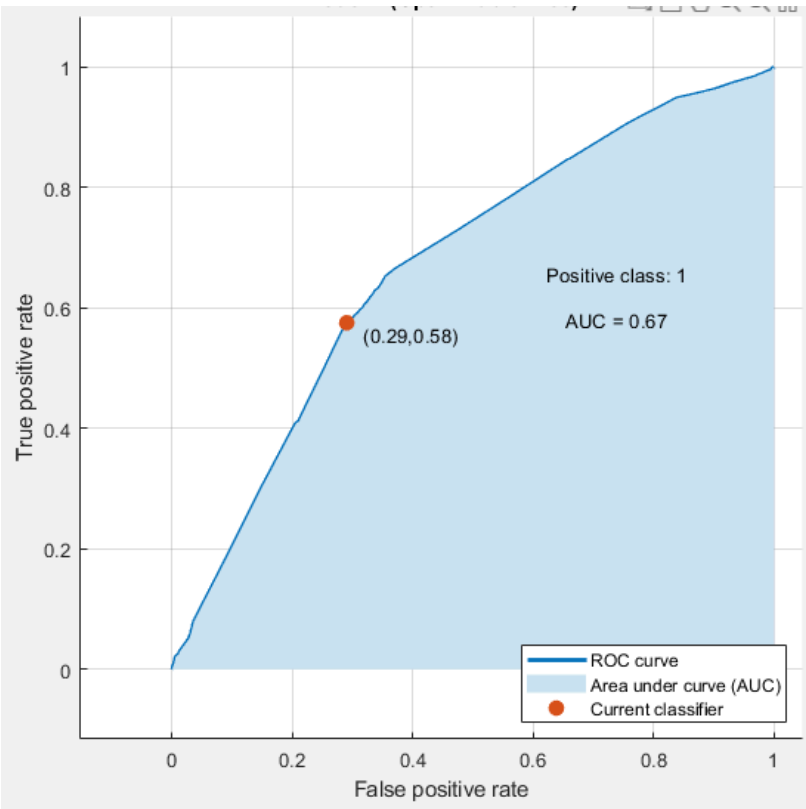
El codigo para entrenar los modelos se encuentra en la carpeta "models".

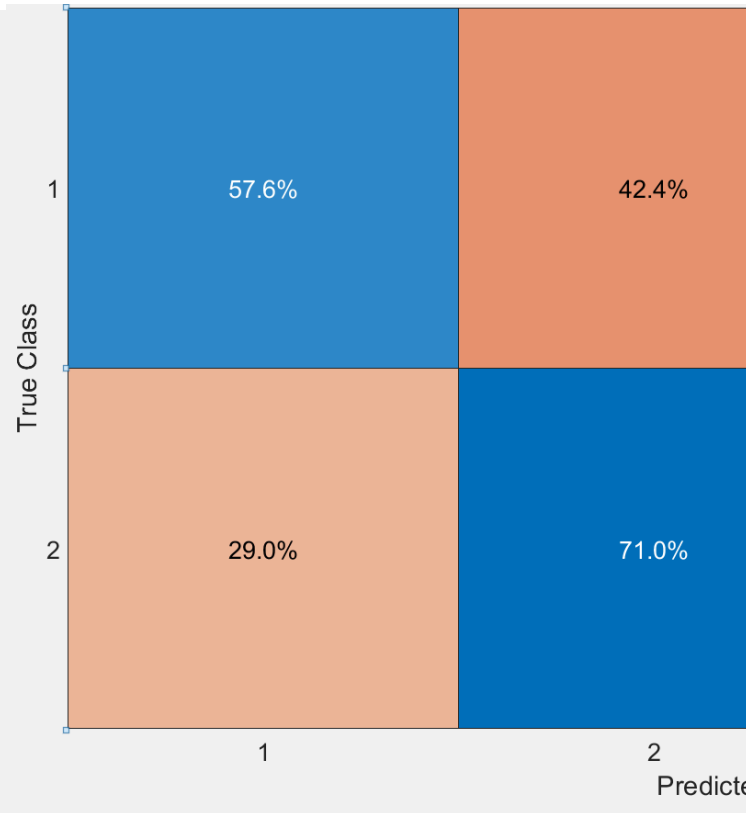
Arboles de decision:



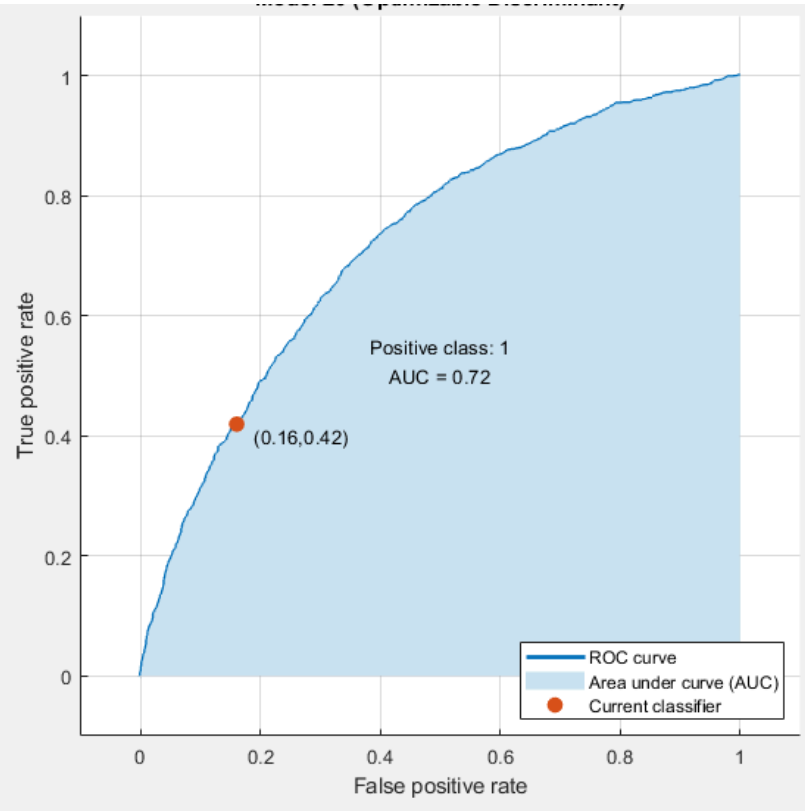


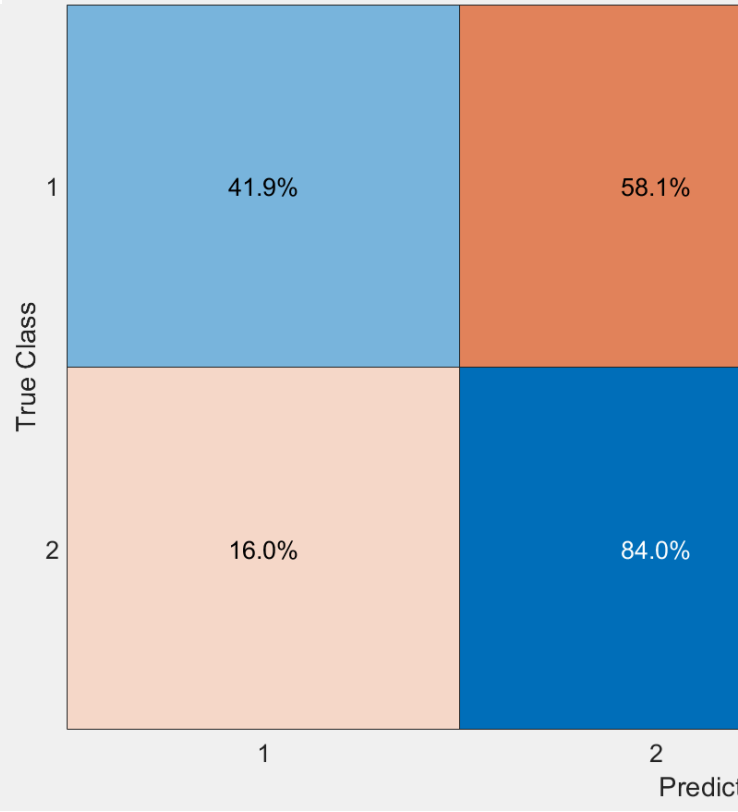
Arboles de decision(7 caracteristicas):



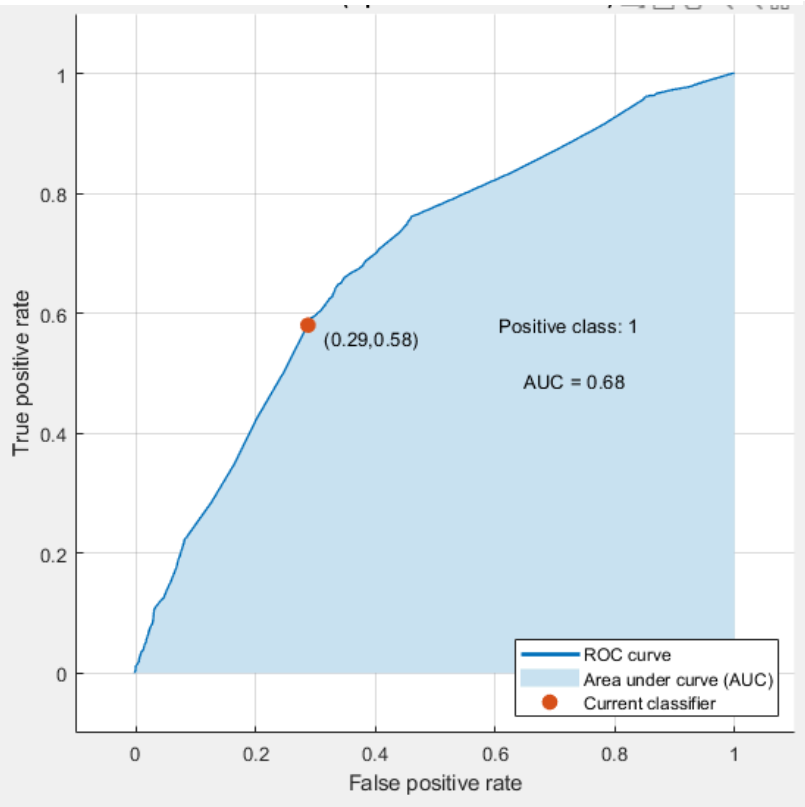


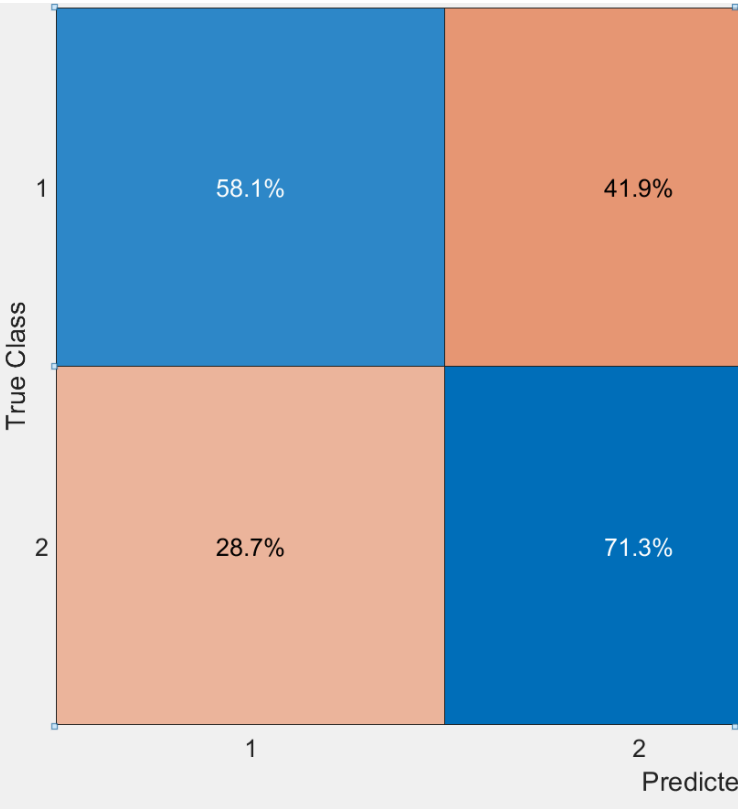
Discriminante:



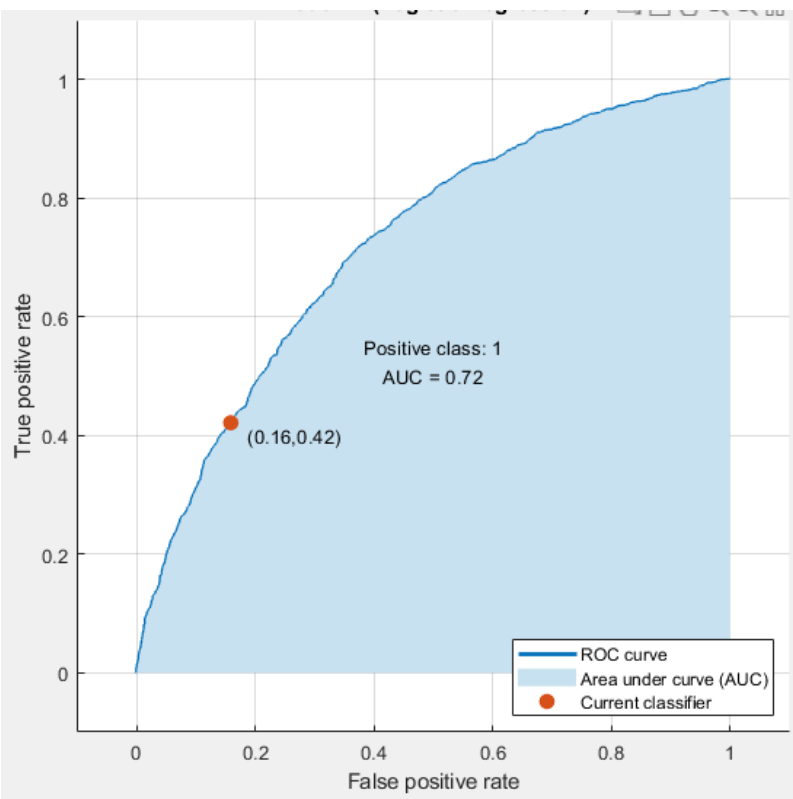


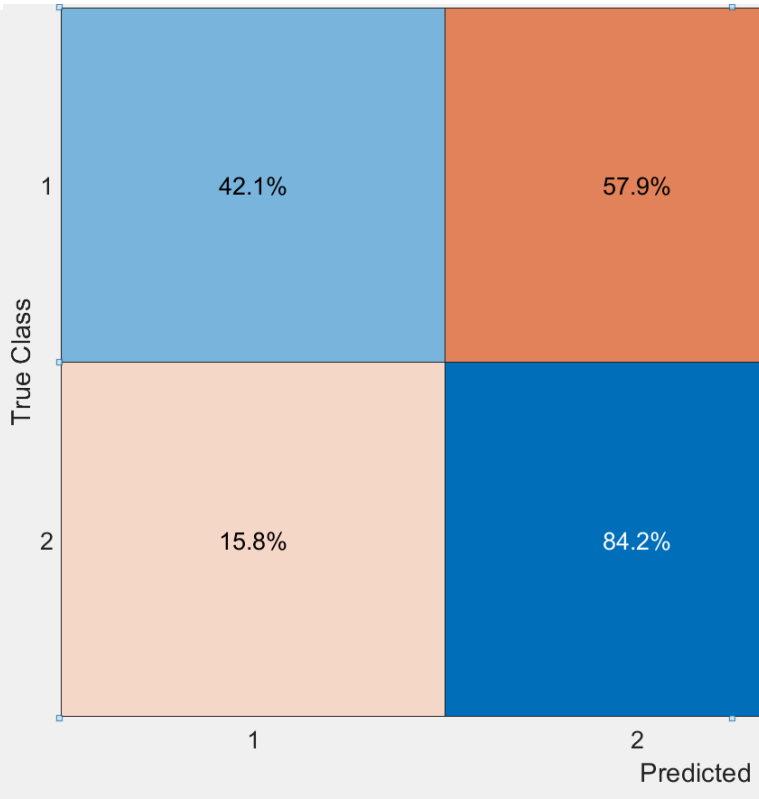
Discriminante(7 caracteristicas):



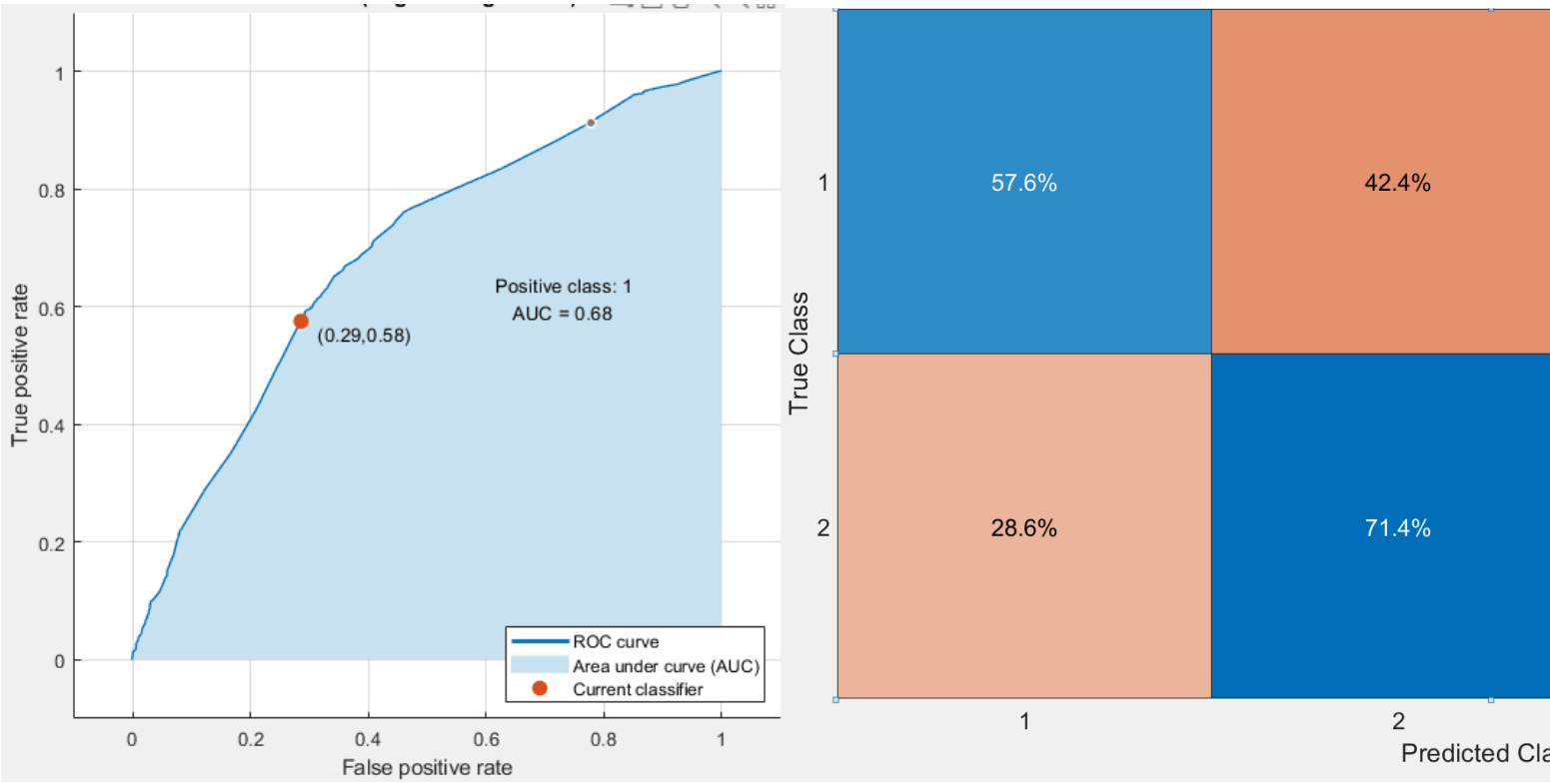


Logistic regression:

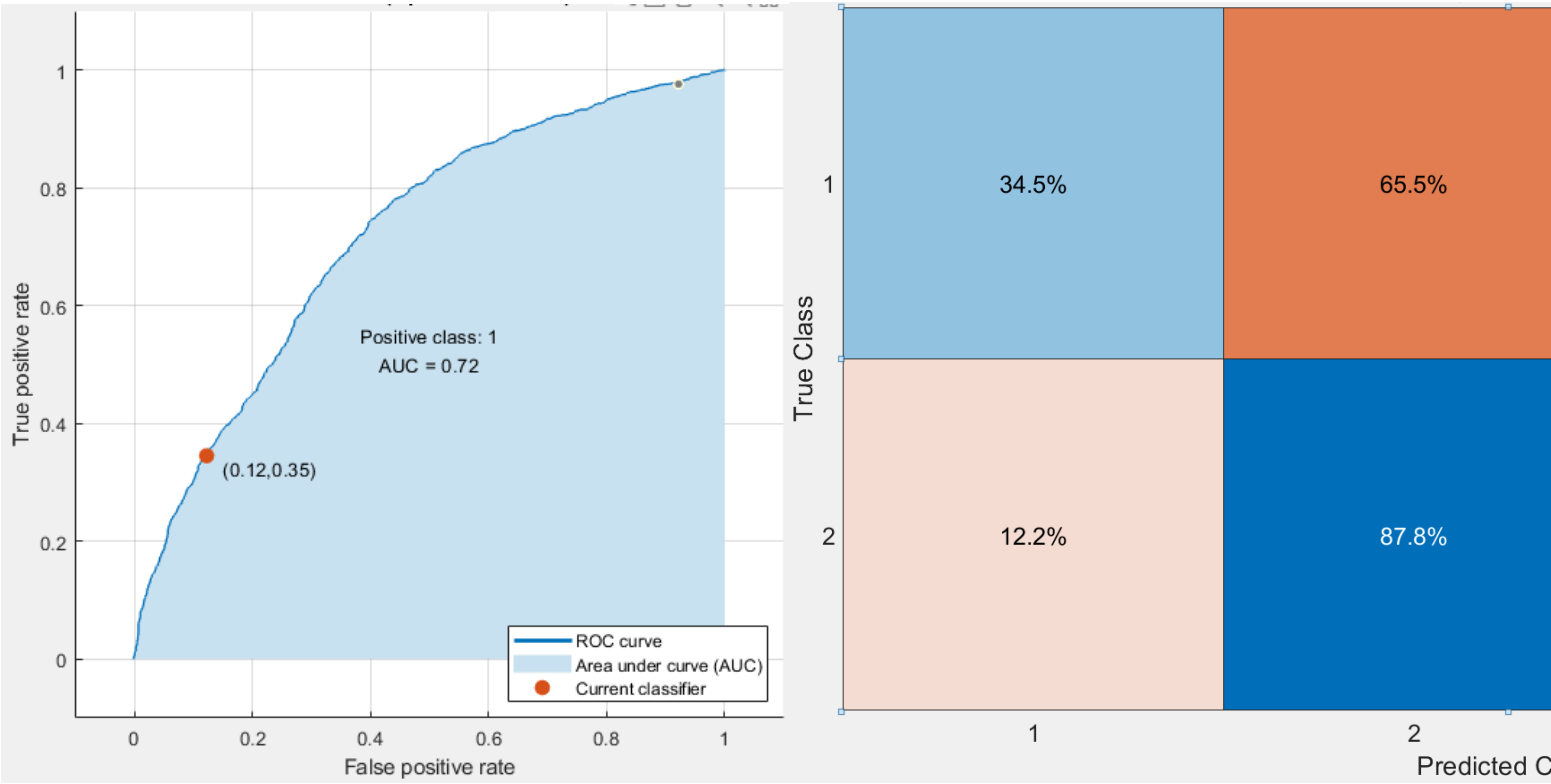




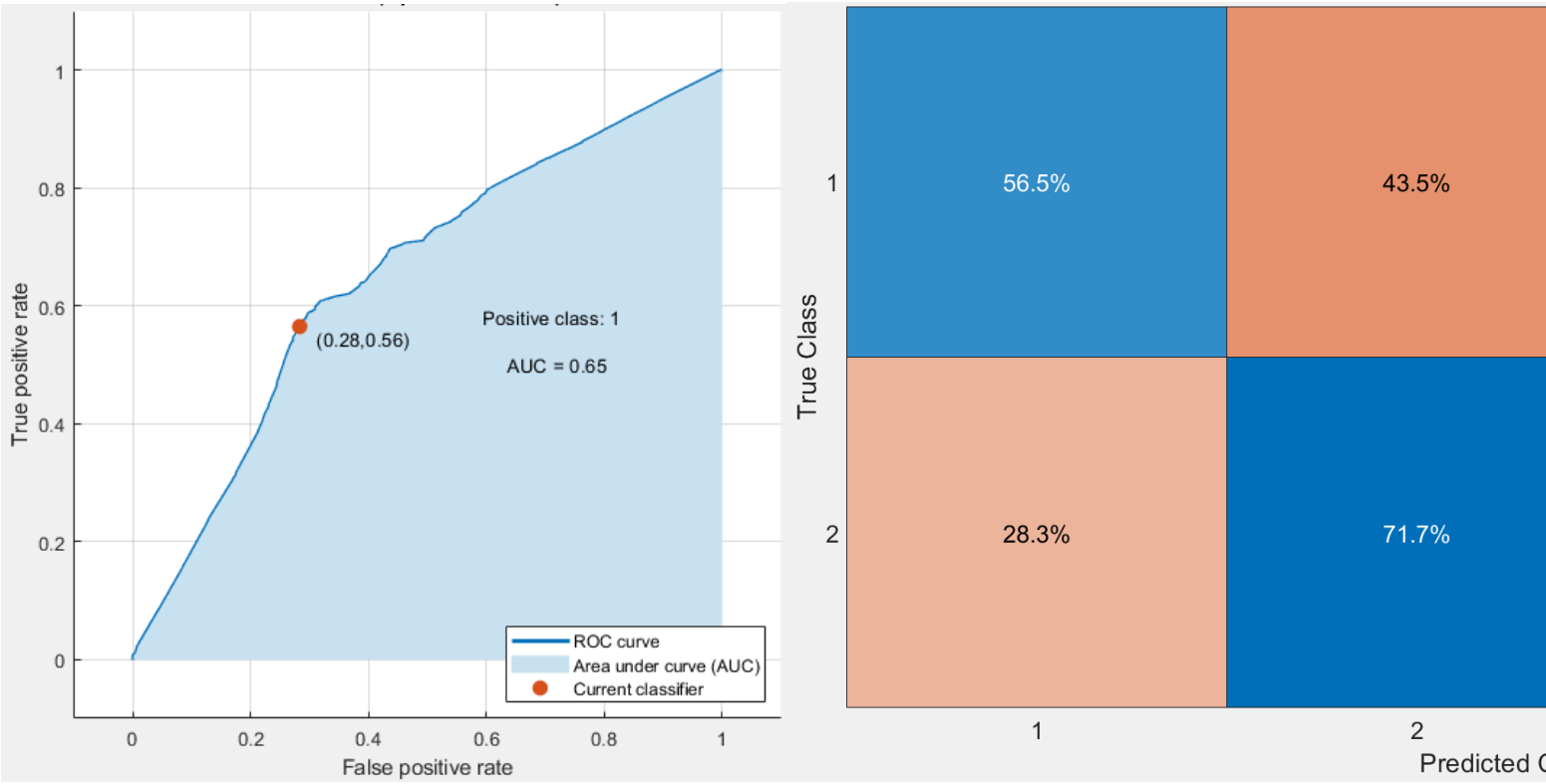
Logistic regression (7 caracteristicas):


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%
Predicted Cla

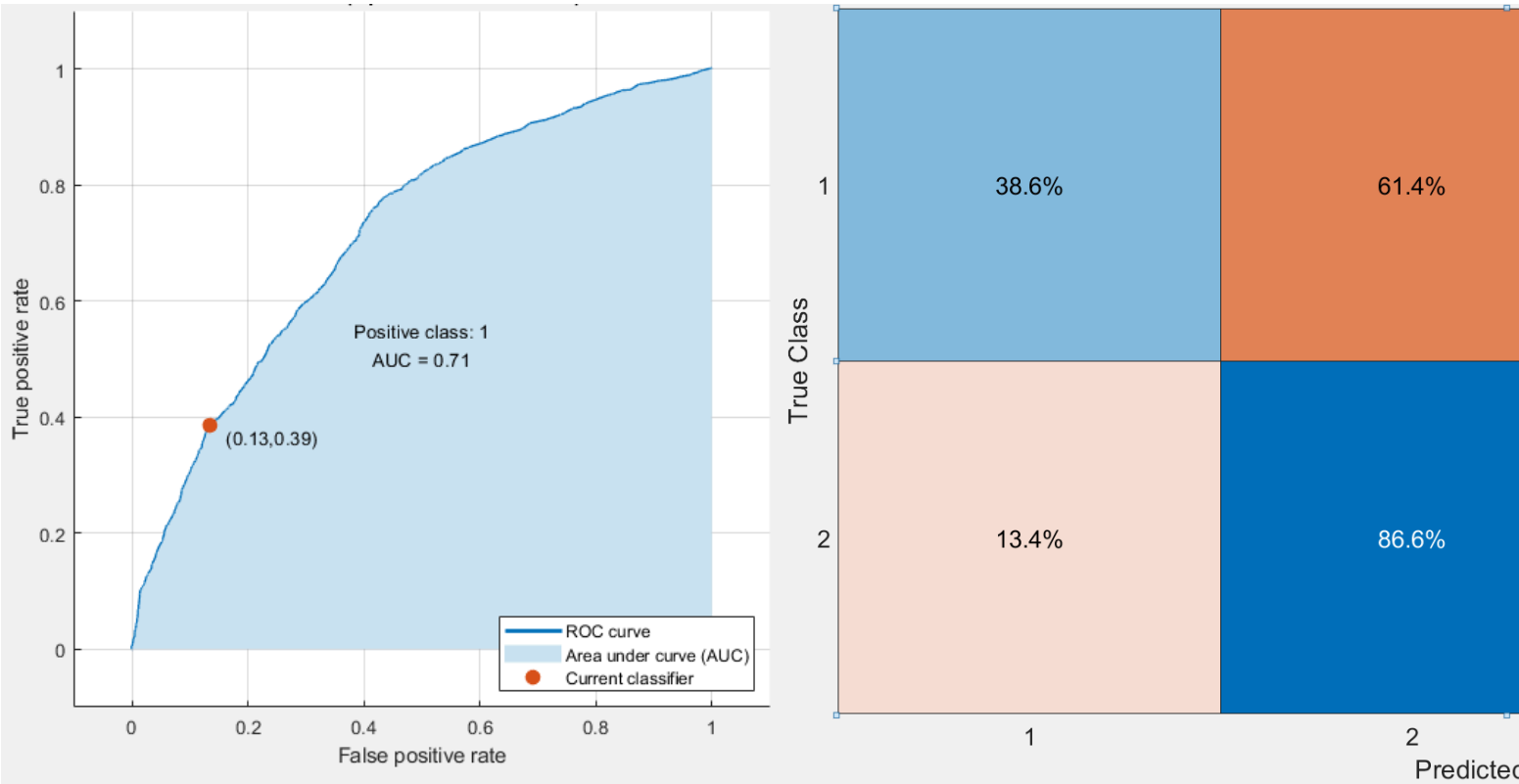
SVM:


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%
Predicted C

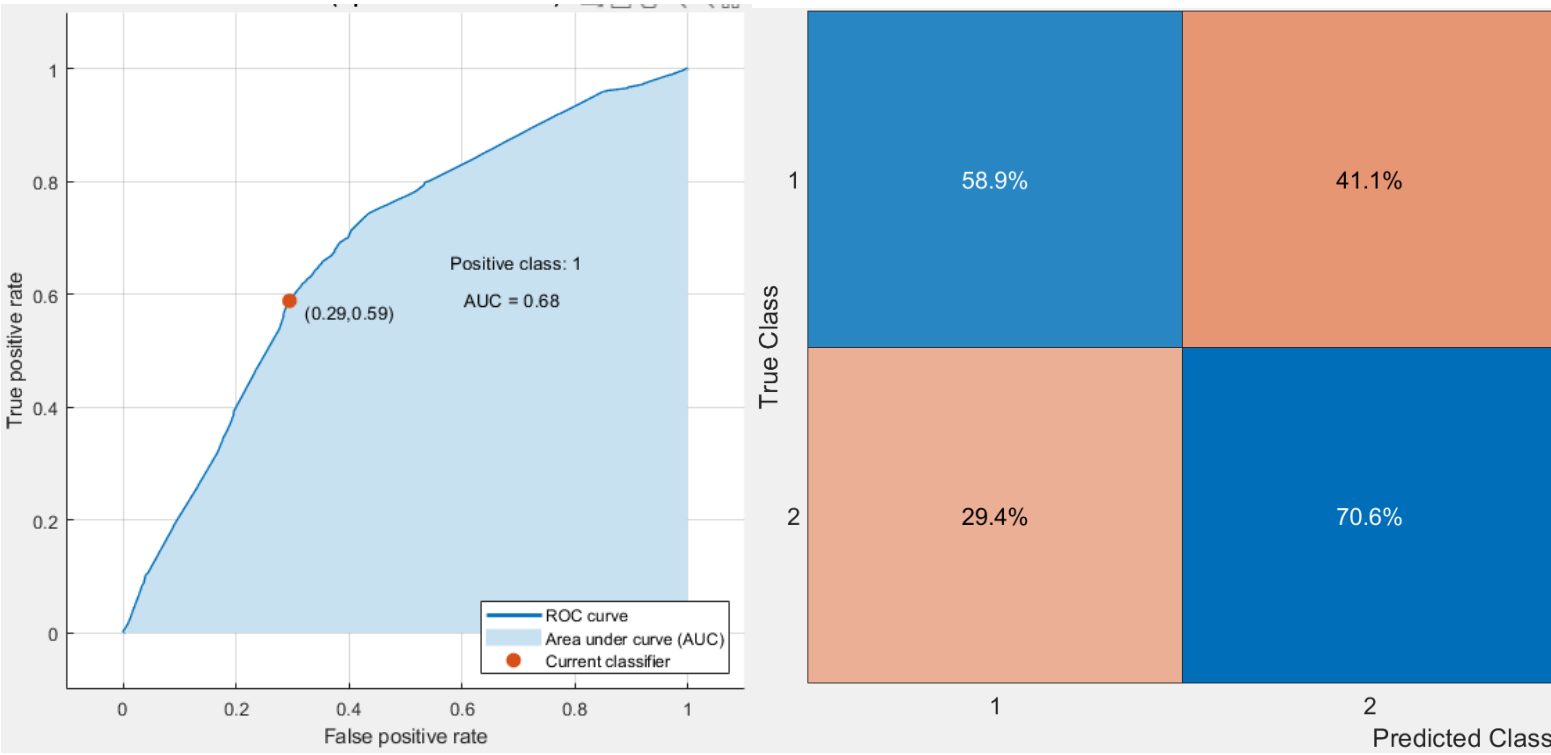
SVM(7 caracteristicas):



Ensemble:



Ensemble(7 caracteristicas):



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.
- La complejidad del modelo, se preferiran modelos mas simples
- Un ultimo factor que se tendra en cuenta para preferir un modelo sobre otro, es la distribcuion 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.

Provisional:

Teniendo en cuenta lo anterior lo que se busca es maximar la prediccion correcta de la etiqueta 1 {si a intentos previos de suicidio}, en este sentido la mayoria de modelos son deficientes. Pero en general los de 7 caracteristicas se comportan mejor que los completos. Asi, uno de los mejores seria el de discriminante linarl con 7 caracteristicas

ya que es el que mas se acerca a lo requerido, tienen poco numero de caracteristicas y no hace parte de los modelos demasiado complejos

Pendinete Escobar:

- **calcular score del modelos,**
- **poner analisis general de ROC y CM**
- **Hacer eso de la predicicon probabilistica y determinista**

Conclusiones

El chorro:

- *Hablar de porque nuestros modelos son tan malos(posibles razones:se hicieron las consas mal? xd. Caracteristicas comunes para aquellos que son reincidentes por lo que es dificl separarlos de aquello que no?. Prefereiblemnete una mayor cantidad de datos(y tambien mejor calidad ya que habia muchos datos faltants))*
- *Comentar que podria significar segun el contexto esas 7 caracteristicas mas importantes*
- *Como seria excelente de acuerdo a la propuesta inicial poder consefuir datasets con informacion de personas que han intentando previamente el sucisdio 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

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**}*