```
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
- F1

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.

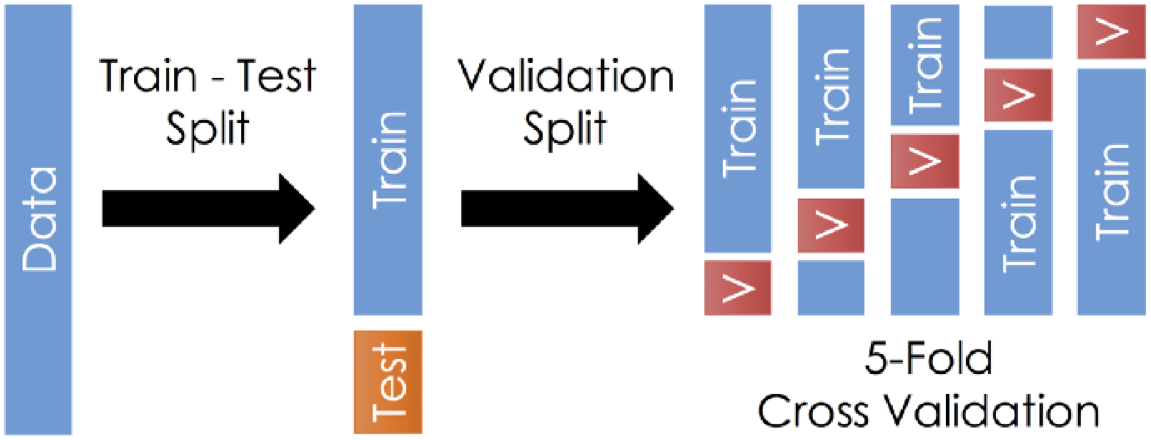
```
%cds = readtable('cds.csv'); size_cds = size(cds)
cds_imputed = readtable('cds_imputed.csv'); size_imputed = size(cds_imputed)
```

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

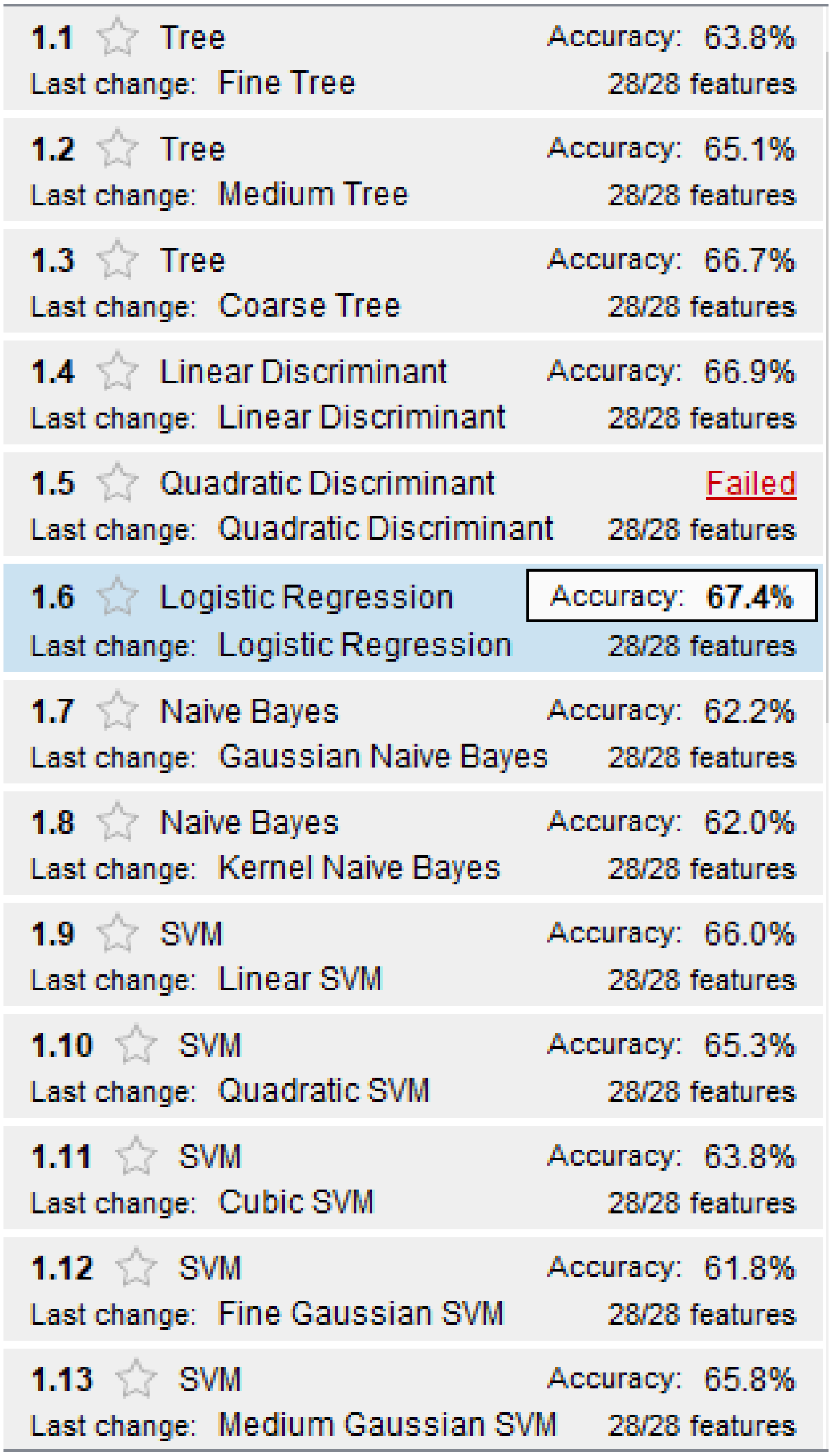
```
cds_imputed = movevars(cds_imputed,'inten_prev','after','tipo_ss_S');
%cds_few = readtable('cds_few.csv'); size_few = size (cds_few)
%cds_few_minus_alcohol = readtable('cds_few_minus_alcohol.csv');
%       size_few_minus_alcohol = size (cds_few_minus_alcohol)
```

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 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

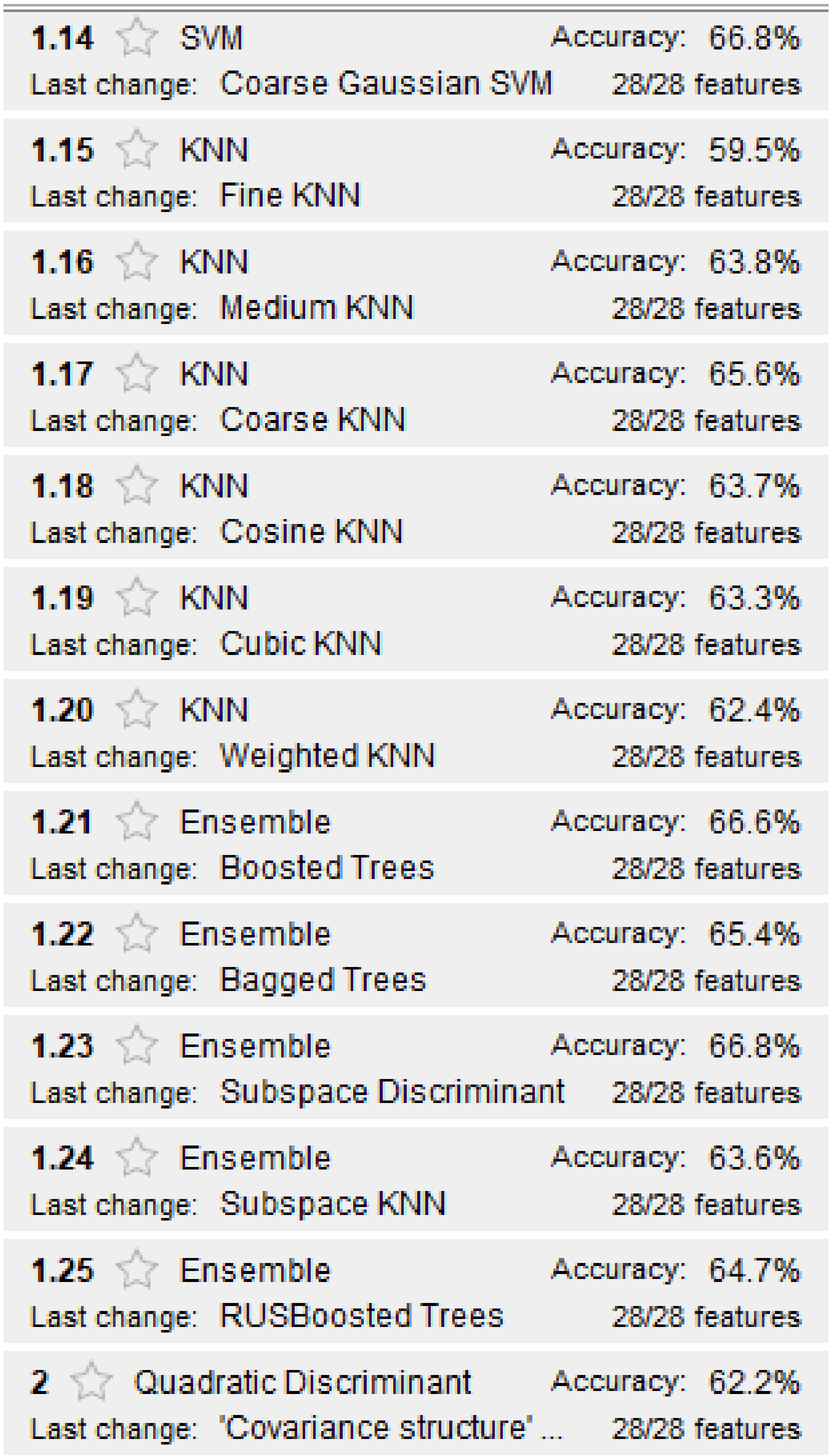
1.9 ☆ SVM Accuracy: 66.0%
Last change: Linear SVM 28/28 features

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

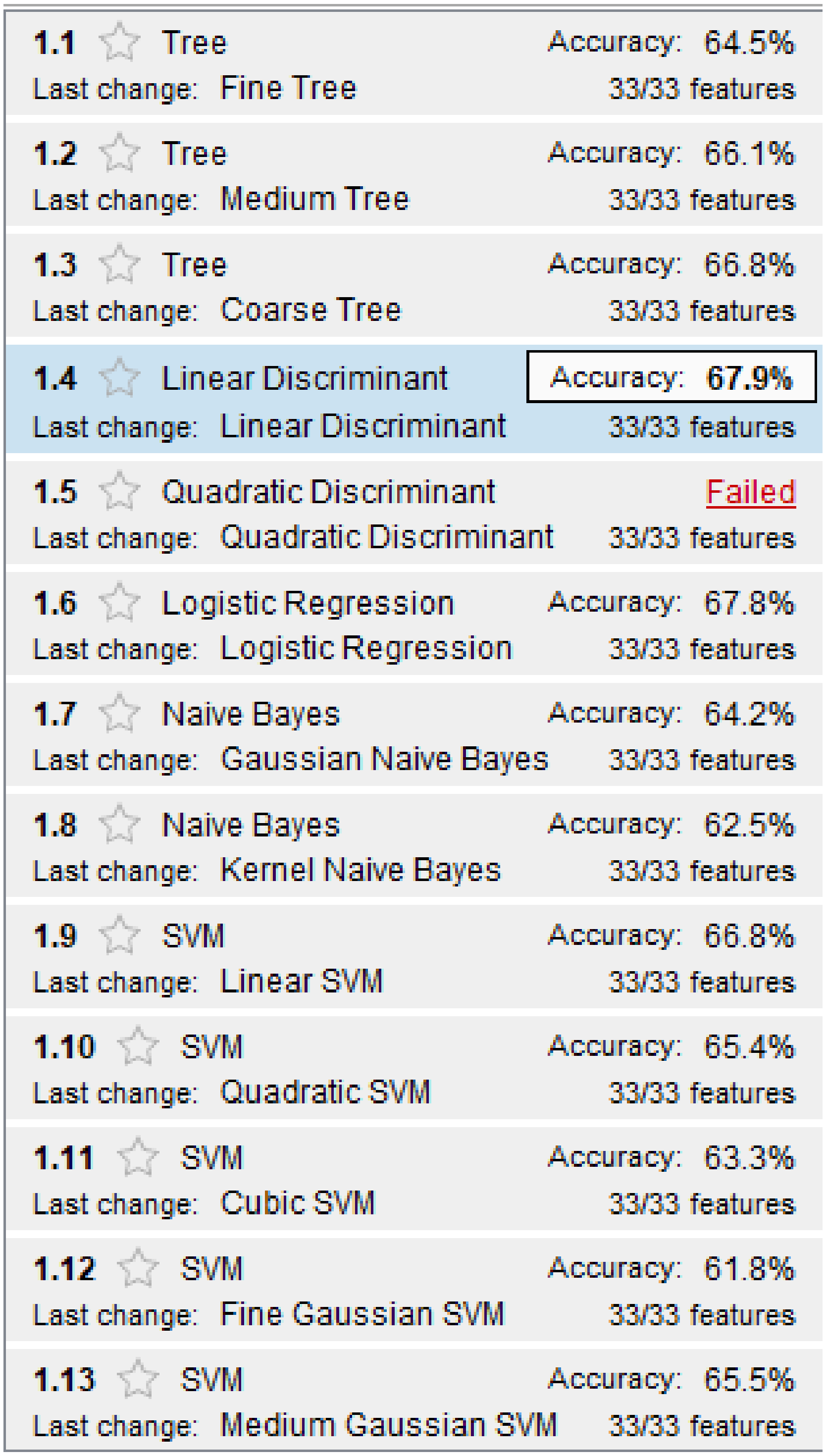
1.23 ☆ Ensemble Accuracy: 66.8%
Last change: Subspace Discriminant 28/28 features

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

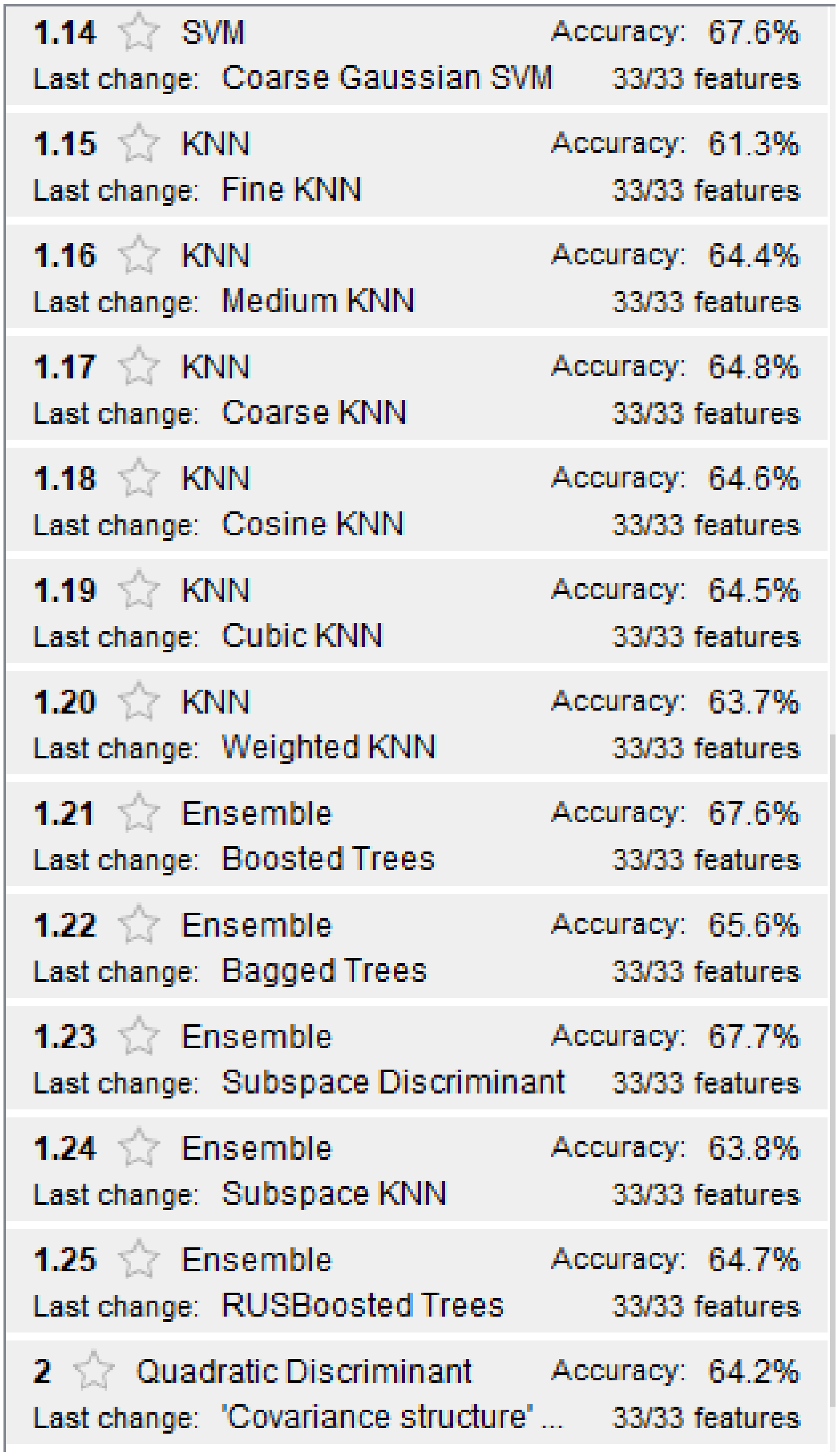
1.9 ☆ SVM Accuracy: 66.8%
Last change: Linear SVM 33/33 features

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



1.14 ☆ SVM Accuracy: 67.6%
Last change: Coarse Gaussian SVM 33/33 features

1.15 ☆ KNN Accuracy: 61.3%
Last change: Fine KNN 33/33 features

1.16 ☆ KNN Accuracy: 64.4%
Last change: Medium KNN 33/33 features

1.17 ☆ KNN Accuracy: 64.8%
Last change: Coarse KNN 33/33 features

1.18 ☆ KNN Accuracy: 64.6%
Last change: Cosine KNN 33/33 features

1.19 ☆ KNN Accuracy: 64.5%
Last change: Cubic KNN 33/33 features

1.20 ☆ KNN Accuracy: 63.7%
Last change: Weighted KNN 33/33 features

1.21 ☆ Ensemble Accuracy: 67.6%
Last change: Boosted Trees 33/33 features

1.22 ☆ Ensemble Accuracy: 65.6%
Last change: Bagged Trees 33/33 features

1.23 ☆ Ensemble Accuracy: 67.7%
Last change: Subspace Discriminant 33/33 features

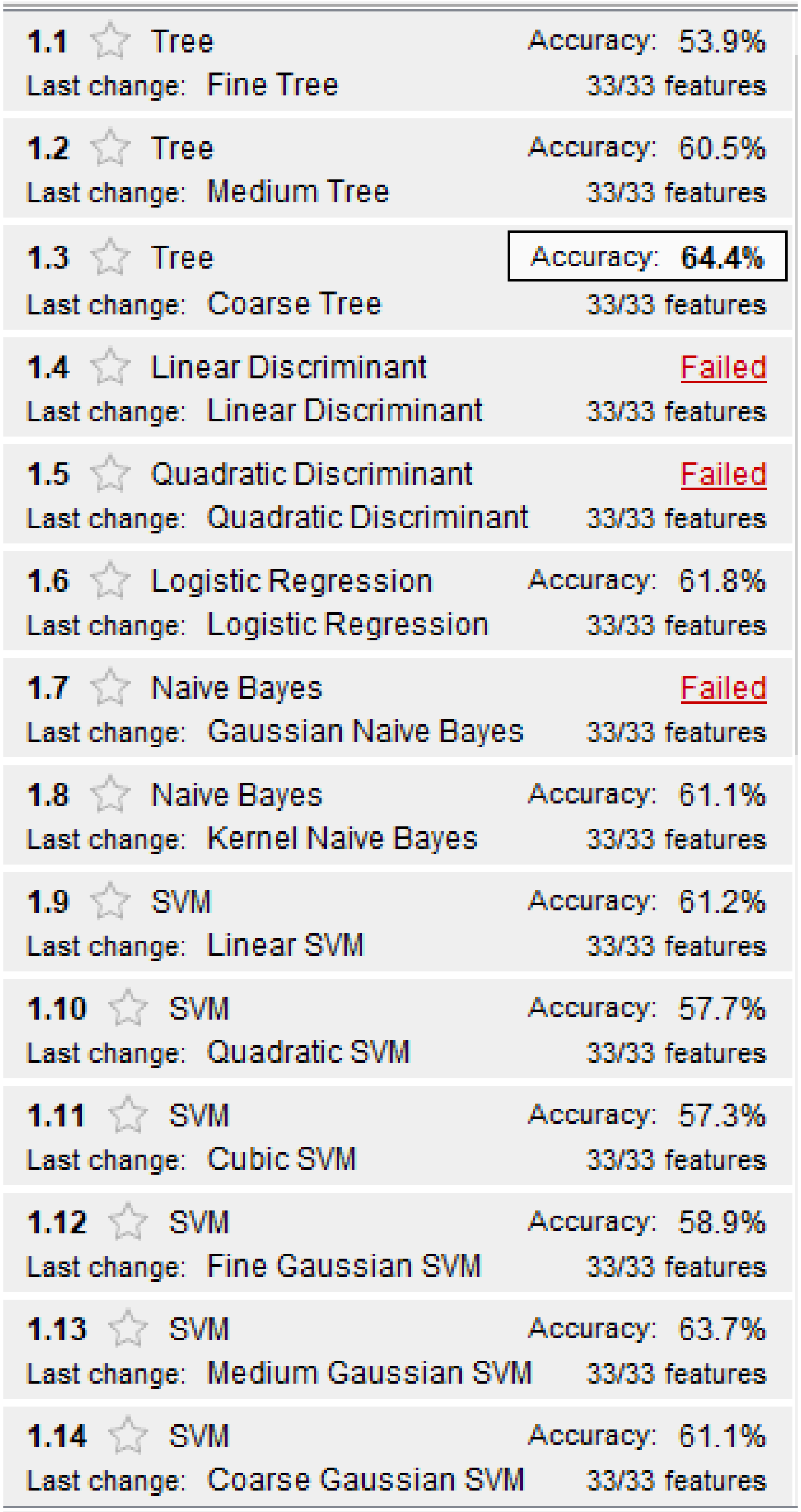
1.24 ☆ Ensemble Accuracy: 63.8%
Last change: Subspace KNN 33/33 features

1.25 ☆ Ensemble Accuracy: 64.7%
Last change: RUSBoosted Trees 33/33 features

2 ☆ 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 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

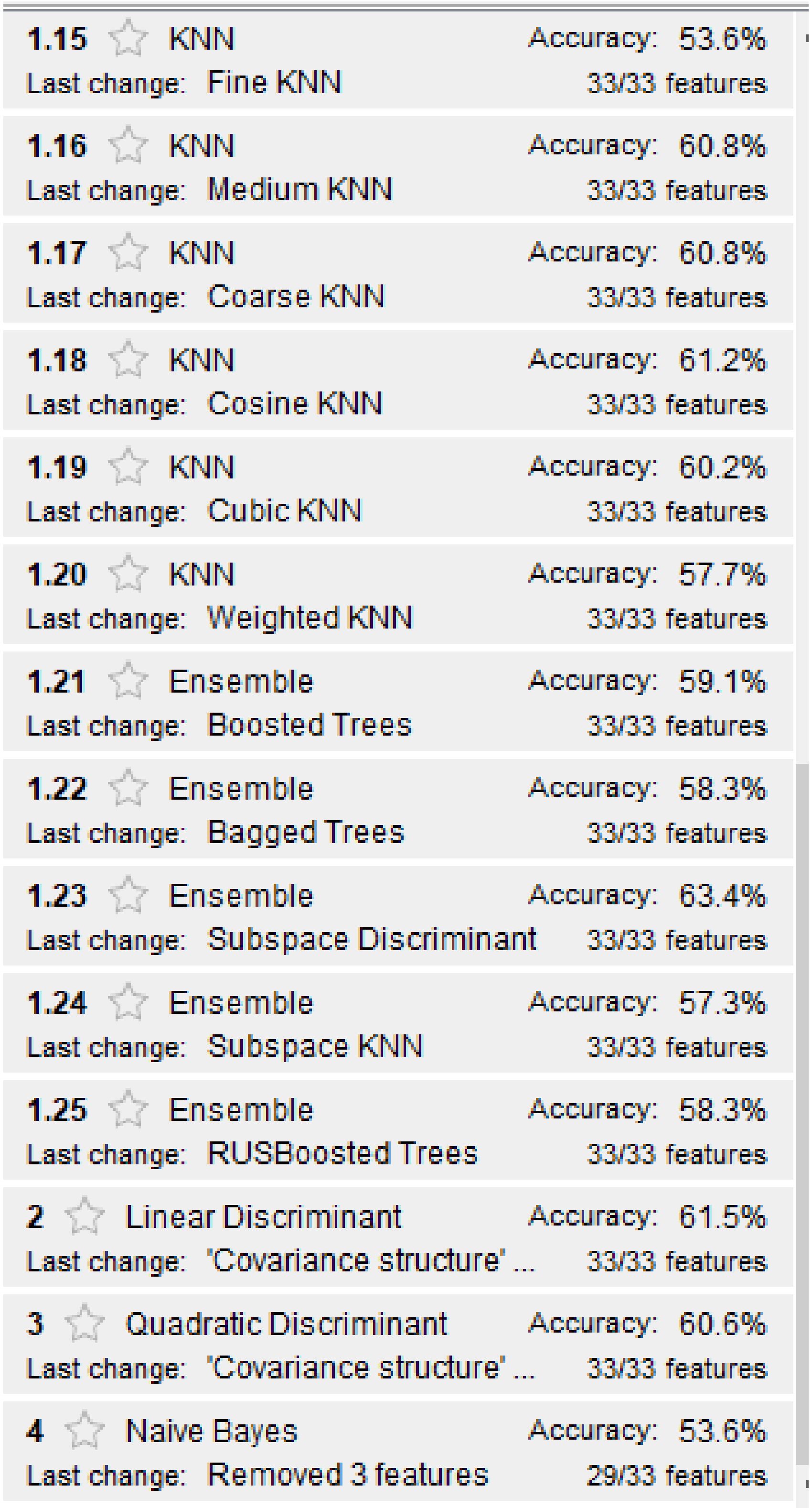
1.10 ☆ SVM Accuracy: 57.7%
Last change: Quadratic SVM 33/33 features

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



1.15 ☆ KNN Accuracy: 53.6%
Last change: Fine KNN 33/33 features

1.16 ☆ KNN Accuracy: 60.8%
Last change: Medium KNN 33/33 features

1.17 ☆ KNN Accuracy: 60.8%
Last change: Coarse KNN 33/33 features

1.18 ☆ KNN Accuracy: 61.2%
Last change: Cosine KNN 33/33 features

1.19 ☆ KNN Accuracy: 60.2%
Last change: Cubic KNN 33/33 features

1.20 ☆ KNN Accuracy: 57.7%
Last change: Weighted KNN 33/33 features

1.21 ☆ Ensemble Accuracy: 59.1%
Last change: Boosted Trees 33/33 features

1.22 ☆ Ensemble Accuracy: 58.3%
Last change: Bagged Trees 33/33 features

1.23 ☆ Ensemble Accuracy: 63.4%
Last change: Subspace Discriminant 33/33 features

1.24 ☆ Ensemble Accuracy: 57.3%
Last change: Subspace KNN 33/33 features

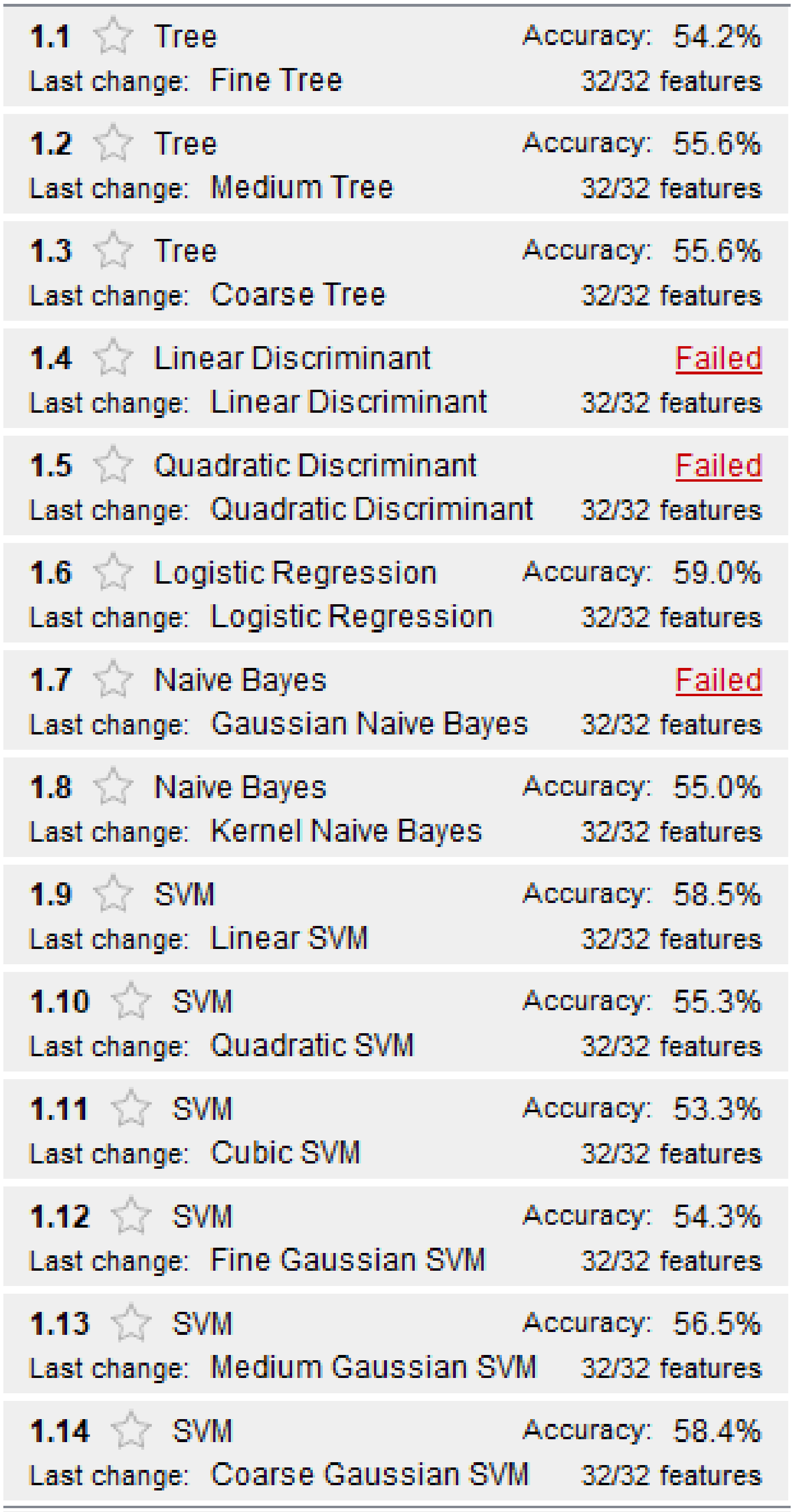
1.25 ☆ Ensemble Accuracy: 58.3%
Last change: RUSBoosted Trees 33/33 features

2 ☆ Linear Discriminant Accuracy: 61.5%
Last change: 'Covariance structure' ... 33/33 features

3 ☆ Quadratic Discriminant Accuracy: 60.6%
Last change: 'Covariance structure' ... 33/33 features

4 ☆ Naive Bayes Accuracy: 53.6%
Last change: Removed 3 features 29/33 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

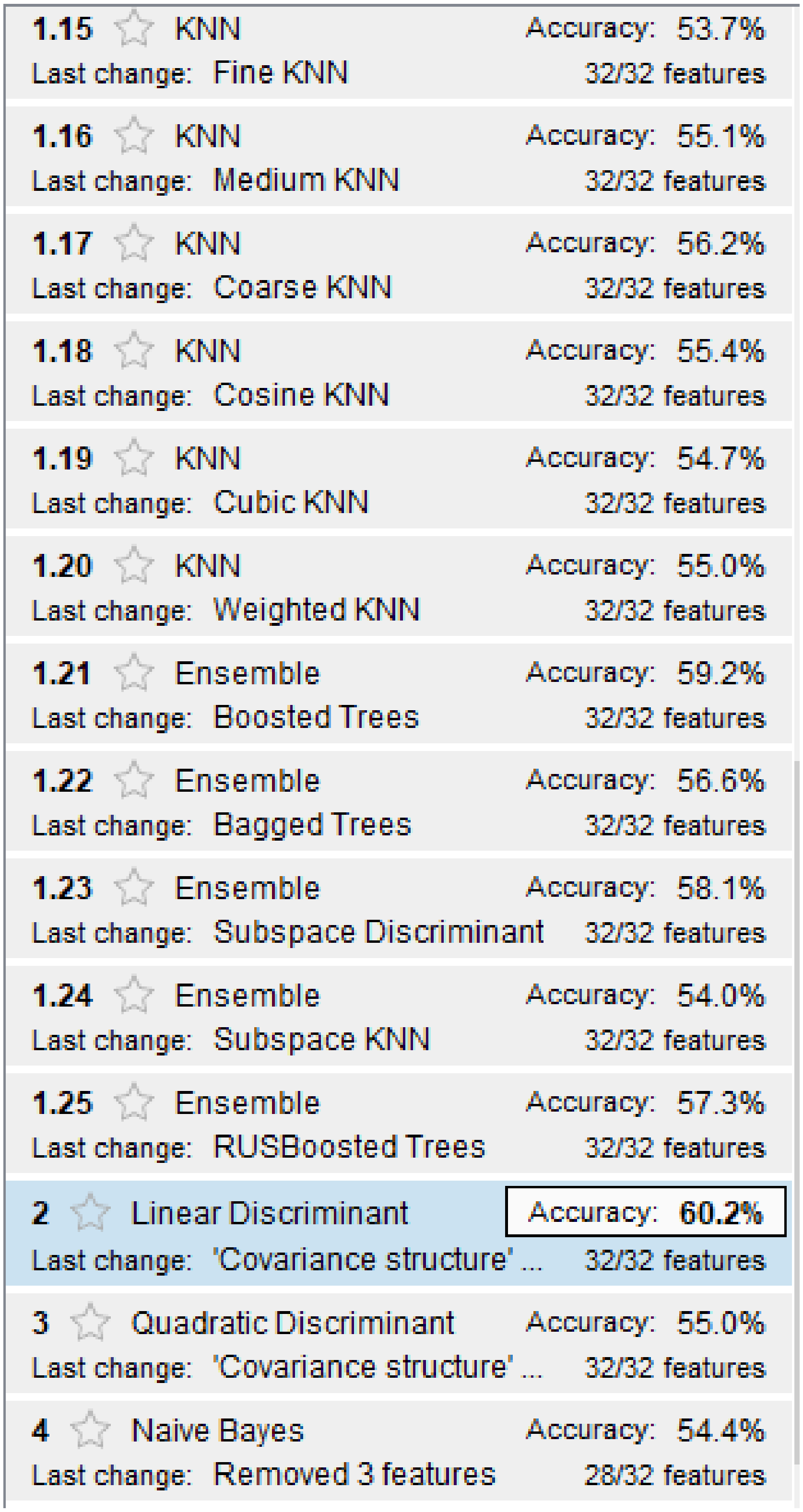
1.10 ☆ SVM Accuracy: 55.3%
Last change: Quadratic SVM 32/32 features

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



1.15 ☆ KNN Accuracy: 53.7%
Last change: Fine KNN 32/32 features

1.16 ☆ KNN Accuracy: 55.1%
Last change: Medium KNN 32/32 features

1.17 ☆ KNN Accuracy: 56.2%
Last change: Coarse KNN 32/32 features

1.18 ☆ KNN Accuracy: 55.4%
Last change: Cosine KNN 32/32 features

1.19 ☆ KNN Accuracy: 54.7%
Last change: Cubic KNN 32/32 features

1.20 ☆ KNN Accuracy: 55.0%
Last change: Weighted KNN 32/32 features

1.21 ☆ Ensemble Accuracy: 59.2%
Last change: Boosted Trees 32/32 features

1.22 ☆ Ensemble Accuracy: 56.6%
Last change: Bagged Trees 32/32 features

1.23 ☆ Ensemble Accuracy: 58.1%
Last change: Subspace Discriminant 32/32 features

1.24 ☆ Ensemble Accuracy: 54.0%
Last change: Subspace KNN 32/32 features

1.25 ☆ Ensemble Accuracy: 57.3%
Last change: RUSBoosted Trees 32/32 features

2 ☆ Linear Discriminant Accuracy: **60.2%**
Last change: 'Covariance structure' ... 32/32 features

3 ☆ Quadratic Discriminant Accuracy: 55.0%
Last change: 'Covariance structure' ... 32/32 features

4 ☆ 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 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 cacteriscas. 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

Se presentara el proceso para cada uno de los modelos reaizados mediante optimizacion bayesiana

Arboles de decision:

Linear discriminant:

Logistic regression

SVM:

Ensamble:

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

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 y como seria excelente de aceurdo a la propuesta inicial poder consefuir datasets con informacion de personas que han intentando previamente el sucisdio 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