Uso de algoritmos de Aprendizaje automático aplicados a Base de datos genéticas

Santiago Navarro Jurado

Máster en Bioinformática y Bioestadística UOC-UB

Programación para la BioInformática

Pau Andrio Baladó

26/12/2016

B) GNU Free Documentation License (GNU FDL)

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Uso de algoritmos de aprendizaje automático aplicados a BD genéticas*
Nombre del autor:	*Santi Navarro*
Nombre del consultor:	*Pau Andrio Balado*
Nombre del PRA	
Fecha de entrega (mm/aaaa):	*12/2016*
Titulación:	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*Programación Bioinformática*
Idioma	*Castellano*
Palabras clave	*Python, Machine learning, ML, scikit-learn, Proyecto Hapmap, SNP, clasificación, agrupamiento, matplotlib, Django, Html5, Bootstrap , Responsive, Ajax, numpy, pandas, Yaml, Json*

Resumen del Trabajo:

En este proyecto se lleva a cabo un análisis de algoritmos de aprendizaje automático centrado en clasificación y clustering, utilizando bases de datos genéticas.

Se ha realizado utilizando datos del proyecto Hapmap (revisión de Fase II en octubre de 2008), el cual pretendía desarrollar un mapa del haplotipo humano a través de la identificación de SNP's de seres humanos de diferentes etnias.

El proyecto se ha desarrollado en lenguaje Python mediante las librerías de aprendizaje automático scikit-learn.

Como desarrollos se han llevado a cabo dos aplicativos, un aplicativo con interfaz Web mediante el framework Django y un aplicativo de línea de comando para procesado en modo Batch.

Ambos aplicativos realizan clasificación y agrupamiento por población a través de la evaluación de varios diferentes algoritmos.

Los resultados obtenidos por los estimadores se mesuran a través de curvas ROC y métricas propias de clasificación y agrupamiento.

Abstract (in English, 250 words or less):

This Project develops an analisys of machine learning algorithms focused on classification and clustering, using genetic databases.

It has been done using Project Hapmap data (Phase II revisión, October 2008), which developed a human haplotype map throught SNP's identification on human beings from different ethnic groups.

It has been developed with Python programming language and using scikit-learn machine learning libraries.

There has been developed two applications, one with Web user interface, using Django framework, and other, a command line program for processing in batch mode.

Both applications do classification and clustering evaluates using several different algorithms.

Obtained results for the different estimators are then evaluated using Roc curves and classification and clustering specific metrics.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

En la última década el volumen de datos biológicos, como secuencias de DNA, simulaciones de dinámica de proteínas o información sobre niveles de expresión, se ha incrementado exponencialmente. Gracias a este incremento ahora somos capaces de describir con más precisión sistemas y procesos biológicos. [1]

Sin embargo, junto con el incremento de volumen en los datos, también se han incrementado la complejidad, la dimensionalidad y el ruido de los mismos.

Se hace necesaria la aplicación de técnicas de minería de datos y de aprendizaje automático para extraer nueva información de estos datos. [1]

En el proyecto Hapmap se pretendía desarrollar un mapa del haplotipo del genoma humano.

A través del proyecto Hapmap se puede acceder a los datos de SNP de hasta 1300 personas [2] en 11 distintas poblaciones.

En este proyecto se pretende realizar un análisis de datos Hapmap y llevar a cabo una comparativa sobre algoritmos de aprendizaje automático.

A partir de los datos del proyecto Hapmap [2] , se realizará un entrenamiento sobre diferentes algoritmos de aprendizaje automático con la finalidad de **clasificar la población** según los datos de SNP.

Se pretende en este proyecto el análisis de datos genómicos de poblaciones con la finalidad de predecir ciertas características sobre estas poblaciones a través de algoritmos de aprendizaje automático (estimadores) de clasificación y agrupamiento .

1.2 Objetivos del Trabajo

El objetivo principal es desarrollar una serie de análisis sobre los datos de HapMap para detectar qué características de estos datos son las más significativas para conseguir agrupar los datos, clasificarlos y realizar predicciones sobre estos.[2]

Objetivo general:

- Creación de un portal web para visualizar modelos de clasificación y modelos de predicción sobre el conjunto de datos de HapMap.

Objetivos secundarios:

- Descargar los datos necesarios
- Preprocesar los datos a un formato apropiado para poder proporcionarlos a los algoritmos de aprendizaje automático (AA).
- Registrarlos en un almacén de datos adecuado para poder ser utlizados
- Desarrollo de un software que utilice estos datos y los proporcione a un algoritmo de AA.
- Desarrollar programario que permita
    - Clasificar los datos
    - hacer predicciones sobre los datos.
- Crear un portal web que permita interactuar con el programario desarrollado y visualizar los resultados.

1.3 Enfoque y método seguido

El proyecto se llevará a cabo como un desarrollo de software nuevo.

Contendrá un entregable final que consistirá en una aplicación web, desde la que el investigador podrá realizar análisis sobre datos HapMap para clasificar y realizar predicciones sobre estos.

Acotado por los entregables de la asignatura de trabajo de final de máster, se planifican las siguientes revisiones del aplicativo :

- Desarrollo de aplicación backend para la conversión de datos Hapmap
- Desarrollo de utilerías backend para el análisis y clasificación de datos
    - Desarrollada en Python, utilizando las librerías Scikit-learn, pandas y numpy
- Desarrollo de frontend web como interfaz de usuario
    - En él se permitirá al usuario la ejecución de los algoritmos de clasificación y agrupamiento contra los datos Hapmap.

- Desarrollada en Django y HTML5 con Bootstrap y Ajax.

Las herramientas durante el desarrollo de todas las partes del proyecto serán de libre uso, o con la licencia apropiada.

Se referenciará la herramienta y versión utilizada dónde sea preciso.

Para una correcta gestión del software, su ampliación y mantenimiento, el desarrollo se llevará a cabo utilizando metodología de Modelo-Vista-Controlador.

Repositorio Bitbucket

Como control de versiones se utilizarán herramienta basadas en Git en línea de comandos y el repositorio de internet provisto por bitbucket.org en

- https://bitbucket.org/PauAndrio/tfmsanti/

para el seguimiento por parte del alumno y el profesorado colaborador.

A Través de este puede consultarse la evolución del proyecto y clonar el código en él.

1.4 Planificación del Trabajo

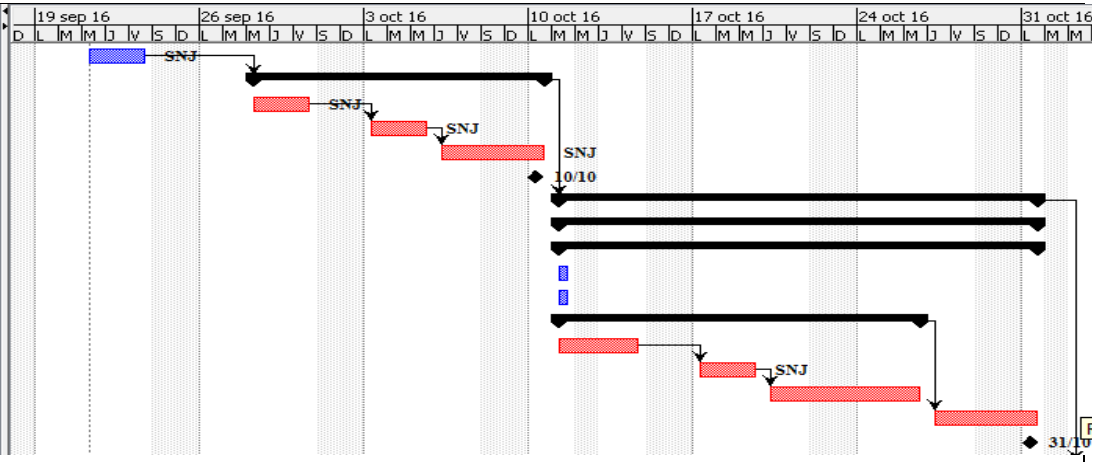
La duración de las tareas está expresada en días, y ha sido ajustada a los hitos de entrega propuestos en la asignatura.

Se utilizó el calendario laboral de Cataluña, que se corresponde con el área geográfica del desarrollador.

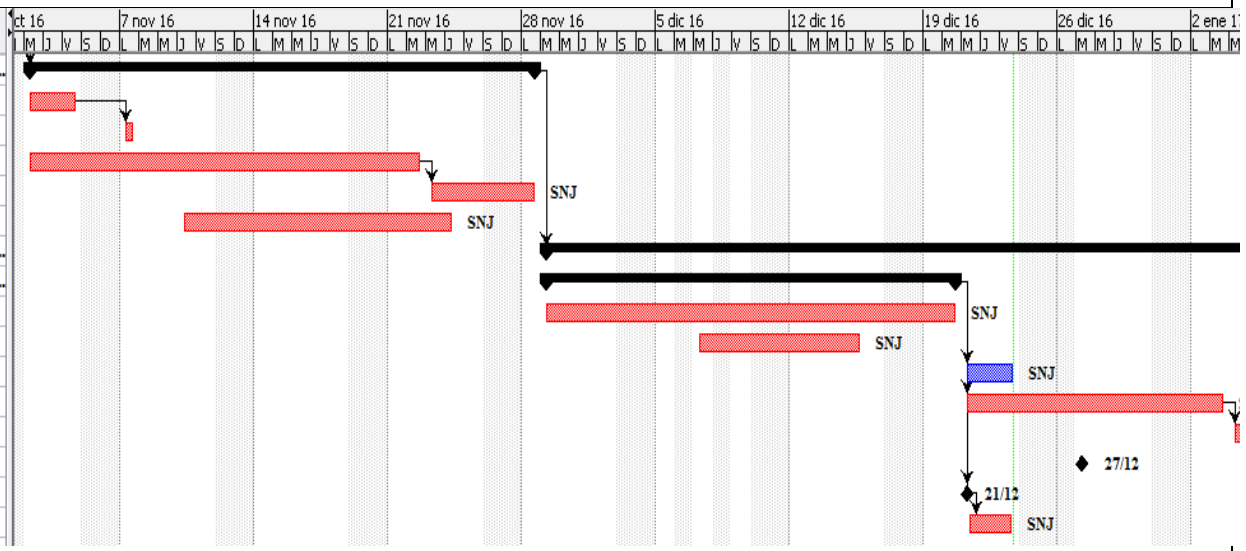
Aún así, la dedicación del desarrollador está ajustada a las horas acotadas por el número de créditos de proyecto, de alrededor de 300 horas. La dedicación horaria por días del desarrollador queda especificada más adelante en el documento.

El diagrama de Gantt del desarrollo final ha quedado así:

Nombre	Dura...	Inicio	Terminado
PEC 0 - Elección del trabajo concreto a realizar	3 days	21/09/16 8:00	23/09/16 17:00
PEC1 - Plan de trabajo	**9 da...**	**28/09/16 8:00**	**10/10/16 17...**
Análisis de requerimientos	3 da...	28/09/16 8:00	30/09/16 17:00
Planificación temporal	3 days	3/10/16 8:00	5/10/16 17:00
Creación de documento	3 days	6/10/16 8:00	10/10/16 17:00
Entregable 1 : Plan de trabajo	0 days	10/10/16 8:00	10/10/16 8:00
PEC2 - Desarrollo del trabajo - Fase 1	**14 d...**	**11/10/16 8:00**	**31/10/16 17...**
Desarrollo de SW clasificación de datos	**14 d...**	**11/10/16 8:00**	**31/10/16 17...**
Gestión inicial de datos	**14 d...**	**11/10/16 8:00**	**31/10/16 17...**
Descarga inicial de datos	1 day?	11/10/16 8:00	11/10/16 17:00
Instalación de herramientas y entorno	1 day?	11/10/16 8:00	11/10/16 17:00
Software de Preprocesado y limpieza de d	**11 d...**	**11/10/16 8:00**	**26/10/16 17...**
Análisis de alternativas de almacenaje	3 days	11/10/16 8:00	14/10/16 17:00
Desarrollo Modelo de Datos	3 days	17/10/16 8:00	19/10/16 17:00
Desarrollo del FCHapmap y Preprocesado	5 days	20/10/16 8:00	26/10/16 17:00
Documentación Entrega 1	3 days	27/10/16 8:00	31/10/16 17:00
Entregable 2 : Desarrollo Fase 1	0 days	31/10/16 9:00	31/10/16 9:00



Nombre	Dura...	Inicio	Terminado
PEC3 - Desarrollo del trabajo - fase 2	**19 d...**	**2/11/16 8:00**	**28/11/16 17...**
Software de Preprocesado y limpieza de datos Hapm	3 days	2/11/16 8:00	4/11/16 17:00
Conversión de datos	1 day?	7/11/16 8:00	7/11/16 17:00
Creación del software ML para predicción de datos	15 d...	2/11/16 8:00	22/11/16 17:00
Documentación	4 days	23/11/16 8:00	28/11/16 17:00
Testing	10 d...	10/11/16 9:00	24/11/16 9:00
Front end Web y Memoria del trabajo final	**32 d...**	**29/11/16 8:00**	**17/01/17 17...**
Front End Web para interactuar con SW desa	**14 d...**	**29/11/16 8:00**	**20/12/16 17...**
Creación de portal	14 d...	29/11/16 8:00	20/12/16 17:00
Creación de middleware para software ML	6 days	6/12/16 8:00	15/12/16 17:00
Memoria - Redacción	3 days	21/12/16 8:00	23/12/16 17:00
Presentación - creación	9 days	21/12/16 8:00	3/01/17 17:00
Guión/Video de Defensa Pública	9 days	4/01/17 8:00	17/01/17 17:00
Entregable 4 : Producto final, Memoria y documentac	0 days	26/12/16 8:00	27/12/16 17:00
Analisis de producto resultante final	0,37...	21/12/16 8:00	21/12/16 11:00
Ajustes y correcciones de Fase 2	2,5 ...	21/12/16 11:00	23/12/16 16:00



1 GANNT

1.5 Breve sumario de productos obtenidos

Datos Hapmap procesados

Los datos Hapmap han sido procesados, desde su versión inicial organizada en ficheros con tablas donde las columnas se corresponden a personas y las filas a los valores de genotipo para cada SNP.

Se almacenan en ficheros planos Preparados para los algoritmos de aprendizaje automático de clasificación y agrupamiento.

Herramienta de procesado de datos

Para este fin se han desarrollado herramientas que permiten el procesado, filtraje y modificación de datos en batch, partiendo de los ficheros originales del estilo

- genotypes_chr**<cromosoma>**_**<población>**_r28_nr.b36_fwd.txt

Su modo de uso no contiene parámetros, se invoca con :

>Preproceso.py

Aplicación Web

Para la interfaz de usuario principal se ha realizado un desarrollo de una aplicación Web, **pyranhap**, a la cual se accede mediante url

http://servidor:8000/pyranhap

En esta que se pueden realizar cálculos de predicción sobre la población de los datos Hapmap, con 16 algoritmos de clasificación y agrupamiento.

De los datos Hapmap originales sólo se pueden utilizar como etiqueta para el aprendizaje el dato sobre origen de la población.

Por tanto, las predicciones de los algoritmos se harán siempre teniendo este dato como objetivo.

Esta aplicación permite seleccionar diferentes cromosomas y diferentes poblaciones y aplicarles algoritmos de aprendizaje automático.

El usuario puede seleccionar algunos parámetros de los algoritmos, permitiendo así el ajuste de estos en la búsqueda de unos resultados mejores.

Aplicación Batch

Con el fin de realizar un análisis exhaustivo de casos sin la necesidad de permanecer delante de la interfaz Web y seleccionar los parámetros manualmente en cada caso, se ha desarrollado

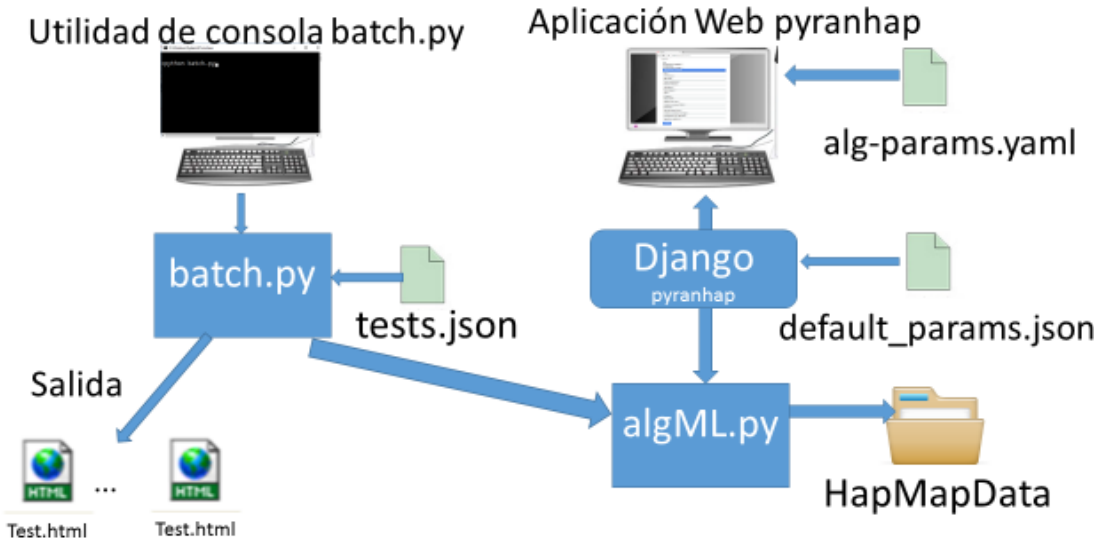
una utilidad de línea de comando que permite la evaluación de los algoritmos de aprendizaje automático en modo batch.

Su modo de uso es:

batch.py –i fichero_test.json

Sin parámetros, se utilizará el fichero por defecto tests.json

Diagrama de la arquitectura de productos



Se muestra en la imagen un diagrama de la arquitectura de productos, consistente en dos aplicaciones, una Web (a la derecha) y una de línea de comando, para generar análisis sobre datos Hapmap

1.6 Breve descripción de los otros capítulos de la memoria

En los próximos capítulos pasaré a describir los datos Hapmap y su proceso de transformación para la utilización dentro de las aplicaciones desarrolladas.

También se describirá la aplicación y el proceso llevado a cabo en su desarrollo.
Se desglosará en una descripción los algoritmos utilizados.

A posteriori se realizarán una serie de pruebas que demuestren la utilización tanto de las herramientas de conversión como de la herramienta Web para la predicción de datos de test.

Se concluye con una evaluación de los diferentes algoritmos de aprendizaje automático realizada a parir de los análisis llevados a cabo a través de las herramientas desarrolladas.

2. Datos Hapmap

Hapmap es el acrónimo de “Haplotipe map”. El proyecto Hapmap pretendía desarrollar una mapa de haplotiplos humano en el que catalogar diferencias genéticas entre individuos con el fin de comprender mejor la relación entre genoma y salud.

Los datos para este proyecto se obtienen del proyecto Hapmap.

El proyecto Hapmap ha sido retirado y su continuidad ha quedado vinculada al proyecto 1000 genomas [3].

A pesar de su discontinuidad, los ficheros de datos del proyecto Hapmap continúan estando disponibles a través del ftp ftp://ftp.ncbi.nlm.nih.gov/hapmap/

Aún así, la revisión y estructuración de estos datos resulta complicada de seguir.

Las fuentes no estan disponibles en el site, del ya deprecado proyecto. Se pierde así la capacidad de explorar herramientas para realizar análisis y visualizar estos en site del proyecto, ya que no está disponible.

Existen además múltiples revisiones y fases de datos disponibles en el ftp de NCBI.

En coordinación con el director de proyecto, se han descargado todos los datos de cadena adelantada en la revisión **de Fase II** en octubre de 2008. Estos se encuentran en:

- ftp://ftp.ncbi.nlm.nih.gov/hapmap/genotypes/2008-10_phaseII/fwd_strand/non-redundant

Los datos están organizados en ficheros de texto plano que contienen tablas de datos.

Con el fin de dividir la información para una mejor gestión de esta, los ficheros están distribuidos por cromosoma y población.

Así por ejemplo, para el cromosoma 15 obtenemos

- genotypes_chr15_ASW_r28_nr.b36_fwd.txt
- genotypes_chr15_CEU_r28_nr.b36_fwd.txt
- genotypes_chr15_CHB_r28_nr.b36_fwd.txt
- genotypes_chr15_CHD_r28_nr.b36_fwd.txt
- genotypes_chr15_GIH_r28_nr.b36_fwd.txt
- genotypes_chr15_JPT_r28_nr.b36_fwd.txt
- genotypes_chr15_LWK_r28_nr.b36_fwd.txt
- genotypes_chr15_MEX_r28_nr.b36_fwd.txt
- genotypes_chr15_MKK_r28_nr.b36_fwd.txt
- genotypes_chr15_TSI_r28_nr.b36_fwd.txt
- genotypes_chr15_YRI_r28_nr.b36_fwd.txt

donde las diferentes etiquetas representan poblaciones sobre las que se han secuenciado datos:

1 POBLACIONES

Descripción	Etiqueta
African ancestry in Southwest USA	ASW
Chinese in Metropolitan Denver, Colorado	CHD
Gujarati Indians in Houston, Texas	GIH
Han Chinese in Beijing, China	CHB
Japanese in Tokyo, Japan	JPT
Luhya in Webuye, Kenya	LWK
Maasai in Kinyawa, Kenya	MKK
Mexican ancestry in Los Angeles, California	MEX
Toscani in Italia	TSI
Utah residents with Northern and Western European ancestry from the CEPH collection	CEU
Yoruba in Ibadan, Nigeria	YRI

Formato de los ficheros Hapmap

Los ficheros de Hapmap contienen tablas que, básicamente, se distribuyen así:

- Cada fila representa los SNPs (etiquetados según nomenclatura de dbSNP[4] como "rs…") e información sobre estos.

Las columnas de un fichero en formato Hapmap son:

- rs# identificadorSNP
- alleles alelo SNP de referencia NCBI dbSNP;
- chrom cromosoma
- pos posición del SNP en el cromosoma
- strand Orientación de la cadena (forward +, reverse -) en el ADN
- assembly# Versión de secuencia de referencia de ensamblado en NCBI
- center nombre del genotipo centro que produce el genotipo
- protLSID Identificador de HapMap protocol;
- assayLSID Ensayo hapmap
- panelLSID Identificador de panel de individuos genotipado
- QCcode Control de calidad
- El resto de columnas representa una persona de los individuos secuenciados (representados por etiquetas NAxxxxxxx)

Los datos de la tabla contienen los nucleótidos de la persona en el SNP de cruce

Como ejemplo

2FORMATO HAPMAP

rs#	alleles	chrom	pos	NA06984	NA06985	NA06986	NA06989
rs4978242	C/T	chr15	18266878	NN	CC	NN	NN
rs6599753	C/T	chr15	18274994	NN	CT	NN	NN
rs7495378	A/G	chr15	18275165	NN	GG	NN	NN
rs7494890	C/T	chr15	18275409	TT	CT	TT	TT
rs12900938	C/T	chr15	18294933	TT	TT	TT	TT
rs8028850	C/T	chr15	18305964	NN	TT	NN	NN
rs12443141	A/T	chr15	18309845	NN	AT	NN	NN

Como existen más nucleótidos que personas secuenciadas en todo el experimento, el formato parece adecuado para la estructura estándar de base de datos, en la que hay muchas más filas que columnas.

Sin embargo, no es el adecuado para los algoritmos que se van a utilizar.

Se debe modificar la estructura de los datos para adecuarlos como entradas del problema que queremos resolver.

3. Funcionalidad de los aplicativos

Tecnologías utilizadas en el desarrollo

Dentro de las opciones para desarrollar las funcionalidades pedidas como objetivos en el proyecto se permitió al desarrollador, en acuerdo con el profesor colaborador la elección de las herramientas de desarrollo.

El conocimiento de las herramientas de desarrollo inicial del desarrollador era muy básico al iniciarse el proyecto, teniendo poca experiencia en proyectos en Python, y ninguna en los frameworks que finalmente han sido seleccionados.

Tanto es así que una parte importante del tiempo de desarrollo ha debido emplearse en el conocimiento y experimentación con las herramientas de desarrollo elegidas.

Lenguaje principal

Se ha escogido Python como lenguaje de programación para este proyecto debido a su capacidad de tratar con grandes datos con rapidez y facilidad de uso.

Python es un lenguaje multipropósito con una gran diversidad de frameworks, que lo hacen adaptable y le permiten la creación de una variedad muy amplia de aplicaciones.

Además resulta portable entre diferentes arquitecturas y sistemas operativos, permitiendo una migración a otros entornos en un tiempo muy corto.

La elección de Python permite la elección de librerías como pandas (Python data análisis library [5]) para tratar con datos de alta dimensionalidad de forma muy fácil. Esto es ideal para el tratamiento de los ficheros de datos Hapmap que vamos a leer y procesar en este proyecto.

Desarrollo de aprendizaje automático

La librería de Python utilizada para llevar a cabo el análisis de aprendizaje automático ha sido scikit-learn [6].

Esta librería nos proveé de los estimadores en los que el proyecto está interesado, que son los de clasificación y agrupamiento, así como un amplio repertorio de funcionalidad, como la evaluación de métricas asociada que cubre los requerimientos del proyecto.

Otras librerías

Además Python proveé librerías como matplotlib, con las que desarrollar gráficos de gran complejidad con una parametrización mínima.

Esto resulta adecuado también para este proyecto y otros muchos de estadística y biología, en el que mostrar la información de forma gráfica resulta de gran ayuda para su fácil y rápida interpretación.

Desarrollo Web

Dentro de las amplias posibilidades que ofrece Python para la presentación de datos se incluyen diferentes frameworks de generación de aplicaciones con interfaz Web.

En el planteamiento del proyecto se discute la posibilidad de presentar una herramienta Web para proveer a los usuarios finales de una interfaz de uso sencilla y sin necesidad de conocimientos técnicos para utilizar las funcionalidades provistas en el proyecto.

Además, la creación de una interfaz Web permite, a través de su despliegue la ejecución remota de las funcionalidades, como un servicio, evitando así la necesidad completa de instalación de ningún software específico.

En la elección del framework de desarrollo Web, se barajaron diferentes opciones, entre ellas Flask y Django.

Esta última fue la elección realizada. Se llevó a cabo una evaluación de ambos frameworks, y aunque los dos resultaban más que adecuados para el desarrollo de la parte Web, debido a la disponibilidad de recursos de aprendizaje por parte del desarrollador del framework Django, éste fue el elegido en última opción.

Para la presentación de la aplicación Web utilizando características de html5 se han utilizado las librerías de CSS y Javascript del framework Bootstrap [7].

Además, estas librerías permiten desarrollar de manera fácil una aplicación que pueda mostrarse en dispositivos con diferentes formatos de pantalla independientemente del tamaño de la pantalla del dispositivo.

Siguiendo las normas de html5 y responsive con bootstrap y una mínima adaptación del código html se puede conseguir que la aplicación sea utilizable de forma cómoda en pantallas también de pequeño formato.

Almacenaje de datos : Spark vs BD relacional vs Ficheros planos

Para nuestro modelo de datos, los datos de los SNP son almacenados en columnas, como variables características de un modelo de aprendizaje automático.

Este es el formato adecuado para que los algoritmos de aprendizaje automático procesen los datos y puedan llevar a cabo predicciones.

Por desgracia, la conversión de nuestros ficheros a un formato con estas características plantea problemas ya que el número de SNP por cromosoma resulta muy grande para su gestión como tablas de base de datos relacionales al uso.

Se pretende conseguir la gestión de los datos masivos de Hapmap en caso que con los sistemas tradicionales de RDBMS no se alcancen los requisitos de acceso a datos y velocidad adecuados.

En las bases de datos tradicionales, existen más restricciones en lo que se refiere a número de columnas en las tablas que en número de filas.

Así por ejemplo, para postgres , el número de columnas queda limitado a 250..1600 columnas, en función del tipo de datos almacenado[8].

En Oracle (u oracle express, que sería el adecuado para el desarrollo del proyecto), 1000 es el límite máximo de columnas permitidas[9].

Para evaluar el almacenaje en base de datos se realiza la instalación de postgres 9.4 y el driver de Python psycopg2.

Se han obtenido erroes que desaconsejan la utilización de bd relacionales para la realización del proyecto.

Por ejemplo, para las pruebas realizadas en postgres, al intentar cargar un fichero de datos con número de filas reducido, aunque gran número de columnas (SNP's):

```
Traceback (most recent call last):
  File "d:\Anaconda3\lib\site-packages\sqlalchemy\engine\base.py", line 1139, in _execute_context
    context)
  File "d:\Anaconda3\lib\site-packages\sqlalchemy\engine\default.py", line 450, in do_execute
    cursor.execute(statement, parameters)
psycopg2.OperationalError: las tablas pueden tener a lo más 1600 columnas
```

3 NÚMERO MÁXIMO DE COLUMNAS

- Se analiza también la posibilidad de utilizar Spark 2.0.1 y hadoop 2.7

Para evaluación de la idoneidad de este sistema de almacenaje y acceso a datos, se han instalado versiones de Spark, que contiene Hadoop como versiones alternativas al uso de base de datos tradicional, con la siguiente motivación:

- Problemas de los rdbms analizados para contener los datos en el formato de tabla que necesitamos
- Velocidad en la gestión de los datos de Hapmap
- Potencial posibilidad de utilización de librería ML de Spark

Solución adoptada

Para superar esta limitación, se ha evaluado la opción de cargar directamente los ficheros planos desde el sistema de ficheros estándar del sistema operativo.

Estos serán procesados dentro de una estructura de datos Python, a través de la librería pandas.

Al cargar varios ficheros de población y cromosoma a la vez, sólo deben cargarse los SNP comunes para poder generar la estructura de datos necesaria para que todos las columnas/variables SNP contengan datos y el análisis de aprendizaje automático se realice de forma adecuada.

Técnicamente, se realiza primero un escaneado de las cabeceras de los ficheros a cargar.

- En estas cabeceras se identifican los SNP de cada fichero.
- Se seleccionan solo los SNP comunes en todos los ficheros a cargar y se guardan en un conjunto.
- Posteriormente, se cargan los datos especificando en la llamada a la función de carga que sólo deben cargarse las columnas (SNP) comunes.

Hacer notar que a más ficheros, menos columnas comunes, y por tanto menos datos por fichero son cargados en memoria.

Descripción de Algoritmos ML utilizados

Se describen cada uno de los algoritmos utilizados y los parámetros que se permite elegir en la aplicación.

Cada uno de los algoritmos podría dar para una tesis en sí mismo, aunque en este caso se describirán de modo simple en unas pocas frases, ya que el objetivo de este trabajo es realizar una comparación funcional entre ellos basándose en datos del proyecto Hapmap.

Algoritmos de Clasificación

Dentro de la aplicación, se han implementado y evaluado los siguientes algoritmos de clasificación:

- Knn Nearest Neighbors : Para los datos de test, calcula la distancia con los datos de entrenamiento, y asigna la clase común en los “k vecinos” más cercanos. Tenemos como parámetros
  - vecinos : Número de vecinos a considerar para asignar clase
  - Tamaño hoja : En la implementación scikit [10] Número de muestras a partir de la cual el estimador cambia su algoritmo de comparación de distancias de uno basado en árbol a comparación de todos contra todos.

- SVM con kernel lineal : Support Vector machine con kernel lineal. Intentará separar el conjunto de datos utilizando una función hiperplano lineal.
  - C: Valor de coste. Se penaliza al error multiplicándolo por este valor. Esto permite que valores queden clasificados en una clasificación errónea, pero “suaviza” los bordes de separación de clasificaciones, permitiendo un mejor ajuste final.

- SVM con kernel radial: Support Vector machine con kernel Gausiano radial. Intentará separa el conjunto de datos por una función Gaussiana.
  - Gamma: Coeficiente gamma utilizado en la función gausiana usada por este kernel

- Decision Tree : Genera un modelo en forma de árbol para clasificar el resultado
    - Produndidad Máxima : Produndidad Máxima de las ramas del árbol

- Random Forest : Metaestimador que estima según varios árboles de decisión y varios subconjuntos de entrenamiento y realiza un promedio para mejorar en precisión y reducir el sobreentrenamiento.
    - Número de estimadores : Número de árboles para estimar
    - Produndidad Máxima : Produndidad Máxima de las ramas del árbol
    - Núm. Características : Criterio para decidir cuándo el árbol partirá en dos ramas. Para simplificar en el desarrollo, en este caso será el número máximo de características en cada rama.

- AdaBoost : Metaestimador que estima según árboles de decisión. Clasifica el conjunto de entrenamiento original, y posteriormente reclasifica realizando copias del estimador original, pero cambiando pesos de casos clasificados erróneamente, con la finalidad de centrarse en estos casos erróneos.

- Naive Bayes: Implementación del algoritmo de clasificación Gaussian Naive Bayes. Asume que todas las características del conjunto de datos son igual de importantes y calcula la probabilidad de las variables salida basándose en la evidencia provista por las variables de entrada.

- QDA (Quadratic Discriminant Analysis) : Análisis discriminante cuadrático. Basado en el teorema de probabilidad condicionada de Bayes.

Agrupamiento

En los algoritmos de agrupamiento se han evaluado los datos de la misma manera que en los de clasificación. En este caso un 33% de los datos se han dejado para test, mientras que en el entrenamiento se utilizó un 66%.
Se han evaluado los datos con los siguientes algoritmos de agrupamiento:

- KMeans : Construye k agrupaciones que minimizan la distancia de los puntos en ellas.
    - Número de agrupaciones : Número de agrupaciones a construir.

- MeanShift : Itera buscando centroides de agrupaciones para descubrir
    - Cuartil de ancho de banda: Dicta el tamaño de cada agrupación. En última instancia estima en el número de agrupaciones a construir.

- MiniBatchKMeans : Variante de kMeans que utiliza subconjuntos de entrenamiento para reducir el tiempo de convergencia del estimador.

- Agrupamiento jerárquico : se agrupan los puntos en sucesivas iteraciones
    - De abajo hacia arriba : Cada punto del conjunto de entrenamiento conforma su propia agrupación, y en sucesivas iteraciones se van agrupando puntos.

    - Ward: Minimiza la suma de cuadrados de las diferencias dentro de las agrupaciones.
    - Average linkage : Minimiza el promedio de las distancias entre cada par de agrupaciones

        - Agrupaciones : Número de agrupamientos a realizar
        - Conectividad vecinos

- Spectral : Utiliza valores propios de la matriz de similaridad (que cuantifica la similaridad entre características). Realiza entonces una reducción de la dimensionalidad y lleva a cabo el agrupamiento de las características.
    - Agrupaciones : Número de agrupamientos.

- DbScan : Density-based spatial clustering of applications with noise. Algoritmo de agrupamiento que considera los agrupamientos como áreas de alta densidad de puntos separados por áreas de baja densidad.
    - Eps: Distancia a partir de la cual se considera que un punto queda fuera del núcleo de un área densa.

- Affinity Propagation: Algoritmo de agrupamiento que estima el número de agrupaciones utilizando el factor de Damping y el factor preferencia. Los puntos se convierten en objetos que almacenan dos vectores, uno de afinidad con los puntos vecinos, y uno de evidencia acumulada de que un vecino es su ejemplar(el punto representativo del cluster).

    - Factor Damping: Factor de suavizado de oscilaciones en el cálculo de los vectores de afinidad y responsabilidad [11]
    - Preferencia : Número de ejemplares

- Birch : algoritmo de agrupamiento basado en árbol de características y subcaracterísticas de estas.
    - Agrupaciones : Número de agrupaciones

Desarrollos realizados

En el principio se partió de datos provistos por el proyecto Hapmap.

Estos resultaron ser ficheros de texto con tablas en un formato que resultaba inadecuado para su procesado por los algoritmos de aprendizaje automático tal como se necesitaban.

Se requería un proceso de conversión inicial de estas tablas,

Conversión de ficheros Hapmap

Procesado de datos

Para realizar pruebas sobre los scripts de conversión de datos Hapmap a formato requerido en el desarrollo, se han preprocesado todos los ficheros de poblaciones para cromosomas 1, 15, y 22.

Con la finalidad de realizar pruebas sobre pocas poblaciones y varios cromosomas, para las poblaciones MEX y ASW también se han generado los ficheros de todos sus cromosomas.

Debido al gran volumen de datos a procesar considerando todos los cromosomas, se ha considerado que este subconjunto es adecuado para cumplir con los objetivos que son:

1.- garantizar la corrección de los productos de software

2.- consecución de conclusiones y predicciones a través de los algoritmos de aprendizaje automático

Estos datos procesados serán utilizados también en el desarrollo de los scripts de aplicación de algoritmos AA.

A partir del script Preproceso.py, que llama a FCHapmap.py, se convierten los ficheros de datos originales de Hapmap, a su versión para poder ser utilizados por los algoritmos de AA

En la siguiente tabla se muestran los datos de los ficheros convertidos para el cromosoma 15.

Nombre Original	Tamaño original	Nombre Procesado	Tamaño procesado
genotypes_chr15_ASW_r28_nr.b36_fwd.txt	21.453.291	15_ASW.txt	8.395.505
genotypes_chr15_CEU_r28_nr.b36_fwd.txt	78.965.800	15_CEU.txt	38.872.680
genotypes_chr15_CHB_r28_nr.b36_fwd.txt	67.696.028	15_CHB.txt	31.429.078
genotypes_chr15_CHD_r28_nr.b36_fwd.txt	20.650.050	15_CHD.txt	8.930.163
genotypes_chr15_GIH_r28_nr.b36_fwd.txt	20.921.193	15_GIH.txt	8.796.532
genotypes_chr15_JPT_r28_nr.b36_fwd.txt	59.812.081	15_JPT.txt	26.395.878
genotypes_chr15_LWK_r28_nr.b36_fwd.txt	24.074.791	15_LWK.txt	10.403.655
genotypes_chr15_MEX_r28_nr.b36_fwd.txt	19.839.865	15_MEX.txt	7.729.221
genotypes_chr15_MKK_r28_nr.b36_fwd.txt	34.553.340	15_MKK.txt	17.089.180
genotypes_chr15_TSI_r28_nr.b36_fwd.txt	20.993.228	15_TSI.txt	8.927.631
genotypes_chr15_YRI_r28_nr.b36_fwd.txt	88.967.764	15_YRI.txt	45.646.808

4 CONVERSIÓN CHR 15

De cada uno de los ficheros de genotipo originales, obtenemos un fichero donde se clasifica el alelo de referencia en dbSNP como 1, con 0 y 2 para los homocigotos.

Se ha incluido la corrección para datos “No disponibles” e indels, sustituyendo estos por el valor de la **moda** del SNP en el fichero de la población.

Datos Procesados

La estructura queda como sigue: las filas contienen datos de personas. Las columnas contienen los nucleótidos del SNP identificado en la columna.

Como referencia, se muestra un ejemplo del estilo de fichero procesado resultante:

Uid	rs7494890	rs12900938	rs12905389	rs34612657	rs9744388	rs9744157	rs10163108	rs6599770	rs28757158	rs28757152	rs7171651	rs28364489	rs28364521	rs28757158	rs28757152	rs17875506
NA19663	1	2	2	0	0	0	1	2	2	2	1	2	0	2	2	2
NA19664	1	1	2	0	0	0	1	1	0	1	1	2	1	0	1	2
NA19665	0	2	2	0	0	0	2	2	2	1	0	2	2	2	1	2
NA19722	2	2	2	0	0	0	0	2	2	0	2	2	1	2	0	2
NA19723	2	2	1	0	0	0	0	1	2	1	2	2	0	2	1	2
NA19649	1	2	2	0	0	0	0	2	2	2	2	2	1	2	2	2
NA19669	2	2	1	0	1	0	0	1	2	0	2	2	1	2	0	2
NA19656	0	2	2	0	0	0	1	2	2	1	1	1	1	2	1	2
NA19657	2	2	2	0	0	0	1	2	1	2	1	1	1	1	2	2
NA19658	2	2	1	0	1	0	0	1	0	2	2	2	0	0	2	2

5 EJEMPLO DE FICHERO PROCESADO

Script Creación de ficheros para tablas de datos

Enumero nombre y función de los ficheros utilizados para la conversión de datos Hapmap

FCHapmap.PY : "Filter and cleaning Hapmap Data".

Su función principal es la de procesar los ficheros de genotipos Hapmap conviertiendo de formato Hapmap al formato de fichero esperado para la utilización con los algoritmos de aprendizaje propuestos

- Procesa datos de fichero primer parámetro y devuelve los resultados en el fichero segundo parámetro.
- Para su ejecución vía prompt en consola del sistema operativo, permite además entrada por stdin y salida por stdout.
- Nuestra tabla procesada tendrá una estructura en columnas del estilo (IdPersona, sn1,…snpN)

SCRIPT Preproceso.PY

Itera sobre los ficheros de genotipos para los cromosomas y las 11 poblaciones y realiza una llamada al proceso de filtrado y limpiado en FCHapmap para cada uno de estos ficheros.

Se generan a través de él los ficheros con datos preprocesados. Ejemplo:

- 15_ASW.txt
- 15_CEU.txt
- 15_CHB.txt
- 15_CHD.txt
- 15_GIH.txt

- 15_JPT.txt
- 15_LWK.txt
- 15_MEX.txt
- 15_MKK.txt
- 15_TSI.txt
- 15_YRI.txt

Implementación de algoritmos de ML

La librería de Python utilizada para llevar a cabo el análisis de Aprendizaje automático ha sido scikit-learn [6].

Una parte importante del tiempo de Fase 2 se ha invertido en el conocimiento de las librerías scikit-learn de aprendizaje automático y a la implementación de un proceso que aplique técnicas de aprendizaje automático sobre el conjunto de datos Hapmap.

Dentro del proyecto de interfaz Web y utilidad de tests Batch se ha integrado el módulo algML.py . Este contiene la funcionalidad desarrollada de aprendizaje automático.

algML.py

Este script se encarga de cargar los datos Hapmap en formato procesado y de aplicar sobre ellos los diferentes algoritmos de clasificación y agrupamiento que se soliciten por parámetro.

Los pasos del script se definen como sigue:

1. Carga de ficheros
2. Generación de conjuntos de entrenamiento y test
    a. 67% para entrenamiento, parametrizable desde genvars.py
3. Plot representación del conjunto
    a. Análisis de PCA. Reducción a dos componentes.
    b. Graficado en 2D (dos componentes principales)
4. Entrenamiento de los algoritmos seleccionados
5. Estimación del conjunto con los algoritmos de clasificación y agrupamiento
    a. Impresión del bloque de métricas de agrupamiento
    b. Visualización gráfica de los conjuntos resultados en 2D
    c. Visualización de curvas ROC para algoritmos de clasificación.

Métricas

Matriz de Confusión

Utilizada principalmente en algoritmos de clasificación nos confronta los casos predichos contra los verdaderos en una matriz.

Los casos correctamente clasificados se encuentran en la diagonal, mientras que los incorrectos, y el error cometido puede verse fuera de ella. Como ejemplo

Predicted	ASW	MEX	__all__
Actual			
ASW	28	0	28
MEX	0	31	31
__all__	28	31	59

6 EJEMPLO DE MATRIZ DE CONFUSIÓN

Valores de precisión y recall

Aunque solo se muestran en la parte de consola de servidor, incluyo también una referencia al cálculo que se realiza sobre los valores de precisión y recall. En este caso la interpretación no es directa ya que deben convertirse los valores 0 y 1 a las etiquetas de la población que se están considerando.

	precision	recall	f1-score	support
0	0.97	0.94	0.95	32
1	0.93	0.96	0.95	27
avg/total	0.95	0.95	0.95	59

7 METRICAS PRECISIÓN, RECALL, F1

Métricas de puntuación

Se ha programado la visualización de diferentes funciones de evaluación, incluidas en scikit learn.

Estas medidas son usadas preferentemente en algoritmos de agrupamiento, ya que son capaces de entender que el valor de los aglomerados pueden no coincidir con las etiquetas verdaderas. Cuando las etiquetas predichas están permutadas con las verdaderas, estas métricas consideran los valores por grupo [12].

- adjusted_rand_score : índice de ajuste Rand . Mide la similaridad de valores en los dos vectores, de valores verdaderos y de resultados. Devuelve 1 para el ajuste perfecto
- v_measure_score : Medida v
    - Se define la homogeneidad como que un agrupamiento solo tiene miembros de una clase.
    - Se define completitud como que todos los miembros de una clase están asignados al mismo agrupamiento [13].
    - La media armónica de estos valores define la puntación de la medida v

    - Permite una interpretación intuitiva en términos de qué tipos de clasificaciones se están realizando mal.
- adjusted_mutual_info_score : Mide la dependencia mutua entre las variables verdadera y predicha. Valores predichos al azar tendrán una puntuación de 0, y un ajuste perfecto de 1. Tiene un orden de magnitud menor al índice de ajuste Rand
- mutual_info_score

Curvas Roc

Para los algoritmos de clasificación el aplicativo genera curvas ROC.

Ejemplos de estas pueden verse en los anexos.

Cuando se clasifican más de dos poblaciones, la clase positiva sobre la cual se calcula la curva ROC será la anotada en último lugar.

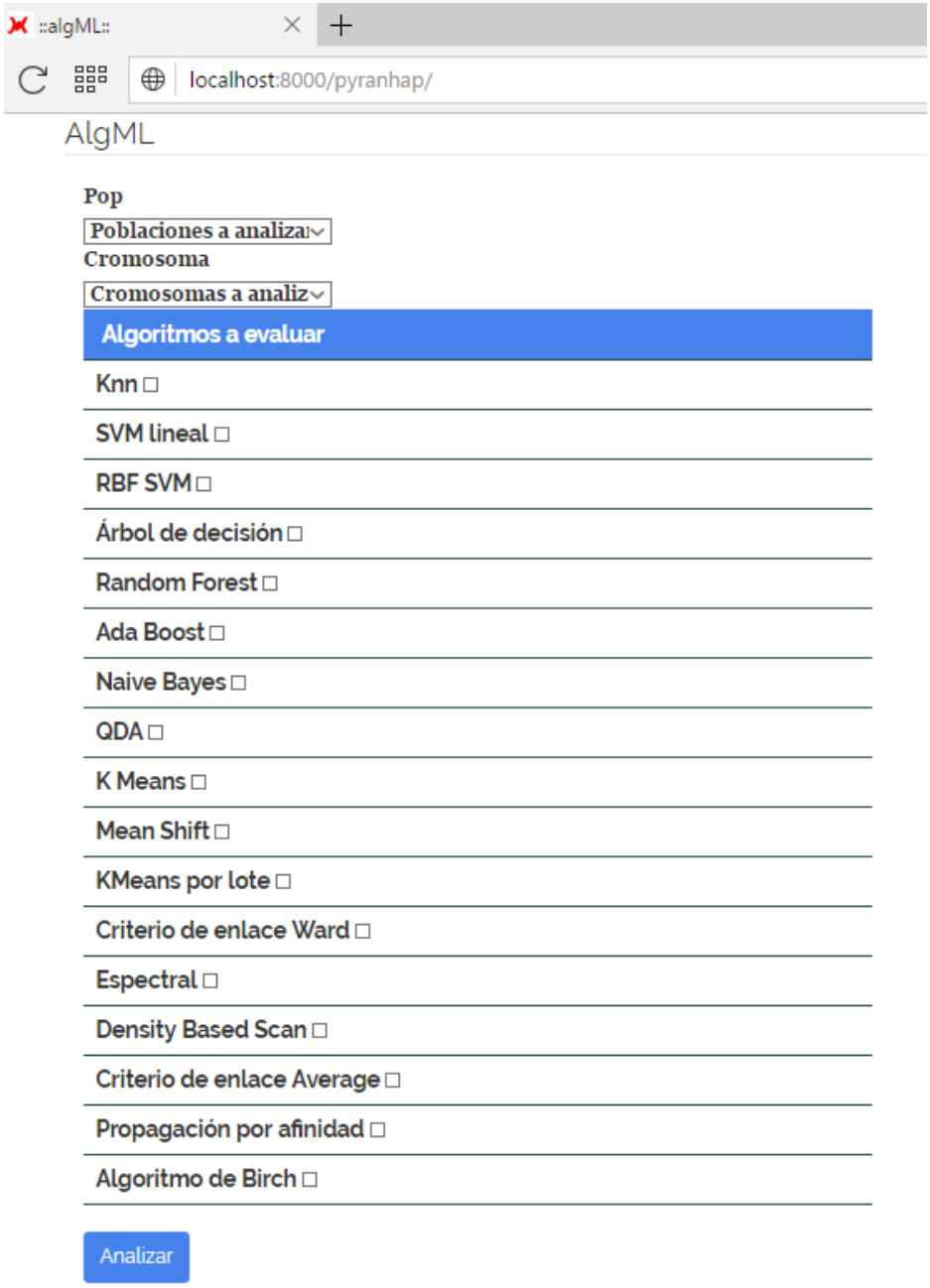
Para los algoritmos de agrupamiento, para poder evaluar la calidad de los parámetros se lleva a cabo una comparación entre las agrupaciones predichas y las etiquetas.

Aplicación con Interfaz web

La aplicación web se accede a través del patrón url

http://servidor:8000/pyranhap

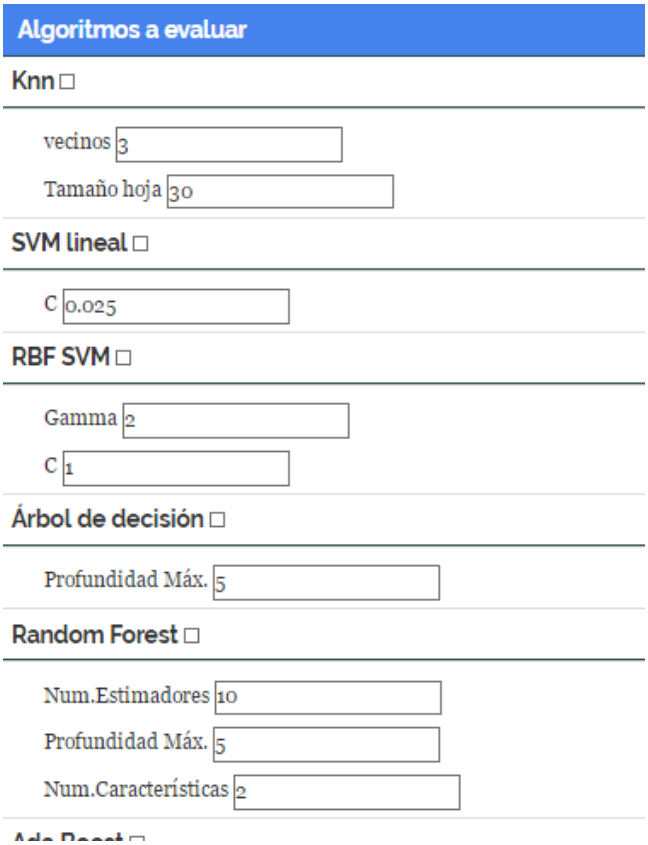
Este es el formulario principal de la aplicación:



8 FORMULARIO PYRANHAP

Los parámetros disponibles se despliegan pulsando sobre el nombre del algoritmo

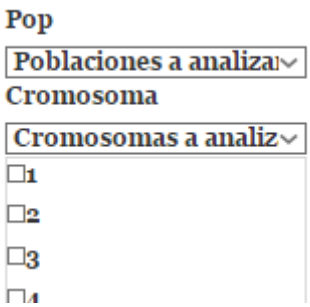
Se muestran así:



9 SELECCIÓN DE PARÁMETROS

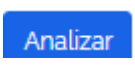
Para seleccionar uno o varias poblaciones para su evaluación, se despliega el control de población y se marcan las deseadas.

Para los cromosomas se procede de forma análoga.



10 SELECCIÓN DE POBLACIONES Y CROMOSOMAS

En esta configuración se ha utilizado Html 5 con Bootstrap y para modularizar el código, inclusión de ficheros (“includes”) en plantillas de Django con Jinja.

Al pulsar el botón 

, a través de una petición mediante Ajax se pasa el control a Python y a partir de aquí se ejecutan los algoritmos de aprendizaje automático seleccionados.

Mientras tanto se muestra en pantalla un mensaje en el que el usuario es informado de que el resultado puede tardar:

11 MENSAJE DE ESPERA

Una vez completado el análisis, se muestran los resultados bajo el formulario:

Contiene los resultados implementados por los algoritmos de aprendizaje automático:

- Representación en 2D del conjunto de datos
- Métricas de puntuación
- Matriz de confusión
- Gráfico de predicción de cada algoritmo seleccionado de puntos de test
    - Para tener una imagen completa de las poblaciones:
        - En color más claro se muestran los puntos del conjunto de entrenamiento
        - En color más oscuro los puntos de test, cada color según la clasificación predicha por el algoritmo
- Curvas ROC para los algoritmos de clasificación.

Utilización en dispositivos móviles

El formulario y la respuesta se han llevado a cabo teniendo en cuenta su posible utilización en dispositivos móviles.

Durante el desarrollo se han utilizado librerías del framework Bootstrap para crear un formulario Responsive, que permite su utilización sin necesidad de cambios en la parte web.

Aunque la aplicación Web puede mostrarse en navegadores móviles falta adaptar algunos de los div de Html para que utilicen clases Bootstrap y tener una aplicación plenamente Responsive.

Parámetros de la aplicación

La aplicación permite seleccionar diferentes parámetros globales sin necesidad de modificar código.

Se han desarrollado mecanismos de carga de parámetros al inicio de la aplicación y también en la carga de formulario, que permiten su parametrización de forma externa al código.

Enumero estos ficheros y su funcionalidad:

default_params.Json

En la parte de algoritmos, podemos seleccionar los parámetros por defecto para el uso de los algoritmos a través del fichero

- default_params.json

En este podemos seleccionar los cromosomas y poblaciones por defecto para cargar en los análisis, así como los parámetros por defecto que tomarán los algoritmos si no se escogen desde el formulario.

La elección del formato json es adecuada para esta finalidad, ya que su carga resulta en un objeto utilizable por Python.

Algs-params.yaml

Asimismo, para los valores y etiquetas de los algoritmos que se mostrarán en formulario, se ha creado un fichero

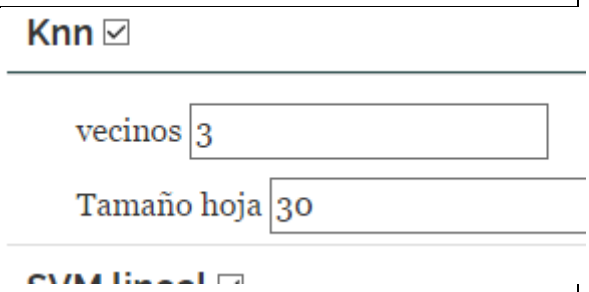
- algs-params.yaml

Este permite modificar, en un formato legible y modificable fácilmente para personal no técnico el valor de parámetros o etiquetas.

Sus cambios se muestran inmediatamente, en la siguiente carga de formulario.

Así por ejemplo

3 PARÁMETROS FORMULARIO

Los Valores	Se muestran como
params : - parent_id : knn name : n_neighbors label : vecinos val : 3 tipo : integer - parent_id : knn name : leaf_size label : Tamaño hoja val : 30 tipo: integer	

Aplicación en modo Batch

Con la finalidad de realizar a cabo análisis masivos de casos se ha desarrollado el script batch.py.

Este script permite la ejecución y generación de informe para casos de análisis de forma masiva, mediante su parametrización en un fichero json de tests.

Su modo de uso es :

batch.py –i fichero_test.json

Sin parámetros, se utilizará el fichero por defecto tests.json

En este fichero se parametrizan las poblaciones y cromosomas sobre los que realizar el análisis, así como los algoritmos y los parámetros a analizar.

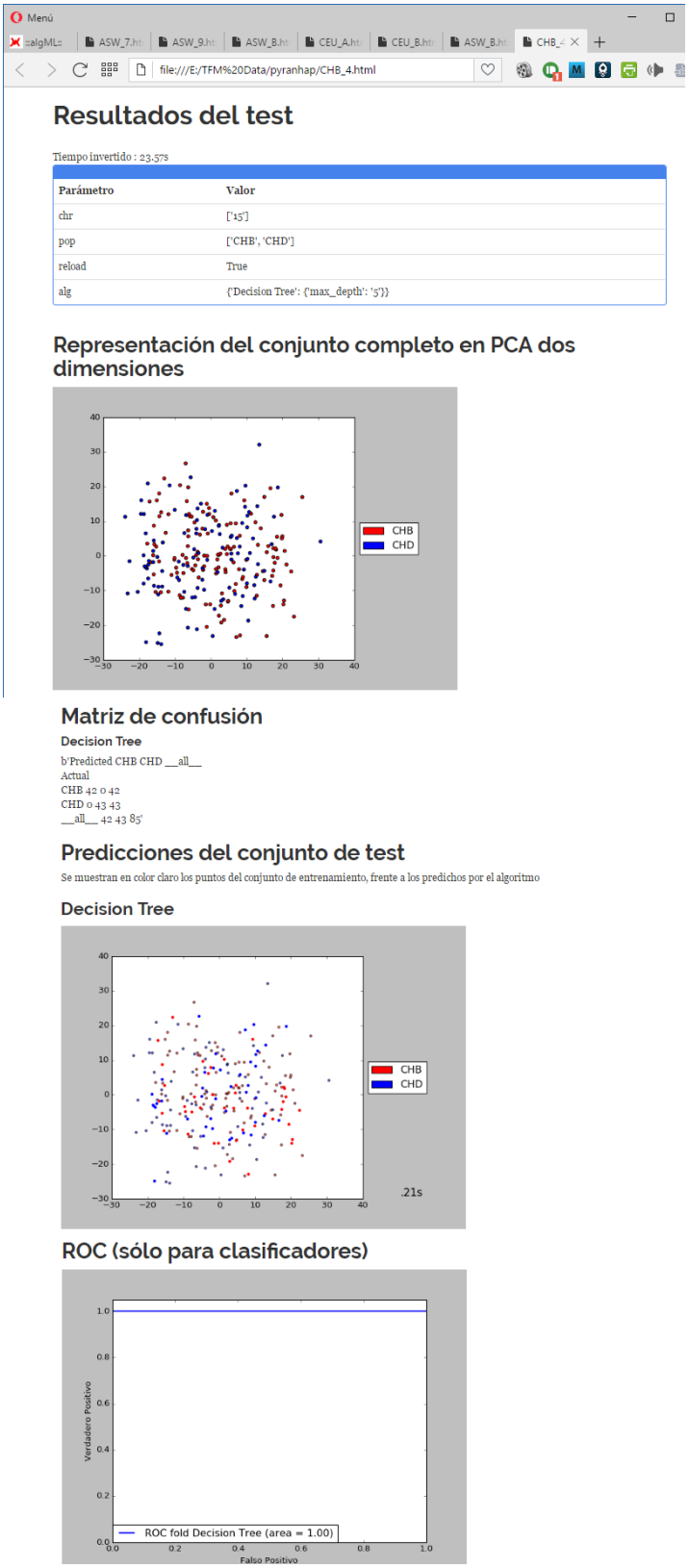
El script genera como salida un informe **html** con las salida respuesta similar a la mostrada por el interfaz Web.

Esto permite por una parte, generar casos de análisis para diferentes algoritmos y parámetros, y por otra la comprobación de algoritmos y parámetros como control de calidad de las rutinas de análisis de aprendizaje automático.

Así un fichero de tests de ejemplo puede contener claves :

- name : Nombre del fichero Html de salida
- Tests : Tests a ejecutar, con parámetros
    - pop : poblaciones a considerar
    - chr : cromosomas a considerar
    - alg : diccionario de algoritmos y parámetros
    - reload: True o False. Parámetro de sintonización, que permite reutilizar los datos cargados en el test anterior sin necesidad de cargarlos de nuevo

Ejemplo de salida



12 EJEMPLO DE SALIDA BATCH

Se realiza la ejecución de un test para contrastar el algoritmo de árbol de decisión para las poblaciones CHB y CHD.

Pruebas de diferentes ejecuciones

Debido al tamaño de la salida, en el anexo B incluyo una ejecución desde la interfaz web, en la que se muestran para el cromosoma 22 y todas las poblaciones, una ejecución de todos los algoritmos, y su salida mostrada, con el siguiente orden:

- Métricas de clustering aplicadas a todos los algoritmos
- Representación de todas las poblaciones en un gráfico de 2D reducido por PCA
- Métrica de clasificación, Matriz de confusión, para todas las poblaciones
- Representación en 2D de la predicción de los algoritmos. Esta tienen las siguientes características :
    - El conjunto de entrenamiento se muestra con una transparencia del 60%. En esta salida se pretende no mostrar sólo los datos de predicción, sino mostrarlos sobre el conjunto de entrenamiento, para tener una visión global de todo el conjunto
    - Sobre las etiquetas en algoritmos de clustering : Los algoritmos de clustering pueden clasificar los datos sin coincidir con las etiquetas reales, ya que no han sido entrenados con este dato. Así por ejemplo, una población 0 puede aparecer en el grupo 10 de la predicción del algoritmo. Es por eso que en las etiquetas no se muestran los nombres de población, sino un nombre de grupo arbitrario.
- Para los algoritmos de clasificación, la curva ROC. En este caso considerando como clase positiva la última población, YRI.

Affinity propagation y DBSCan me proporcionan siempre resultados muy diferentes a las etitquetas reales. DBScan tiene un comportamiento de “no clasificación”, ya que devuelve -1 para varios de los casos.

Los parámetros de eps y factores de Damping y preferencia para estos algoritmos resulta ser de gran importancia. Aunque en varios ejemplos he visto funcionar con éxito estos algoritmos, para el conjunto de datos Hapmap no parecen obtener buenos resultados.

Vemos que los algoritmos de clasificación para este conjunto de datos están funcionando de una manera bastante aceptable, consiguiendo ratios de acierto que pueden considerarse como buenos.

En el caso de poblaciones separadas, ambos tipos de algoritmos consiguen buenos resultados.

Para poblaciones cercanas, como CHD y CHB los algoritmos de clustering tienen un rendimiento peor que los de clasificación, que continúan obteniendo buenos resultados.

Tiene sentido si examinamos la nube de puntos en el gráfico de dimensiones reducidas: no hay agrupaciones claras, y los algoritmos de clustering tienen dificultad en estos casos.

Comentarios sobre rendimiento

Procesados de datos

En la siguiente tabla se muestran los datos de los ficheros convertidos para el cromosoma 15, y el tiempo invertido en procesarlos

4 TIEMPOS DE CONVERSIÓN

Nombre Original	Tamaño original	Nombre Procesado	Tiempo	Tamaño procesado
genotypes_chr15_ASW_r28_nr.b36_fwd.txt	21.453.291	15_ASW.txt	10.00 segundos	8.395.505
genotypes_chr15_CEU_r28_nr.b36_fwd.txt	78.965.800	15_CEU.txt	50.51 segundos	38.872.680
genotypes_chr15_CHB_r28_nr.b36_fwd.txt	67.696.028	15_CHB.txt	41.87 segundos	31.429.078
genotypes_chr15_CHD_r28_nr.b36_fwd.txt	20.650.050	15_CHD.txt	10.87 segundos	8.930.163
genotypes_chr15_GIH_r28_nr.b36_fwd.txt	20.921.193	15_GIH.txt	10.39 segundos	8.796.532
genotypes_chr15_JPT_r28_nr.b36_fwd.txt	59.812.081	15_JPT.txt	34.67 segundos	26.395.878
genotypes_chr15_LWK_r28_nr.b36_fwd.txt	24.074.791	15_LWK.txt	12.36 segundos	10.403.655
genotypes_chr15_MEX_r28_nr.b36_fwd.txt	19.839.865	15_MEX.txt	9.27 segundos	7.729.221
genotypes_chr15_MKK_r28_nr.b36_fwd.txt	34.553.340	15_MKK.txt	19.72 segundos	17.089.180
genotypes_chr15_TSI_r28_nr.b36_fwd.txt	20.993.228	15_TSI.txt	10.52 segundos	8.927.631
genotypes_chr15_YRI_r28_nr.b36_fwd.txt	88.967.764	15_YRI.txt	59.48 segundos	45.646.808

CARGA DE DATOS

En varias ejecuciones y con diferentes tomas de datos para los ficheros 15_ASW y 15_MEX se obtienen de media estos tiempos de carga:

- Procesado de cabeceras (inmediato)

15_MEX.txt procesando...
15_ASW.txt procesando...

- Carga de datos

chr 15
MEX
Leido 15_MEX en 6.83 segundos
ASW
Leido 15_ASW en 6.78 segundos
chr 15 shape (173, 41067)
Total shape (173, 41067)
22_MEX.txt procesando...
22_ASW.txt procesando...
chr 22
MEX
Leido 22_MEX en 2.29 segundos
ASW
Leido 22_ASW en 2.75 segundos
chr 22 shape (173, 19842)
Total shape (173, 60909)

Hacer notar que a más poblaciones, añade casos (filas), pero reduce el número de columnas comunes, y por tanto menos datos por fichero son cargados en memoria.
En cambio, la adición de cromosomas, añade columnas al conjunto de datos.

ESTIMADORES

En los gráficos de predicción de datos se incluye el tiempo tardado por el estimador en realizar el proceso de aprendizaje.

Cabe recordar que cada una de estas métricas está en función de los datos cargados. Así los tiempos de carga y entrenamiento de dos poblaciones como ASW y MEX del cromosoma 22 serán mucho menores que si hablamos del cromosoma 1.

Conclusiones sobre el proyecto

En este proyecto se ha llevado a cabo el estudio de algoritmos de aprendizaje automático y realizado una comparativa de estos en función de los datos analizados.

El proyecto se ha realizado con datos biológicos reales.

Se ha desarrollado aplicaciones para usuario final, tanto para usuario Web como para línea de comandos

Se han evaluado los algoritmos y se ha obtenido una imagen de sus fortalezas y debilidades utilizando datos de interés.

Se han asimilado conceptos no conocidos de de aprendizaje automático.

La realización de mejores gráficos y métricas hubiese permitido ampliar interpretaciones a las que no se ha podido llegar con la información que la aplicación explotaba.

La presentación en los html devueltos también debe ser mejorada.

Esto ha sido debido en parte a la inclusión de muchos algoritmos para su evaluación. Lo cual ha requerido de una dedicación bastante amplia para realizar pruebas y ensayos sobre el código.

Aún así, sería deseable la integración de más algoritmos, y la realización de más tests, profundizando en las particularidades de cada algoritmo, esto es, realizando más variaciones en los parámetros llevando a cabo un afinado pueden detectarse variaciones en la solución, y llegar así a mejoras no esperadas.

Por otra parte, con la utilidad de batch hemos ido un poco más allá de lo solicitado inicialmente en los objetivos del proyecto. Esta utilidad se ha mostrado de gran utilidad a la hora de llevar a cabo comparativas de los datos arrojados por los algoritmos, permitiendo de forma automática la generación de informes para comparar a posteriori.

Por otro lado, los datos disponibles de Hapmap nos imposibilitan la opción de preguntar más allá de la variable población. El proyecto podría resultar más ambicioso y excitante con la posibilidad de obtener conclusiones sobre cuestiones de relevancia biológica.

Mejoras en las interfaces de usuario también serían deseables para conseguir una mejora en la experiencia de uso.

Una reorganización de los datos de salida, aunque es sencilla de realizar, también ha quedado fuera del ámbito de lo posible a realizar en el proyecto

Por otra parte, la planificación inicial se ha visto modificada por varios factores, entre ellos el más importante el desconocimiento de las herramientas de desarrollo, desde Python, pasando por scikit learn y terminando con Django.

La distribución de las horas de desarrollo también se ha modificado con respecto a la propuesta inicialmente y esto ha llevado a un cierto retraso en el inicio del desarrollo.

Glosario

Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

Algoritmo de Clasificación: Procedimiento de agrupación automática de una serie de variables a partir de un entrenamiento previo con datos ya clasificados.

Algoritmo de Agrupamiento : Procedimiento de agrupación automática de una serie de variables de acuerdo a un criterio.

Haplotipo : combinación de alelos de diferentes loci transmitidos juntos.

SNP : Polimorfismo de nucleótido único, incluyendo inserciones y deleciones, que se dá en al menos un 1% de la población

Git : Software de control de vesiones diseñado inicialmente por Linus Torvalds para el sistema operativo GNU/linux.

Genotipo : Información genética de un organismo en forma de ADN.

CSS: Cascade Style Sheet. Hoja de estilo en cascada. Lenguaje utilizado para la presentación de documentos HTML.

Javascript : Lenguaje de programación generalmente utilizado para Web en el lado del navegador.

Diseño Responsive: Técnica de diseño que busca la correcta visualización de páginas Web en diferentes dispositivos.

Bibliografía

[1] Documento de Descripción de trabajo de final de Máster . Area3_TFM_ProgramaciónBioinformática_20161.pdf

[2] http://www.sanger.ac.uk/resources/downloads/human/hapmap3.html , Octubre 2016

[3] http://www.hapmap.org , octubre 2016

[3] https://www.ncbi.nlm.nih.gov/variation/news/NCBI_retiring_HapMap/ , Octubre 2016

[4] https://www.ncbi.nlm.nih.gov/SNP/ , Octubre 2016

[5] : http://pandas.pydata.org , Noviembre 2016

[6] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[7] http://getbootstrap.com , Diciembre 2016

[8] https://www.postgresql.org/about/ , Octubre 2016

[9] https://docs.oracle.com/cd/B28359_01/server.111/b28320/limits003.htm

[10] : http://scikit-learn.org/stable/modules/neighbors.html

[11] http://www.psi.toronto.edu/affinitypropagation/faq.html , Dicembre 2016

[12] http://scikit-learn.org/stable/modules/classes.html#clustering-metrics , Diciembre 2016

[13] http://scikit-learn.org/stable/modules/generated/sklearn.metrics.v_measure_score.html#sklearn.metrics.v_measure_score , Diciembre 2016

Anexos

A. Entorno de trabajo

Un primer paso para poder desarrollar el software que lleve a cabo las tareas que necesitamos consiste en la instalación de un entorno de trabajo que contenga las herramientas necesarias para el desarrollo.

Lenguaje de programación

En la preparación del entorno de desarrollo se ha instalado:

- Anaconda 1.3.1 : Gestión de paquetes y entornos de Python
    - IDEs adicionales incluidos: Jupyter notebook 4.2.3 y Spyder 3.0.0
- Python (múltiples versiones a través de Anaconda)
- Soporte para Shell scripting de unix a través de Cygwin 64

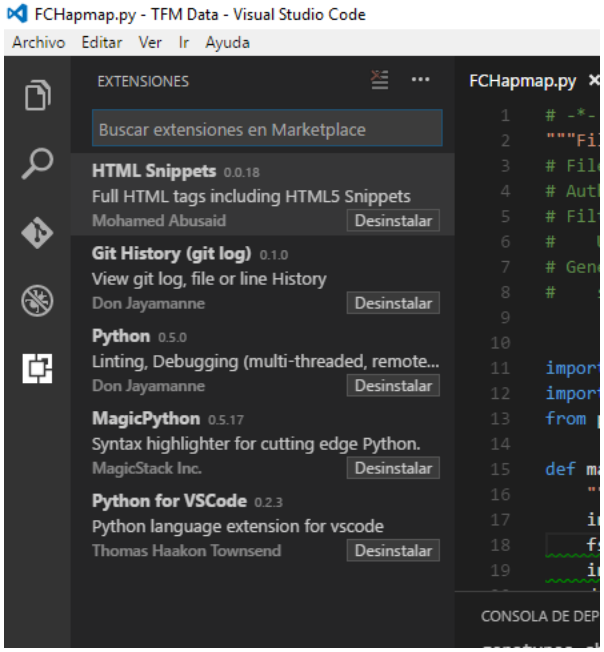
Bases/ Almacenamiento de datos

No ha sido la solución elegida para el almacenamiento de datos, pero se han instalado para evaluar su viabilidad de uso los siguientes softwares

- Postgress 9.4
    - Psycopg2 : Driver para Python
    - Oracle Sqldeveloper : Ejecución de código sql
    - Pgadmin III : Gestión de Base de datos
- Spark 2.01 y hadoop 2.7

Setup de entorno de programación preferido

- Para el desarrollo de las rutinas de aprendizaje automático
    - Spyder 3.0.0
- Para el desarrollo del Preproceso de datos e Interfaz Web
    - Microsoft Visual Studio Code VScode
        - Con Extensiones
            - Python : Debug e intellisense
            - MagicPython : marcador de sintaxis
            - Python for VSCode : Extensión para vscode



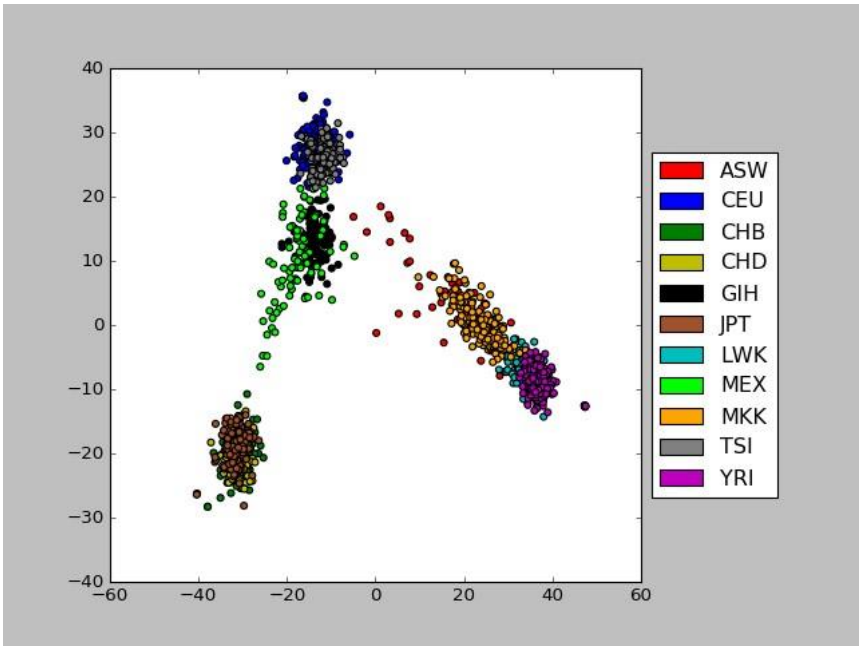
B Ejecución por Interfaz Batch

Resultados del test

Tiempo invertido : 1178.54s

Parámetro	Valor
pop	['ASW', 'CEU', 'CHB', 'CHD', 'GIH', 'JPT', 'LWK', 'MEX', 'MKK', 'TSI', 'YRI']
reload	True
al g	{'AdaBoost': {}, 'DBScan': {'eps': '0.2'}, 'Affinity Propagation': {'damping': '0.5', 'preference': '11'}, 'Average linkage': {'n_clusters': '11','n_neighbors': '10'}, 'MeanShift': {'quantile': '0.3'}, 'Spectral': {'n_clusters': '11'}, 'Decision Tree': {'max_depth': '5'}, 'KMeans':{'n_clusters': '11'}, 'Birch': {'n_clusters': '11'}, 'RBF SVM': {'gamma': 'auto', 'C': '1'}, 'Linear SVM': {'C': '0.025'}, 'Ward': {'n_clusters':'11', 'n_neighbors': '10'}, 'Nearest Neighbors': {'leaf_size': '30', 'n_neighbors': '11'}, 'Naive Bayes': {}, 'QDA': {}, 'MiniBatchKMeans':{'n_clusters': '11'}, 'Random Forest': {'max_depth': '5', 'n_estimators': '10', 'max_features': '2'}}
ch r	['22]

Representación del conjunto completo en PCA dos dimensiones



Matriz de confusión

AdaBoost

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 8 0 0 0 1 0 8 3 2 0 5 27

CEU 6 41 0 0 7 0 1 3 1 0 0 59

CHB 0 0 3 2 5 31 0 1 0 0 0 42

CHD 0 0 0 2 2 31 0 0 0 0 0 35

GIH 6 14 0 0 7 1 0 1 0 0 0 29

JPT 0 0 1 2 5 35 0 0 0 0 0 43

LWK 3 0 0 0 0 0 27 0 1 0 6 37

MEX 4 12 1 0 11 0 0 5 0 0 0 33

MKK 4 0 0 0 2 0 29 1 13 0 15 64

TSI 8 16 0 0 3 1 0 3 0 0 0 31

YRI 1 1 0 0 0 0 56 0 5 0 19 82

__all__ 40 84 5 6 43 99 121 17 22 0 45 482

DBScan

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 0 0 0 0 0 0 0 0 0 27 27

CEU 0 0 0 0 0 0 0 0 0 0 59 59

CHB 0 0 0 0 0 0 0 0 0 0 42 42

CHD 0 0 0 0 0 0 0 0 0 0 35 35

GIH 0 0 0 0 0 0 0 0 0 0 29 29

JPT 0 0 0 0 0 0 0 0 0 0 43 43

LWK 0 0 0 0 0 0 0 0 0 0 37 37

MEX 0 0 0 0 0 0 0 0 0 0 33 33

MKK 0 0 0 0 0 0 0 0 0 0 64 64

TSI 0 0 0 0 0 0 0 0 0 0 31 31

YRI 0 0 0 0 0 0 0 0 0 0 82 82

__all__ 0 0 0 0 0 0 0 0 0 0 482

482

Anity Propagation

Predicted 12 CHB CHD ASW CEU GIH JPT LWK MEX MKK TSI YRI __all__

Actual 12 0 0 0 0 0 0 0 0 0 0 0

CHB 42 0 0 0 0 0 0 0 0 0 0 42

CHD 35 0 0 0 0 0 0 0 0 0 0 35

ASW 27 0 0 0 0 0 0 0 0 0 0 27

CEU 59 0 0 0 0 0 0 0 0 0 0 59

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 43 0 0 0 0 0 0 0 0 0 0 43

LWK 37 0 0 0 0 0 0 0 0 0 0 37

MEX 33 0 0 0 0 0 0 0 0 0 0 33

MKK 63 0 1 0 0 0 0 0 0 0 0 64

TSI 31 0 0 0 0 0 0 0 0 0 0 31

YRI 81 1 0 0 0 0 0 0 0 0 0 82

__all__ 480 1 1 0 0 0 0 0 0 0 0 482

Average linkage

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 0 0 23 3 0 1 0 0 0 0 27

CEU 1 0 58 0 0 0 0 0 0 0 0 59

CHB 0 0 0 0 0 42 0 0 0 0 0 42

CHD 0 0 0 0 0 35 0 0 0 0 0 35

GIH 25 0 0 0 0 0 0 2 0 2 0 29

JPT 0 0 0 0 0 43 0 0 0 0 0 43

LWK 0 0 0 37 0 0 0 0 0 0 0 37

MEX 0 28 1 0 0 1 0 0 1 0 2 33

MKK 0 0 0 64 0 0 0 0 0 0 0 64

TSI 0 0 31 0 0 0 0 0 0 0 0 31

YRI 0 0 0 82 0 0 0 0 0 0 0 82

__all__ 26 28 90 206 3 121 1 2 1 2 2 482

MeanShift

MeanShift

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 27 0 0 0 0 0 0 0 0 0 0 27

CEU 59 0 0 0 0 0 0 0 0 0 0 59

CHB 42 0 0 0 0 0 0 0 0 0 0 42

CHD 35 0 0 0 0 0 0 0 0 0 0 35

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 43 0 0 0 0 0 0 0 0 0 0 43

LWK 37 0 0 0 0 0 0 0 0 0 0 37

MEX 33 0 0 0 0 0 0 0 0 0 0 33

MKK 64 0 0 0 0 0 0 0 0 0 0 64

TSI 31 0 0 0 0 0 0 0 0 0 0 31

YRI 82 0 0 0 0 0 0 0 0 0 0 82

__all__ 482 0 0 0 0 0 0 0 0 0 0 482

Spectral

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 3 24 0 0 0 0 0 0 0 0 27

CEU 0 0 54 0 5 0 0 0 0 0 0 59

CHB 2 0 8 0 0 0 8 0 1 23 0 42

CHD 3 0 7 0 0 0 7 0 0 18 0 35

GIH 0 0 29 0 0 0 0 0 0 0 0 29

JPT 26 0 6 0 0 0 9 0 1 1 0 43

LWK 0 2 32 3 0 0 0 0 0 0 0 37

MEX 0 0 33 0 0 0 0 0 0 0 0 33

MKK 0 0 18 27 0 0 0 12 0 0 7 64

TSI 0 0 31 0 0 0 0 0 0 0 0 31

YRI 0 67 13 0 0 2 0 0 0 0 0 82

__all__ 31 72 255 30 5 2 24 12 2 42 7 482

Decision Tree

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 2 0 0 2 0 4 2 10 0 7 27

CEU 0 47 0 3 5 0 0 0 3 1 0 59

CHB 1 1 35 2 0 1 0 2 0 0 0 42

CHD 0 0 8 20 1 3 0 1 1 0 1 35

GIH 0 1 0 2 15 0 0 2 1 8 0 29

JPT 0 0 2 4 1 34 0 1 1 0 0 43

LWK 1 1 0 0 2 0 14 0 7 0 12 37

MEX 1 0 0 5 11 0 0 12 0 4 0 33

MKK 8 2 0 0 5 0 9 0 33 0 7 64

TSI 1 2 0 4 8 0 0 0 1 15 0 31

YRI 2 1 0 1 2 0 3 0 10 0 63 82

__all__ 14 57 45 41 52 38 30 20 67 28 90 482

KMeans

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 8 0 2 1 4 0 0 0 12 0 0 27

CEU 0 0 2 26 0 0 0 0 0 31 0 59

CHB 0 1 0 0 0 0 27 14 0 0 0 42

CHD 0 1 0 0 0 0 25 9 0 0 0 35

GIH 0 0 0 0 0 0 29 0 0 0 0 0 29

JPT 0 8 0 0 0 0 1 34 0 0 0 43

LWK 0 0 0 0 0 0 0 0 0 37 0 0 37

MEX 0 0 0 4 0 0 1 0 0 2 26 33

MKK 56 0 0 0 0 0 0 0 8 0 0 64

TSI 0 0 0 17 0 0 0 0 0 14 0 31

YRI 0 0 0 0 79 0 0 0 3 0 0 82

__all__ 64 10 4 48 83 29 54 57 60 47 26 482

Birch

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 1 2 0 13 0 0 7 0 0 4 0 27

CEU 0 59 0 0 0 0 0 0 0 0 0 59

CHB 0 0 42 0 0 0 0 0 0 0 0 42

CHD 0 0 35 0 0 0 0 0 0 0 0 35

GIH 0 25 0 0 0 4 0 0 0 0 0 29

JPT 0 0 43 0 0 0 0 0 0 0 0 43

LWK 0 0 0 33 0 0 0 0 0 4 0 37

MEX 0 10 0 0 23 0 0 0 0 0 0 33

MKK 32 0 0 21 0 0 0 4 2 1 4 64

TSI 0 31 0 0 0 0 0 0 0 0 0 31

YRI 0 0 0 73 0 0 0 0 0 9 0 82

__all__ 33 127 120 140 23 4 7 4 2 18 4 482

RBF SVM

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 17 1 0 0 0 0 0 0 0 0 9 27

CEU 0 59 0 0 0 0 0 0 0 0 0 59

CHB 0 0 42 0 0 0 0 0 0 0 0 42

CHD 0 0 35 0 0 0 0 0 0 0 0 35

GIH 0 0 0 0 29 0 0 0 0 0 0 29

JPT 0 0 9 0 0 34 0 0 0 0 0 43

LWK 0 0 0 0 0 0 32 0 2 0 3 37

MEX 0 3 0 0 0 0 0 30 0 0 0 33

MKK 0 0 0 0 0 0 0 0 63 0 1 64

TSI 0 31 0 0 0 0 0 0 0 0 0 31

YRI 0 0 0 0 0 0 0 0 0 0 82 82

__all__ 17 94 86 0 29 34 32 30 65 0 95 482

Linear SVM

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 24 0 0 0 0 0 0 0 1 2 27

CEU 0 57 0 0 0 0 0 0 2 0 59

CHB 0 0 26 15 0 1 0 0 0 0 42

CHD 0 0 21 14 0 0 0 0 0 0 35

GIH 0 0 0 0 29 0 0 0 0 0 29

JPT 0 0 5 0 0 38 0 0 0 0 43

LWK 0 0 0 0 0 0 35 0 2 0 0 37

MEX 0 0 0 0 0 0 0 32 0 1 0 33

MKK 0 0 0 0 0 0 1 0 63 0 0 64

TSI 0 7 0 0 0 0 0 0 0 24 0 31

YRI 0 0 0 0 0 0 0 0 0 0 82 82

__all__ 24 64 52 29 29 39 36 32 65 28 84 482

Ward

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 0 19 8 0 0 0 0 0 0 0 27

CEU 0 0 0 0 54 0 0 0 0 0 5 59

CHB 0 0 0 0 0 0 42 0 0 0 0 42

CHD 0 0 0 0 0 0 0 35 0 0 0 0 35

GIH 0 0 0 0 0 29 0 0 0 0 0 29

JPT 0 0 0 0 0 0 0 43 0 0 0 0 43

LWK 0 0 0 37 0 0 0 0 0 0 0 37

MEX 0 25 0 0 6 1 1 0 0 0 0 33

MKK 52 0 0 1 0 0 0 4 0 7 0 64

TSI 0 0 0 0 31 0 0 0 0 0 0 31

YRI 0 0 0 79 0 0 0 0 3 0 0 82

__all__ 52 25 19 125 91 30 121 4 3 7 5

482

Nearest Neighbors

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 2 7 0 0 0 0 0 0 0 18 27

CEU 0 59 0 0 0 0 0 0 0 0 59

CHB 0 0 30 4 0 8 0 0 0 0 42

CHD 0 0 20 8 0 7 0 0 0 0 35

GIH 0 18 0 0 11 0 0 0 0 0 29

JPT 0 0 3 1 0 39 0 0 0 0 43

LWK 0 0 0 0 0 0 9 0 0 28 37

MEX 0 18 0 0 0 0 0 15 0 0 33

MKK 0 4 0 0 0 0 3 0 42 0 15 64

TSI 0 28 0 0 0 0 0 0 0 3 0 31

YRI 0 0 0 0 0 0 0 0 0 0 82 82

__all__ 2 134 53 13 11 54 12 15 42 3 143 482

Naive Bayes

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 13 0 0 0 2 0 2 0 10 0 0 27

CEU 1 38 0 0 8 0 0 3 2 7 0 59

CHB 0 0 17 18 4 3 0 0 0 0 0 42

CHD 0 0 17 6 10 2 0 0 0 0 0 35

GIH 2 0 0 0 24 0 0 3 0 0 0 29

JPT 0 0 14 6 4 17 0 2 0 0 0 43

LWK 14 0 0 0 0 0 6 0 17 0 0 37

MEX 4 1 0 0 7 0 0 18 2 1 0 33

MKK 6 0 0 0 1 0 1 0 56 0 0 64

TSI 3 4 0 0 7 0 0 2 4 11 0 31

YRI 21 0 0 0 0 0 10 0 7 0 44 82

__all__ 64 43 48 30 67 22 19 28 98 19 44 482

QDA

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__ Actual

ASW 4 1 0 3 6 5 3 0 1 1 3 27

CEU 8 8 1 5 3 2 8 7 4 10 3 59

CHB 1 3 5 8

1 6 3 6 3 1 5 42

CHD 0 3 1 3 7 4 1 4 5 5 2 35

GIH 3 3 2 2 2 2 1 10 1 1 2 29

JPT 2 4 6 2 3 11 3 3 2 5 2 43

LWK 6 1 0 6 4 4 6 0 3 5 2 37

MEX 0 4 1 3 4 7 4 6 1 2 1 33

MKK 9 5 0 2 7 11 9 6 4 4 7 64

TSI 1 2 1 0 8 4 1 5 5 3 1 31

YRI 11 2 2 1 9 12 15 3 16 10 1 82

__all__ 45 36 19 35 54 68 54 50 45 47 29 482

MiniBatchKMeans

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 2 0 2 0 3 1 0 0 19 0 27

CEU 0 0 0 43 0 0 16 0 0 0 0 59

CHB 13 0 0 0 0 0 0 29 0 0 0 42

CHD 7 0 0 0 0 0 0 28 0 0 0 35

GIH 0 0 0 0 0 0 0 0 0 0 29 29

JPT 1 0 0 0 0 0 0 42 0 0 0 43

LWK 0 2 0 0 0 31 0 0 0 4 0 37

MEX 0 0 26 0 0 0 6 1 0 0 0 33

MKK 0 62 0 0 0 1 0 0 0 1 0 64

TSI 0 0 0 27 0 0 4 0 0 0 0 31

YRI 0 0 0 0 0 66 0 0 0 16 0 82

__all__ 21 66 26 72 0 101 27 100 0 40 29 482

Random Forest

Random Forest

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 1 4 0 0 1 1 0 0 7 0 13 27

CEU 0 51 1 0 1 0 0 0 4 1 1 59

CHB 0 0 32 7 0 2 0 0 1 0 0 42

CHD 0 1 23 8 0 2 0 0 1 0 0 35

GIH 0 9 4 1 6 0 0 1 4 4 0 29

JPT 0 0 27 7 0 5 1 0 1 1 1 43

LWK 0 0 0 0 0 0 2 0 16 0 19 37

MEX 0 14 1 1 3 10 6 3 3 1 33

MKK 0 2 0 0 0 0 2 0 34 0 26 64

TSI 0 25 0 0 0 1 0 0 1 3 1 31

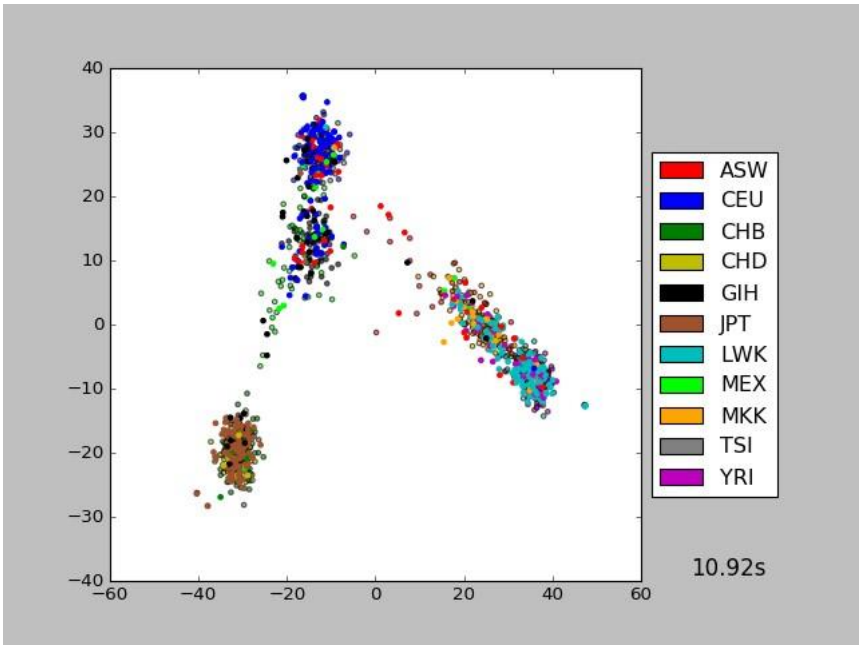
YRI 0 1 0 0 0 0 2 0 12 0 67 82

__all__ 1 107 88 24 11 12 7 7 84 12 129 482

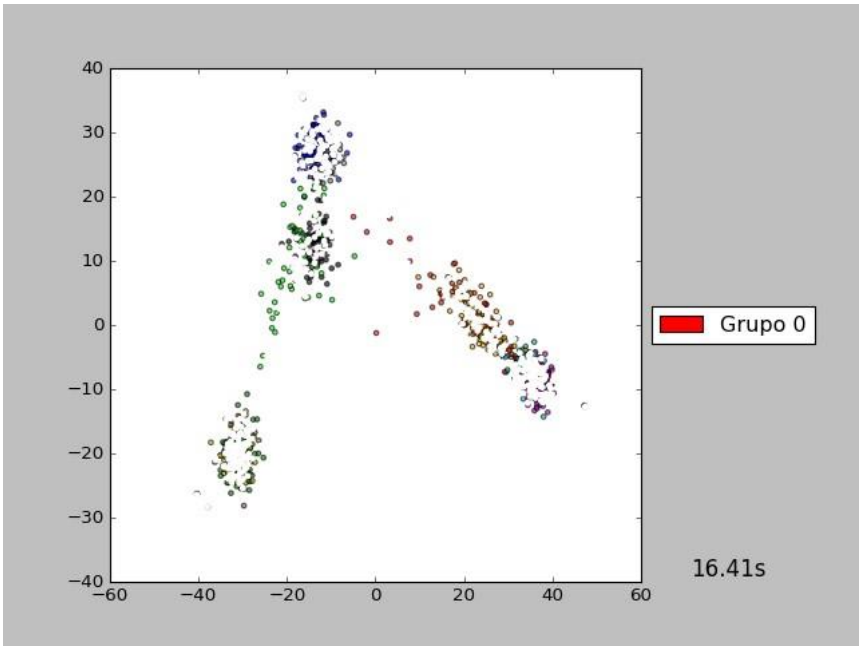
Predicciones del conjunto de test

Se muestran en color claro los puntos del conjunto de entrenamiento, frente a los predichos por el algoritmo

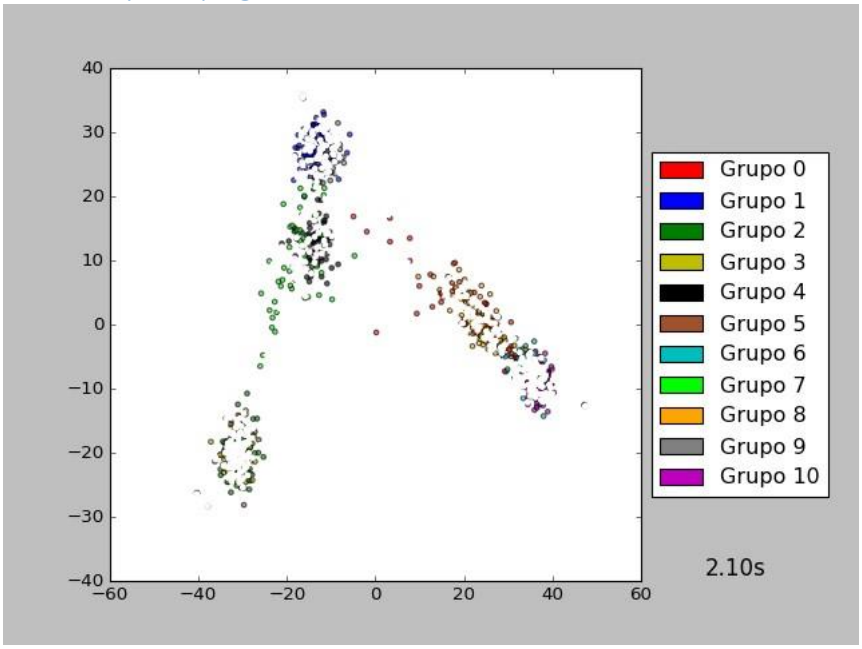
AdaBoost



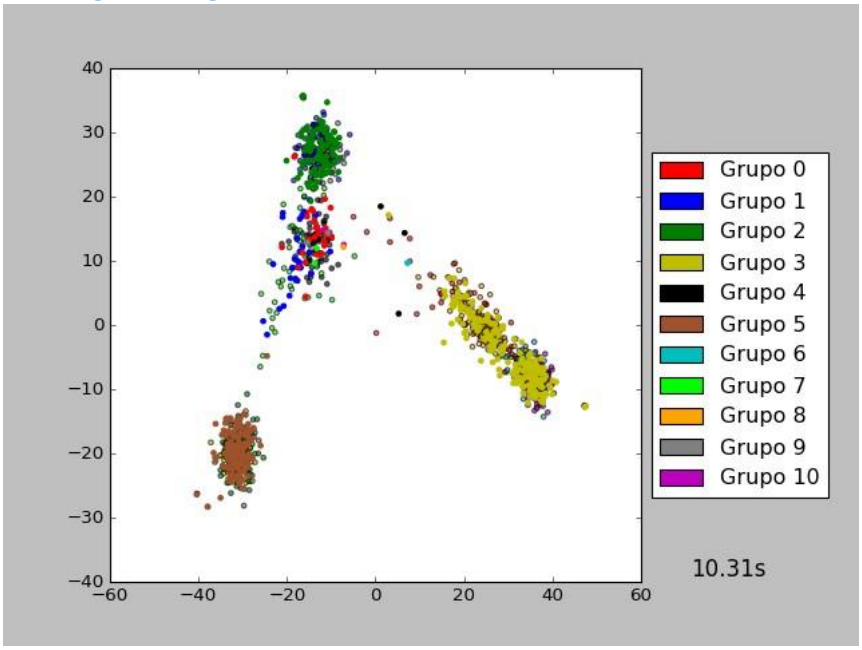
DBScan



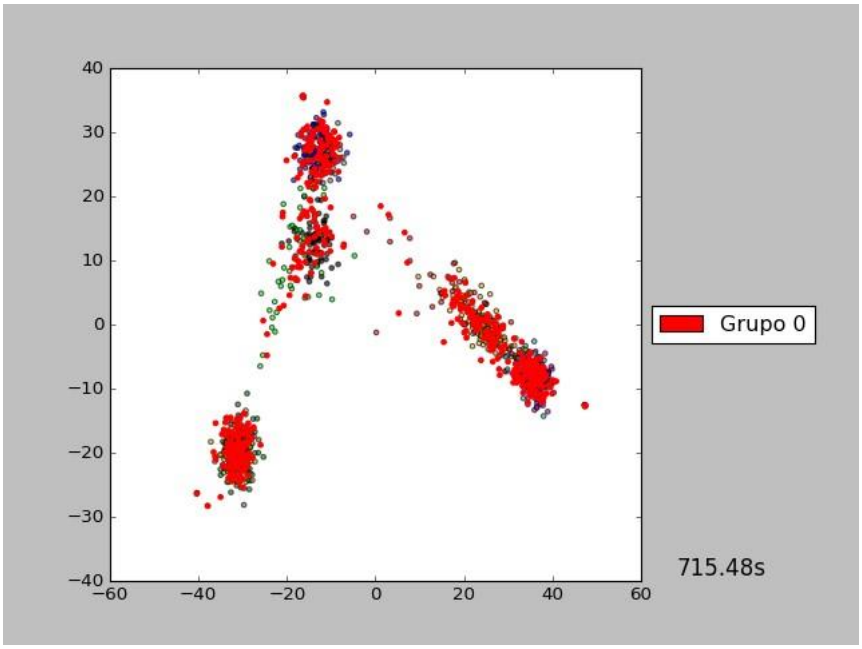
A nity Propagation



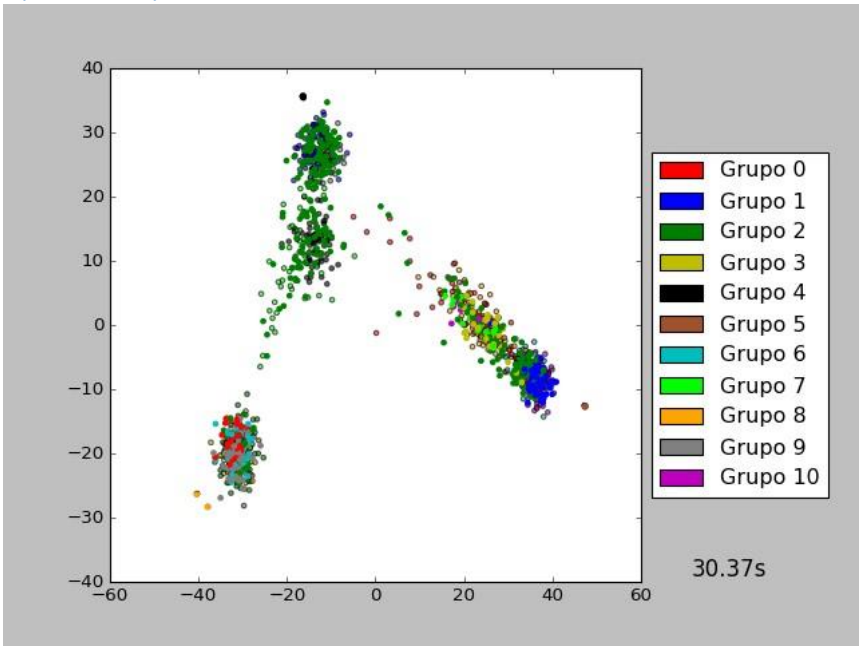
Average linkage



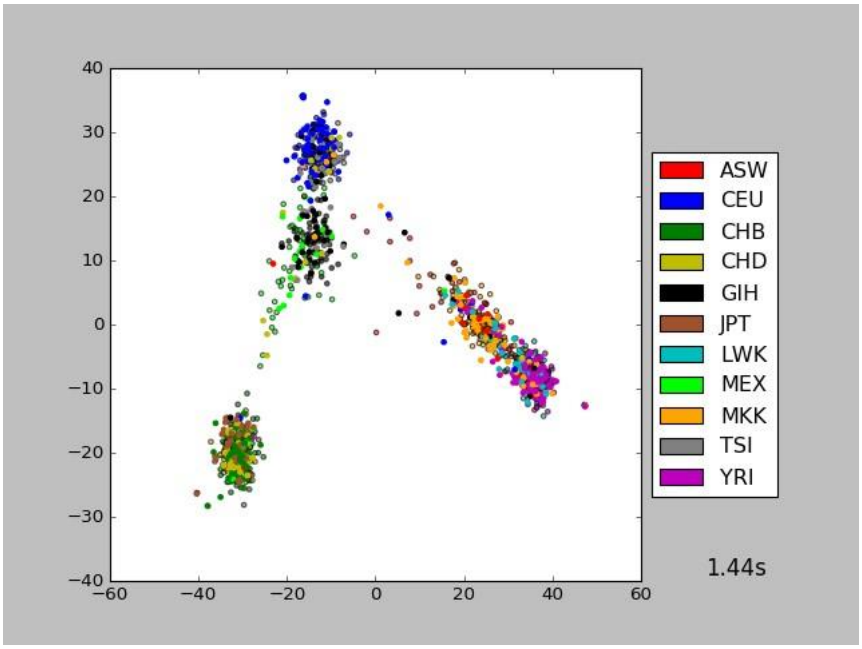
MeanShift



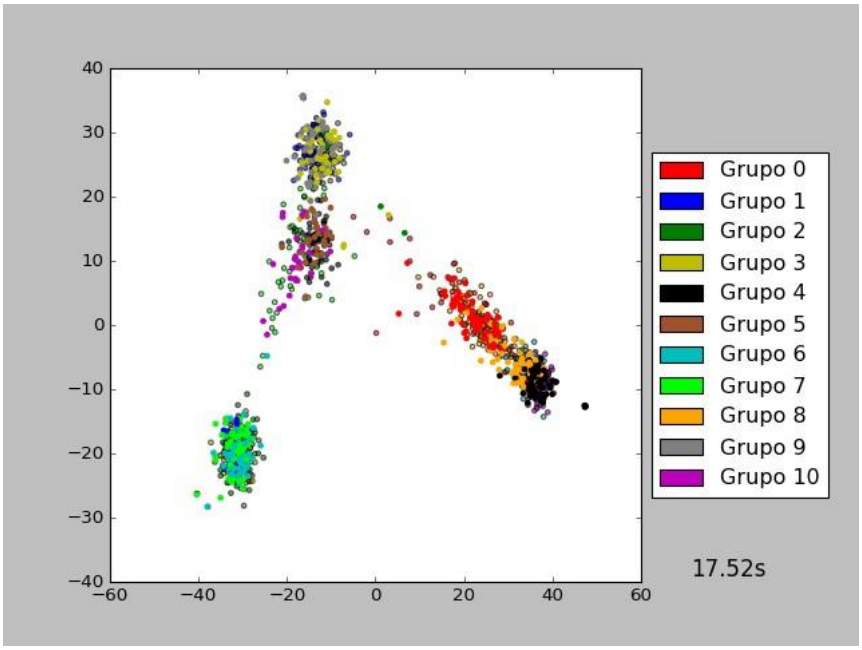
Spectral Spectral



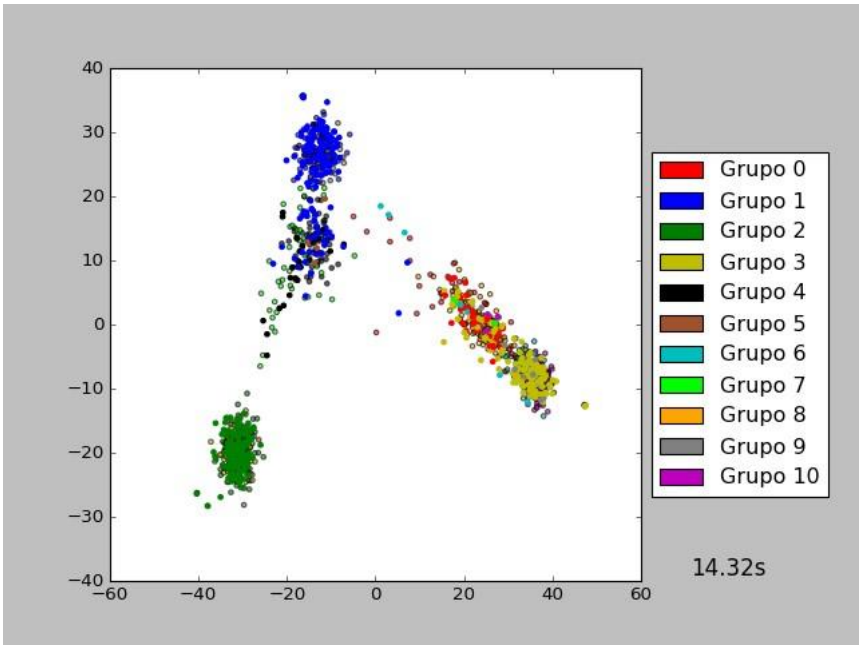
Decision Tree



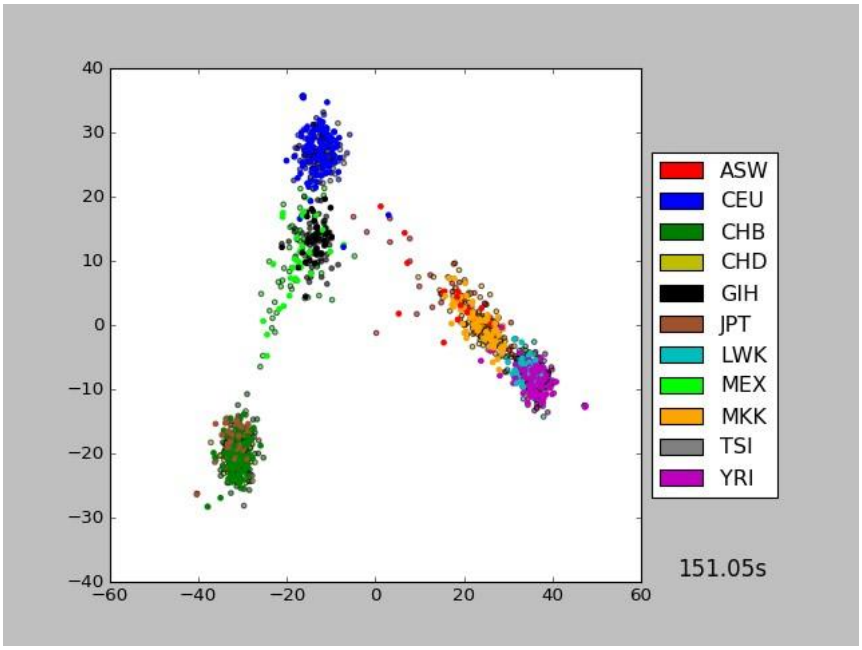
KMeans



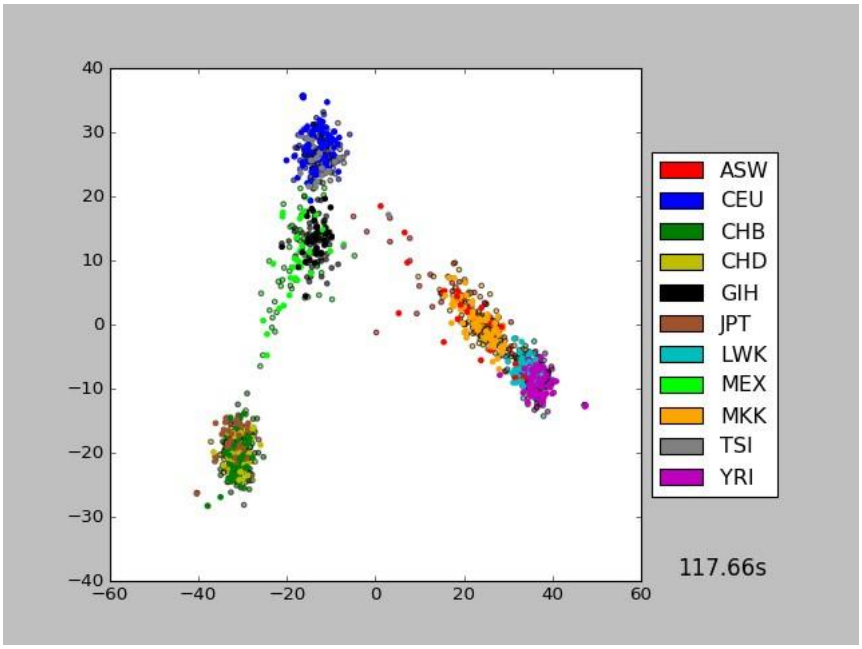
Birch



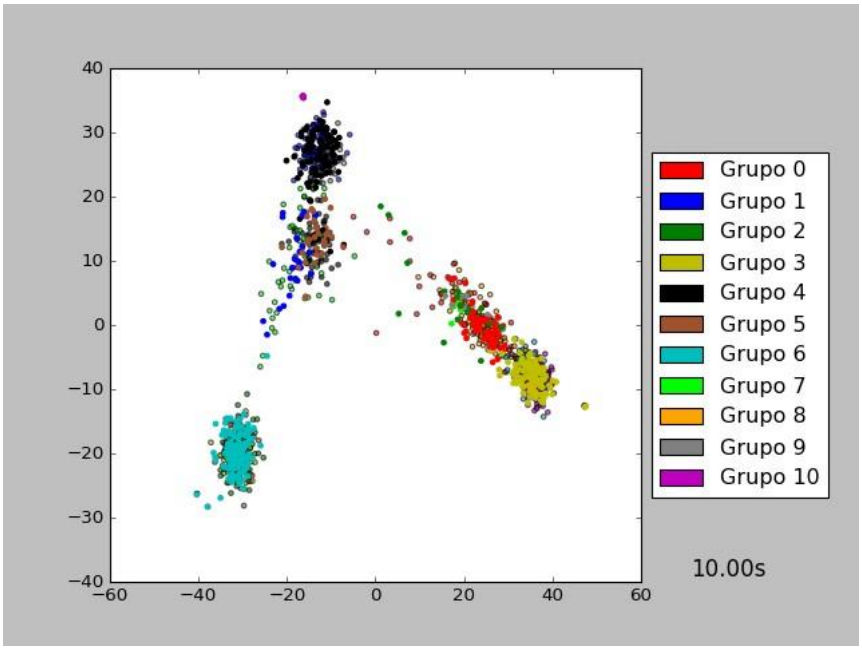
RBF SVM



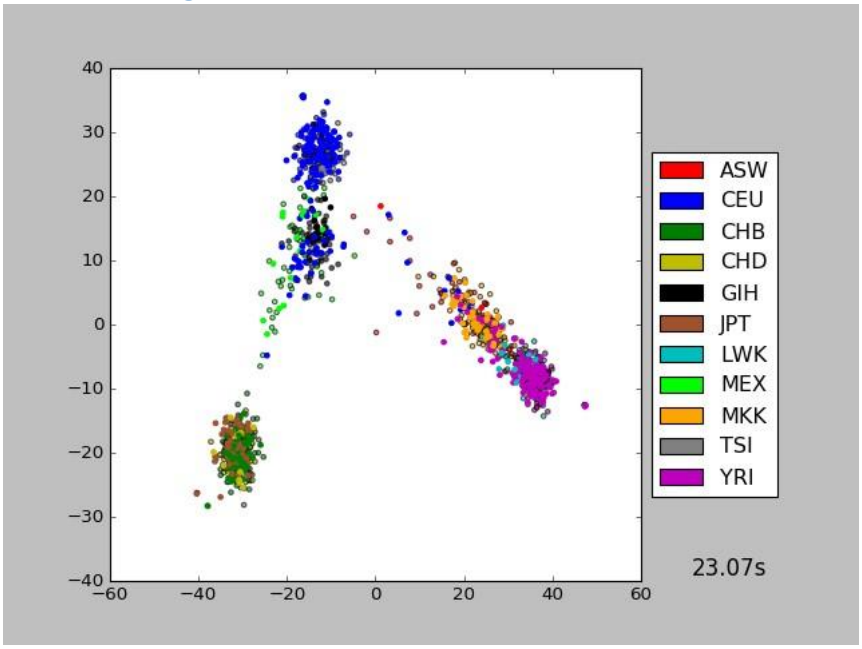
Linear SVM Linear SVM



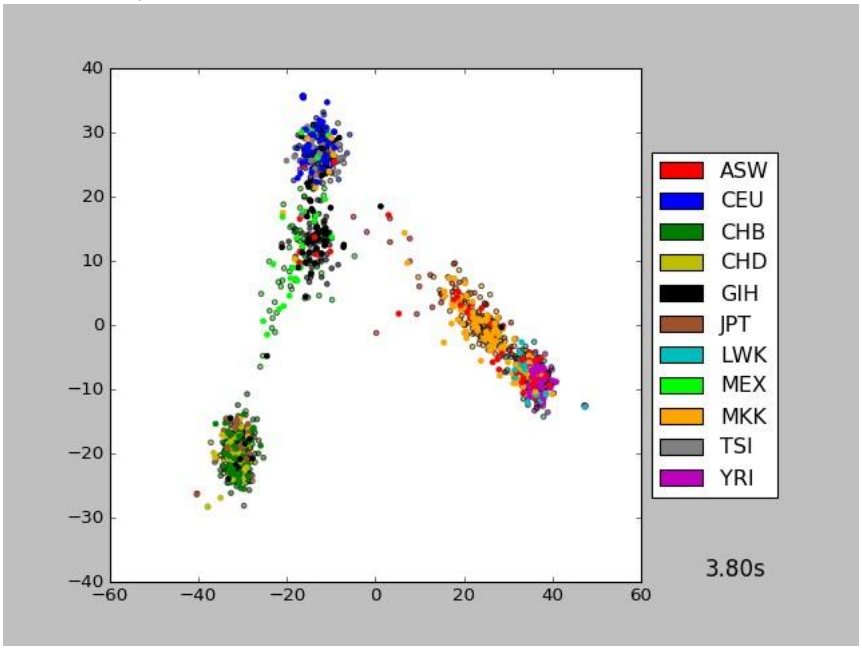
Ward



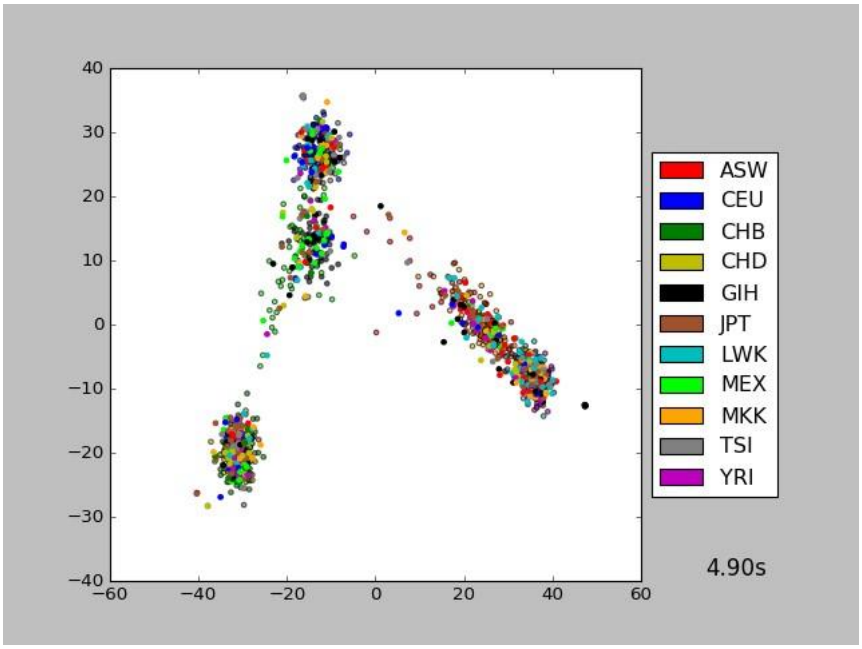
Nearest Neighbors



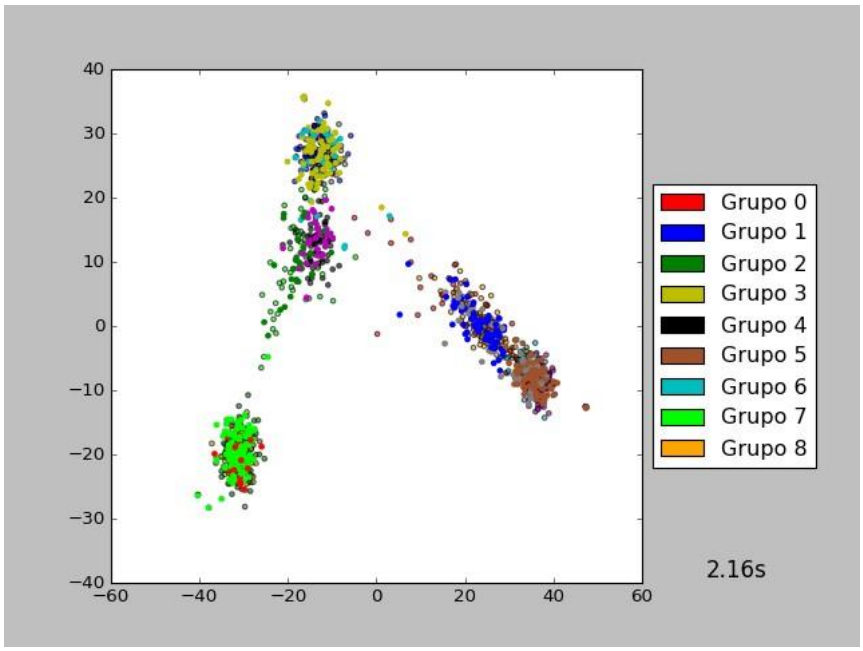
Naive Bayes



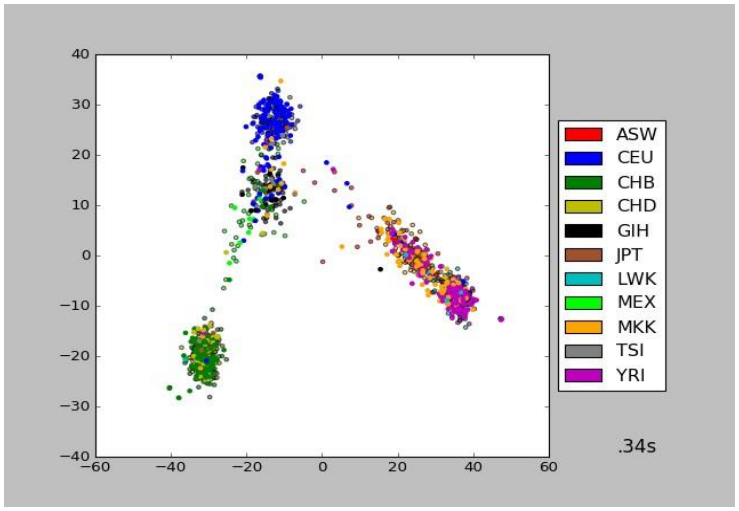
QDA



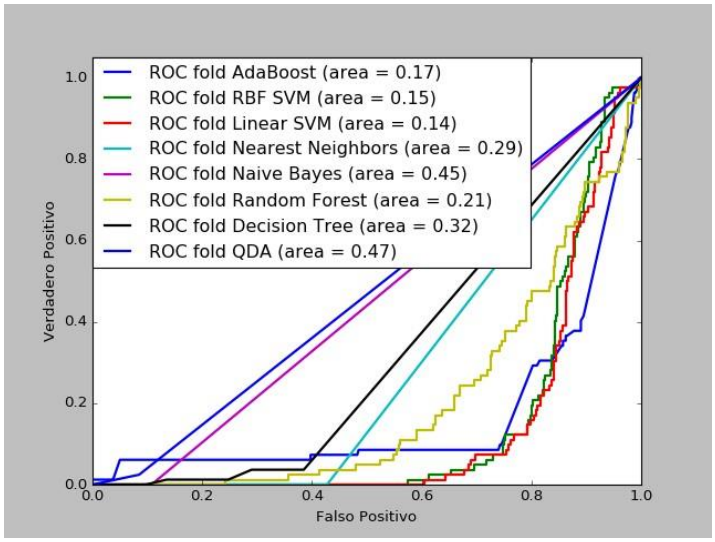
MiniBatchKMeans MiniBatchKMeans



Random Forest

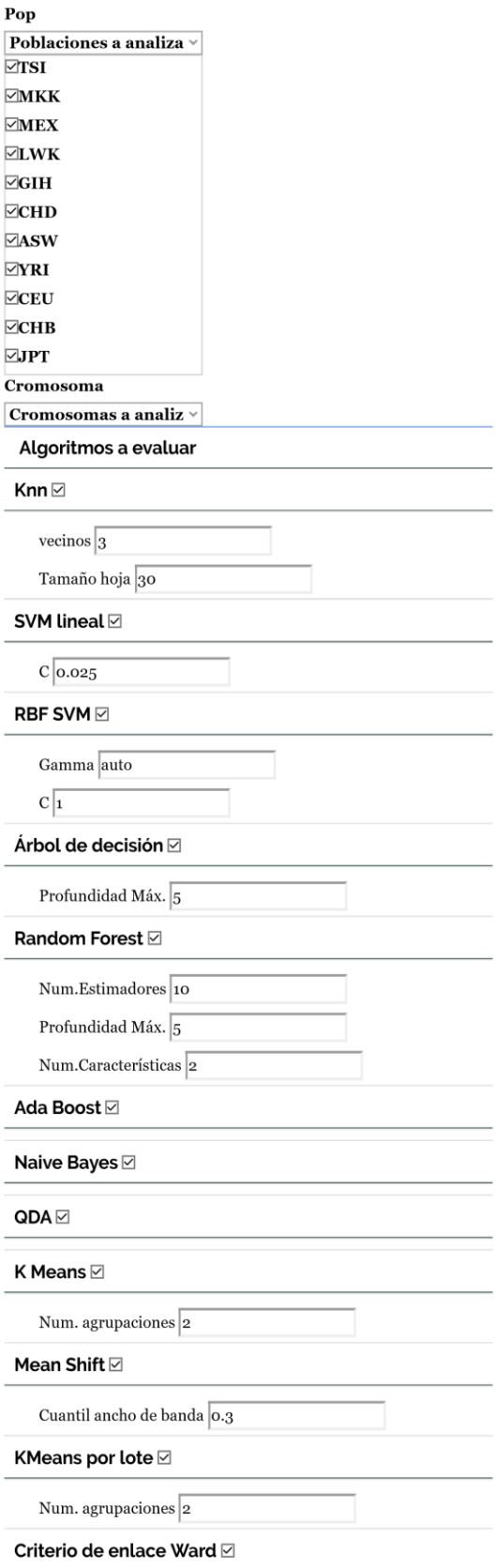


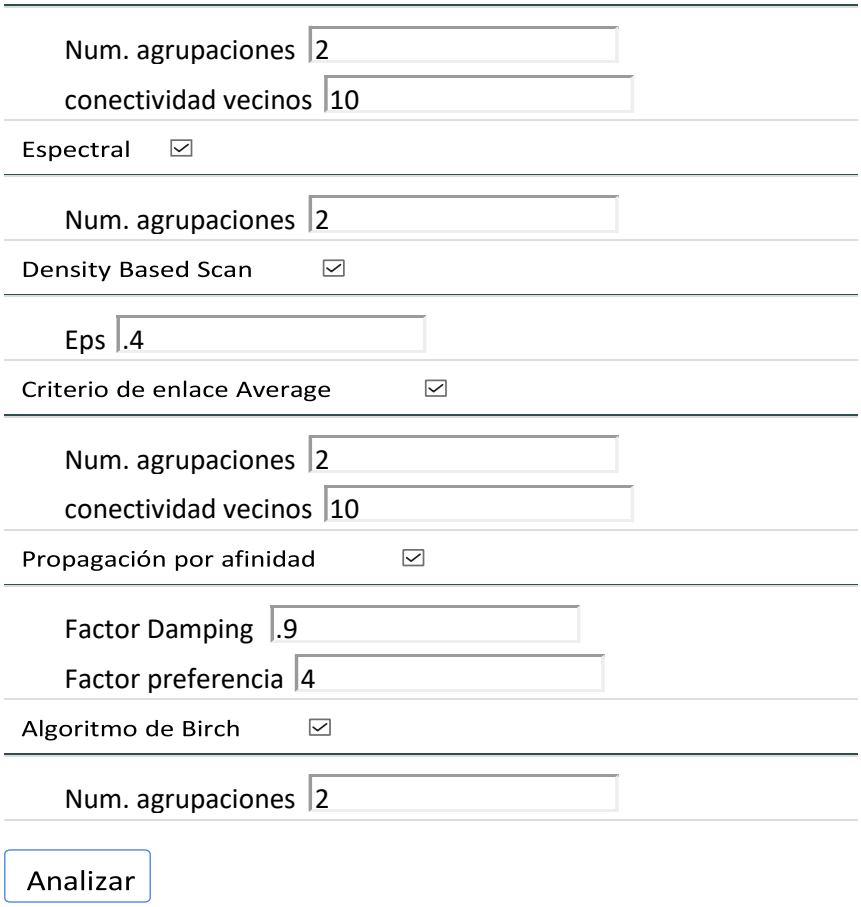
ROC (sólo para clasificadores)



C Ejecución de Interfaz Web

AlgML





Resultados de Pyranhap-AlgML

KMeans

Metrica	Valor
v_measure_score	0.42834748292980157
mutual_info_score	0.6467506059721713
adjusted_rand_score	0.18120554116604204
adjusted_mutual_info_score	0.27244672168372946

Nearest Neighbors

Metrica	Valor
v_measure_score	0.625755534340069
mutual_info_score	1.24282921106171
adjusted_rand_score	0.35281762812457557
adjusted_mutual_info_score	0.5121039431708663

Linear SVM

Metrica	Valor
v_measure_score	0.847904773011573
mutual_info_score	1.9777401284000475
adjusted_rand_score	0.7918195695012726

Metrica	Valor
adjusted_mutual_info_score	0.8351063596633732

Random Forest

Metrica	Valor
v_measure_score	0.45484357119668295
mutual_info_score	0.9491087638061634
adjusted_rand_score	0.2696109291516511
adjusted_mutual_info_score	0.376106322545067

RBF SVM

Metrica	Valor
v_measure_score	0.843335186267072
mutual_info_score	1.8459873203967656
adjusted_rand_score	0.6996306101419062
adjusted_mutual_info_score	0.7783533388350677

Average linkage

Metrica	Valor
v_measure_score	0.43773088673611094
mutual_info_score	0.6614389836083252
adjusted_rand_score	0.18490182331575122
adjusted_mutual_info_score	0.2787370340814149

Naive Bayes

Metrica	Valor
v_measure_score	0.5161018304245616
mutual_info_score	1.187871520654344
adjusted_rand_score	0.3382125813194698
adjusted_mutual_info_score	0.48101667841923545

Decision Tree

Metrica	Valor
v_measure_score	0.5333232875958471
mutual_info_score	1.2379036041430582
adjusted_rand_score	0.40501035057722024
adjusted_mutual_info_score	0.5035957873297964

Spectral

Metrica	Valor
v_measure_score	0.013756722012146748
mutual_info_score	0.01665675689786122
adjusted_rand_score	-0.00054230955313031
adjusted_mutual_info_score	0.0022859031599917834

QDA

Metrica	Valor
v_measure_score	0.05075440755105539
mutual_info_score	0.11900022740500585
adjusted_rand_score	0.0005131752561600315
adjusted_mutual_info_score	0.0024492597091868916

AdaBoost

Metrica	Valor
v_measure_score	0.46810788328772657
mutual_info_score	1.0196478278930634
adjusted_rand_score	0.2820666601409987
adjusted_mutual_info_score	0.40624785107467776

Affinity Propagation

Metrica	Valor
v_measure_score	0.5438591031193712
mutual_info_score	1.6515799691865263
adjusted_rand_score	0.2461082991380762
adjusted_mutual_info_score	0.26087859661177687

DBScan

Metrica	Valor
v_measure_score	-2.3664672917235377e-16
mutual_info_score	-2.7755575615628914e-16
adjusted_rand_score	0.0
adjusted_mutual_info_score	-2.1416038021780236-16

MeanShift

Metrica	Valor
v_measure_score	-2.3664672917235377e-16
mutual_info_score	-2.7755575615628914e-16
adjusted_rand_score	0.0

Metrica	Valor
adjusted_mutual_info_score	-2.141603802178023e-16

Birch

Metrica	Valor
v_measure_score	0.44209030532356514
mutual_info_score	0.6681937268761412
adjusted_rand_score	0.1862513998893652
adjusted_mutual_info_score	0.28162970779084545

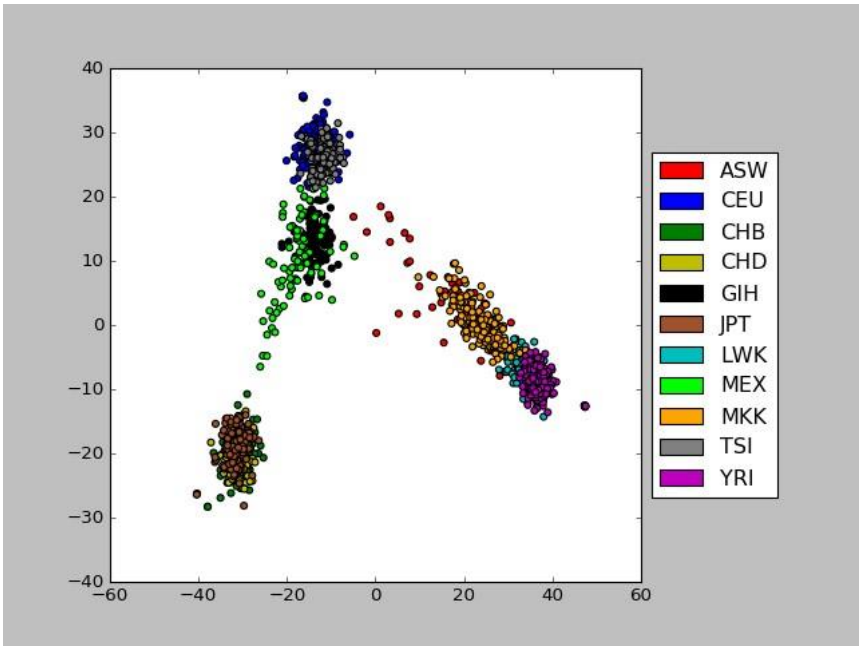
Ward

Metrica	Valor
v_measure_score	0.43413148548959224
mutual_info_score	0.6558318217251737
adjusted_rand_score	0.18361106098286675
adjusted_mutual_info_score	0.2763357820326716

MiniBatchKMeans

Metrica	Valor
v_measure_score	0.4310427157121222
mutual_info_score	0.6509948054302326
adjusted_rand_score	0.18237900566242982
adjusted_mutual_info_score	0.2742643235360763

Representación del conjunto completo en PCA dos dimensiones



Matriz de confusión

KMeans

KMeans

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 23 5 0 0 0 0 0 0 0 0 0 28

CEU 0 61 0 0 0 0 0 0 0 0 0 61

CHB 0 49 0 0 0 0 0 0 0 0 0 49

CHD 0 32 0 0 0 0 0 0 0 0 0 32

GIH 0 29 0 0 0 0 0 0 0 0 0 29

JPT 0 46 0 0 0 0 0 0 0 0 0 46

LWK 39 0 0 0 0 0 0 0 0 0 0 39

MEX 0 28 0 0 0 0 0 0 0 0 0 28

MKK 70 0 0 0 0 0 0 0 0 0 0 70

TSI 0 38 0 0 0 0 0 0 0 0 0 38

YRI 62 0 0 0 0 0 0 0 0 0 0 62

__all__ 194 288 0 0 0 0 0 0 0 0 0 482

Nearest Neighbors

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 2 4 0 0 0 0 0 0 0 0 22 28

CEU 0 61 0 0 0 0 0 0 0 0 0 61

CHB 0 0 32 7 0 10 0 0 0 0 0 49

CHD 0 0 18 2 0 12 0 0 0 0 0 32

GIH 0 28 0 0 1 0 0 0 0 0 0 29

JPT 0 0 5 1 0 40 0 0 0 0 0 46

LWK 0 0 0 0 0 0 1 0 0 0 38 39

MEX 0 16 0 0 0 0 0 12 0 0 0 28

MKK 0 0 0 0 0 0 0 0 28 0 42 70

TSI 0 38 0 0 0 0 0 0 0 0 0 38

YRI 0 0 0 0 0 0 0 0 0 0 62 62

__all__ 2 147 55 10 1 62 1 12 28 0 164 482

Linear SVM

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 23 1 0 0 0 0 1 0 0 0 3 28

CEU 0 60 0 0 0 0 0 0 0 1 0 61

CHB 0 0 28 20 0 1 0 0 0 0 0 49

CHD 0 0 15 16 0 1 0 0 0 0 0 32

GIH 0 1 0 0 28 0 0 0 0 0 0 29

JPT 0 0 4 2 0 40 0 0 0 0 0 46

LWK 0 0 0 0 0 0 38 0 0 0 1 39

MEX 0 2 1 0 0 0 0 24 0 1 0 28

MKK 0 0 0 0 0 0 0 2 0 68 0 0 70

TSI 0 11 0 0 0 0 0 0 0 0 27 0 38

YRI 0 0 0 0 0 0 0 0 0 0 0 62 62

__all__ 23 75 48 38 28 42 41 24 68 29 66 482

Anity Propagation

Predicted 12 CEU GIH JPT MEX ASW CHB CHD LWK MKK TSI YRI __all__

Actual

ASW 12 0 0 0 0 0 0 0 0 0 0 0 0

CEU 58 2 0 0 1 0 0 0 0 0 0 0 61

GIH 26 0 0 0 3 0 0 0 0 0 0 0 29

JPT 0 0 0 46 0 0 0 0 0 0 0 0 46

MEX 26 0 1 1 0 0 0 0 0 0 0 0 28

ASW 28 0 0 0 0 0 0 0 0 0 0 0 28

CHB 0 0 0 49 0 0 0 0 0 0 0 0 49

CHD 1 0 0 31 0 0 0 0 0 0 0 0 32

LWK 39 0 0 0 0 0 0 0 0 0 0 0 39

MKK 70 0 0 0 0 0 0 0 0 0 0 0 70

TSI 35 0 0 0 3 0 0 0 0 0 0 0 38

YRI 62 0 0 0 0 0 0 0 0 0 0 0 62

__all__ 345 2 1 127 7 0 0 0 0 0 0 0 482

Random Forest

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 1 5 0 0 1 0 1 0 2 0 18 28

CEU 1 51 0 0 0 0 0 3 2 3 1 61

CHB 0 3 24 16 0 3 0 0 1 0 2 49

CHD 0 2 16 9 0 4 0 1 0 0 0 32

GIH 0 16 0 0 9 0 0 0 2 1 1 29

JPT 0 0 15 16 0 12 0 0 1 0 2 46

LWK 0 1 0 0 0 0 1 0 7 0 30 39

MEX 0 16 2 0 2 1 1 5 1 0 0 28

MKK 0 1 0 0 1 0 0 0 28 0 40 70

TSI 0 33 0 1 0 0 0 0 3 1 0 38

YRI 0 0 0 0 0 0 0 1 0 2 0 59 62

__all__ 2 128 57 42 13 20 4 9 49 5 153 482

RBF SVM

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 18 0 0 0 0 0 0 0 0 0 10 28

CEU 0 61 0 0 0 0 0 0 0 0 0 61

CHB 0 0 48 0 0 1 0 0 0 0 0 49

CHD 0 0 32 0 0 0 0 0 0 0 0 32

GIH 0 1 0 0 28 0 0 0 0 0 0 29

JPT 0 0 12 0 0 34 0 0 0 0 0 46

LWK 0 0 0 0 0 0 34 0 0 0 5 39

MEX 0 4 1 0 0 0 0 23 0 0 0 28

MKK 0 0 0 0 0 0 0 0 69 0 1 70

TSI 0 38 0 0 0 0 0 0 0 0 0 38

YRI 0 0 0 0 0 0 0 0 0 0 62 62

__all__ 18 104 93 0 28 35 34 23 69 0 78 482

Average linkage

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 2 26 0 0 0 0 0 0 0 0 0 28

CEU 61 0 0 0 0 0 0 0 0 0 0 61

CHB 49 0 0 0 0 0 0 0 0 0 0 49

CHD 32 0 0 0 0 0 0 0 0 0 0 32

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 46 0 0 0 0 0 0 0 0 0 0 46

LWK 0 39 0 0 0 0 0 0 0 0 0 39

MEX 28 0 0 0 0 0 0 0 0 0 0 28

MKK 0 70 0 0 0 0 0 0 0 0 0 70

TSI 38 0 0 0 0 0 0 0 0 0 0 38

YRI 0 62 0 0 0 0 0 0 0 0 0 62

__all__ 285 197 0 0 0 0 0 0 0 0 0 482

Naive Bayes

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 14 0 0 0 1 0 2 1 10 0 0 28

CEU 1 42 0 0 3 0 0 1 3 11 0 61

CHB 0 0 24 17 6 1 0 1 0 0 0 49

CHD 2 0 11 4 12 0 0 3 0 0 0 32

GIH 1 0 0 0 17 0 0 7 3 1 0 29

JPT 0 0 10 12 5 15 0 4 0 0 0 46

LWK 16 0 0 0 0 0 6 0 17 0 0 39

MEX 1 1 0 0 5 0 0 20 1 0 0 28

MKK 7 0 0 0 0 0 0 0 63 0 0 70

TSI 0 1 0 0 12 0 0 7 3 15 0 38

YRI 13 0 0 0 0 0 15 0 10 0 24 62

__all__ 55 44 45 33 61 16 23 44 110 27 24 482

Decision Tree

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 3 2 0 0 0 0 5 1 3 4 10 28

CEU 0 50 0 0 8 0 1 0 1 0 1 61

CHB 0 0 36 7 2 4 0 0 0 0 0 49

CHD 0 1 2 15 1 11 0 1 0 0 1 32

GIH 1 1 0 0 16 0 0 3 0 6 2 29

JPT 0 0 0 2 0 42 0 1 1 0 0 46

LWK 3 0 0 1 1 0 16 1 5 0 12 39

MEX 1 1 0 0 5 0 0 12 0 9 0 28

MKK 4 0 0 0 2 0 27 1 27 0 9 70

TSI 1 2 1 0 8 0 1 1 1 21 2 38

YRI 0 0 0 0 0 0 13 0 3 0 46 62

__all__ 13 57 39 25 43 57 63 21 41 40 83 482

Spectral

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 28 0 0 0 0 0 0 0 0 0 0 28

CEU 58 3 0 0 0 0 0 0 0 0 0 61

CHB 47 2 0 0 0 0 0 0 0 0 0 49

CHD 32 0 0 0 0 0 0 0 0 0 0 32

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 44 2 0 0 0 0 0 0 0 0 0 46

LWK 39 0 0 0 0 0 0 0 0 0 0 39

MEX 28 0 0 0 0 0 0 0 0 0 0 28

MKK 70 0 0 0 0 0 0 0 0 0 0 70

TSI 38 0 0 0 0 0 0 0 0 0 0 38

YRI 62 0 0 0 0 0 0 0 0 0 62

__all__ 475 7 0 0 0 0 0 0 0 0 0 482

QDA

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 1 5 3 1 2 2 4 4 2 2 2 28

CEU 2 11 2 4 6 11 3 8 4 9 1 61

CHB 1 2 4 5 6 8 3 5 5 5 5 49

CHD 2 3 6 3 4 4 0 6 2 1 1 32

GIH 2 3 1 1 4 6 2 4 3 3 0 29

JPT 2 8 3 4 3 11 0 6 5 2 2 46

LWK 7 1 5 2 2 7 2 3 4 5 1 39

MEX 2 3 3 3 3 5 0 2 4 2 1 28

MKK 9 6 8 1 5 9 8 6 7 8 3 70

TSI 5 2 3 1 2 8 1 7 3 3 3 38

YRI 11 7 2 0 7 8 9 5 5 4 4 62

__all__ 44 51 40 25 44 79 32 56 44 44 23 482

AdaBoost

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 1 4 0 0 4 0 1 3 9 0 6 28

CEU 0 46 0 0 7 0 0 4 0 4 0 61

CHB 0 0 44 2 3 0 0 0 0 0 0 49

CHD 0 0 28 3 0 1 0 0 0 0 0 32

GIH 0 19 0 0 8 0 0 2 0 0 0 29

JPT 0 1 34 5 2 3 0 1 0 0 0 46

LWK 2 0 0 0 1 0 5 0 4 0 27 39

MEX 0 11 2 0 7 0 0 8 0 0 0 28

MKK 10 1 0 0 8 0 10 0 24 0 17 70

TSI 0 26 0 0 6 0 0 4 0 2 0 38

YRI 4 0 0 0 0 0 6 0 9 0 43 62

__all__ 17 108 108 10 46 4 22 22 46 6 93 482

DBScan

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 0 0 0 0 0 0 0 0 0 0 28 28

CEU 0 0 0 0 0 0 0 0 0 0 61 61

CHB 0 0 0 0 0 0 0 0 0 0 49 49

CHD 0 0 0 0 0 0 0 0 0 0 32 32

GIH 0 0 0 0 0 0 0 0 0 0 29 29

JPT 0 0 0 0 0 0 0 0 0 0 46 46

LWK 0 0 0 0 0 0 0 0 0 0 39 39

MEX 0 0 0 0 0 0 0 0 0 0 28 28

MKK 0 0 0 0 0 0 0 0 0 0 70 70

TSI 0 0 0 0 0 0 0 0 0 0 38 38

YRI 0 0 0 0 0 0 0 0 0 0 62 62

__all__ 0 0 0 0 0 0 0 0 0 0 482 482

MeanShift

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__ Actual

ASW 28 0 0 0 0 0 0 0 0 0 0 28

CEU 61 0 0 0 0 0 0 0 0 0 0 61

CHB 49 0 0 0 0 0 0 0 0 0 0 49

CHD 32 0 0 0 0 0 0 0 0 0 0 32

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 46 0 0 0 0 0 0 0 0 0 0 46

LWK 39 0 0 0 0 0 0 0 0 0 0 39

MEX 28 0 0 0 0 0 0 0 0 0 0 28

MKK 70 0 0 0 0 0 0 0 0 0 0 70

TSI 38 0 0 0 0 0 0 0 0 0 0 38

YRI 62 0 0 0 0 0 0 0 0 0 0 62

__all__ 482 0 0 0 0 0 0 0 0 0 0 482

Birch

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__ Actual

ASW 1 27 0 0 0 0 0 0 0 0 0 28

CEU 61 0 0 0 0 0 0 0 0 0 0 61

CHB 49 0 0 0 0 0 0 0 0 0 0 49

CHD 32 0 0 0 0 0 0 0 0 0 0 32

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 46 0 0 0 0 0 0 0 0 0 0 46

LWK 0 39 0 0 0 0 0 0 0 0 0 39

MEX 28 0 0 0 0 0 0 0 0 0 0 28

MKK 0 70 0 0 0 0 0 0 0 0 0 70

TSI 38 0 0 0 0 0 0 0 0 0 0 38

YRI 0 62 0 0 0 0 0 0 0 0 0 62

__all__ 284 198 0 0 0 0 0 0 0 0 0 482

Ward

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__

Actual

ASW 3 25 0 0 0 0 0 0 0 0 0 28

CEU 61 0 0 0 0 0 0 0 0 0 0 61

CHB 49 0 0 0 0 0 0 0 0 0 0 49

CHD 32 0 0 0 0 0 0 0 0 0 0 32

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 46 0 0 0 0 0 0 0 0 0 0 46

LWK 0 39 0 0 0 0 0 0 0 0 0 39

MEX 28 0 0 0 0 0 0 0 0 0 0 28

MKK 0 70 0 0 0 0 0 0 0 0 0 70

TSI 38 0 0 0 0 0 0 0 0 0 0 38

YRI 0 62 0 0 0 0 0 0 0 0 0 62

__all__ 286 196 0 0 0 0 0 0 0 0 0 482

MiniBatchKMeans

Predicted ASW CEU CHB CHD GIH JPT LWK MEX MKK TSI YRI __all__ Actual

ASW 4 24 0 0 0 0 0 0 0 0 0 28

CEU 61 0 0 0 0 0 0 0 0 0 0 61

CHB 49 0 0 0 0 0 0 0 0 0 0 49

CHD 32 0 0 0 0 0 0 0 0 0 0 32

GIH 29 0 0 0 0 0 0 0 0 0 0 29

JPT 46 0 0 0 0 0 0 0 0 0 0 46

LWK 0 39 0 0 0 0 0 0 0 0 0 39

MEX 28 0 0 0 0 0 0 0 0 0 0 28

MKK 0 70 0 0 0 0 0 0 0 0 0 70

TSI 38 0 0 0 0 0 0 0 0 0 0 38

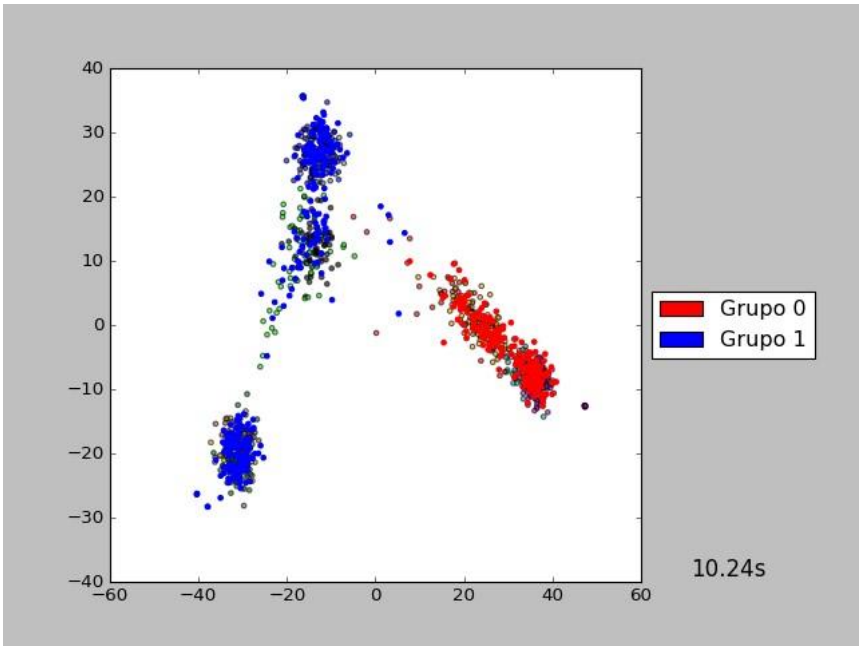
YRI 0 62 0 0 0 0 0 0 0 0 0 62

__all__ 287 195 0 0 0 0 0 0 0 0 0 482

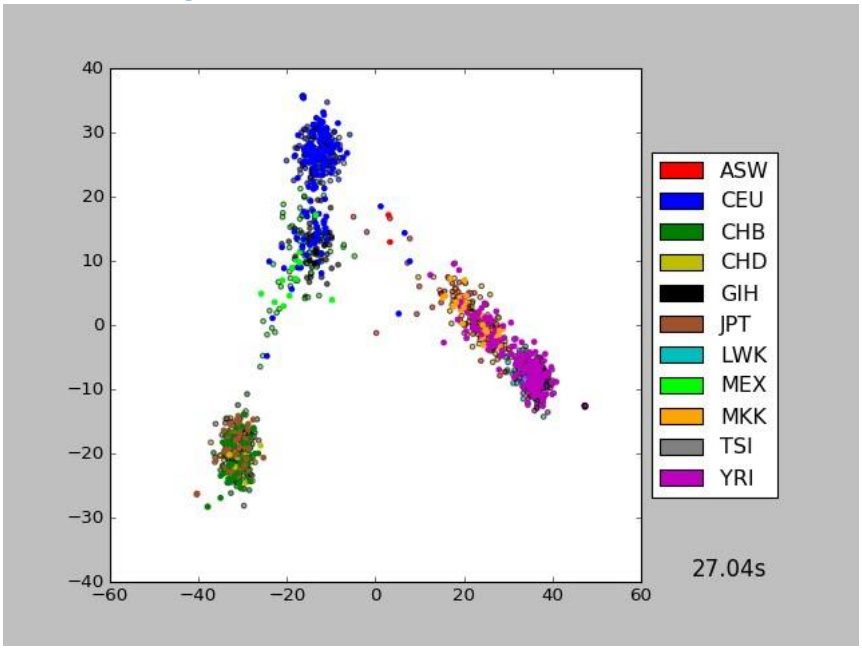
Predicciones del conjunto de test

Se muestran en color claro los puntos del conjunto de entrenamiento, frente a los predichos por el algoritmo

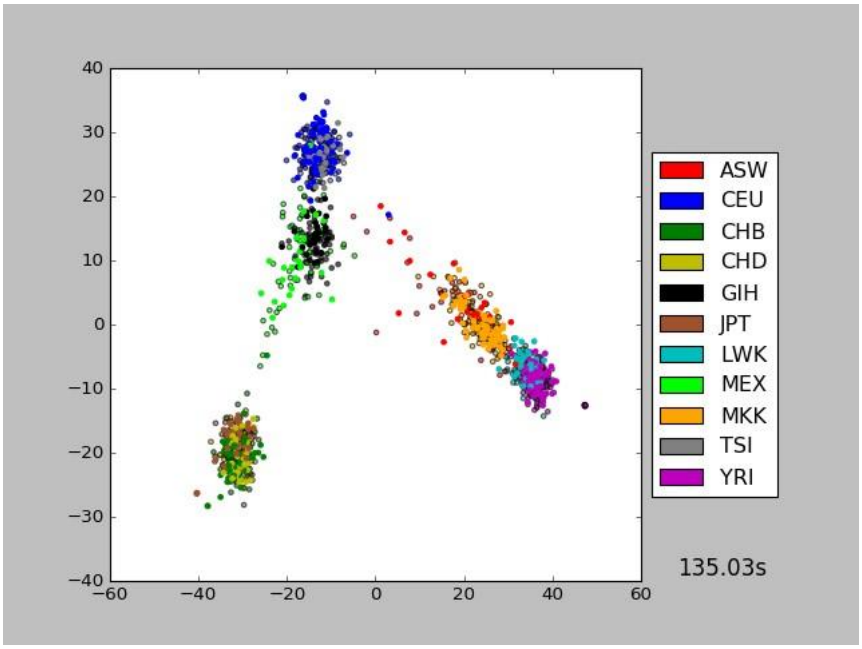
KMeans



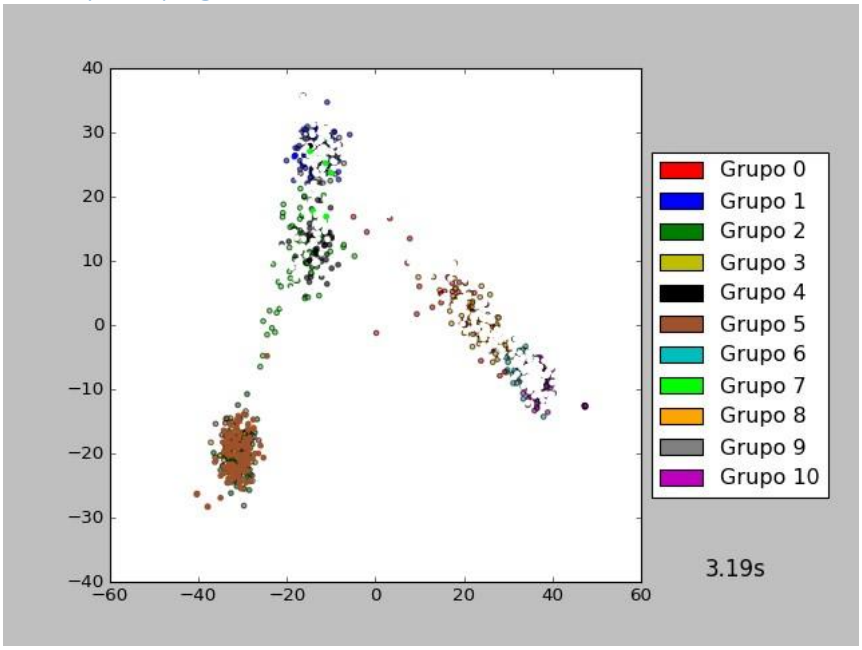
Nearest Neighbors



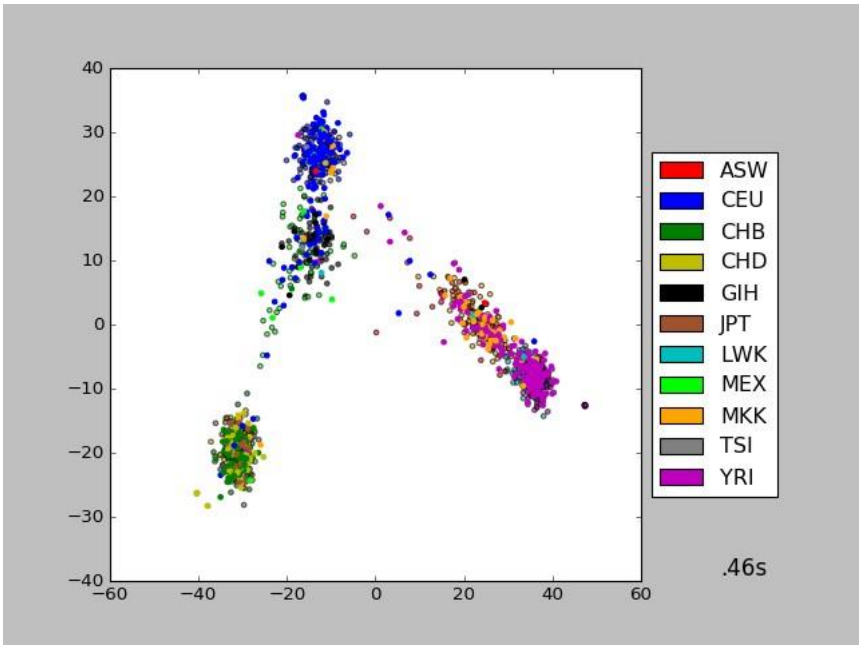
Linear SVM



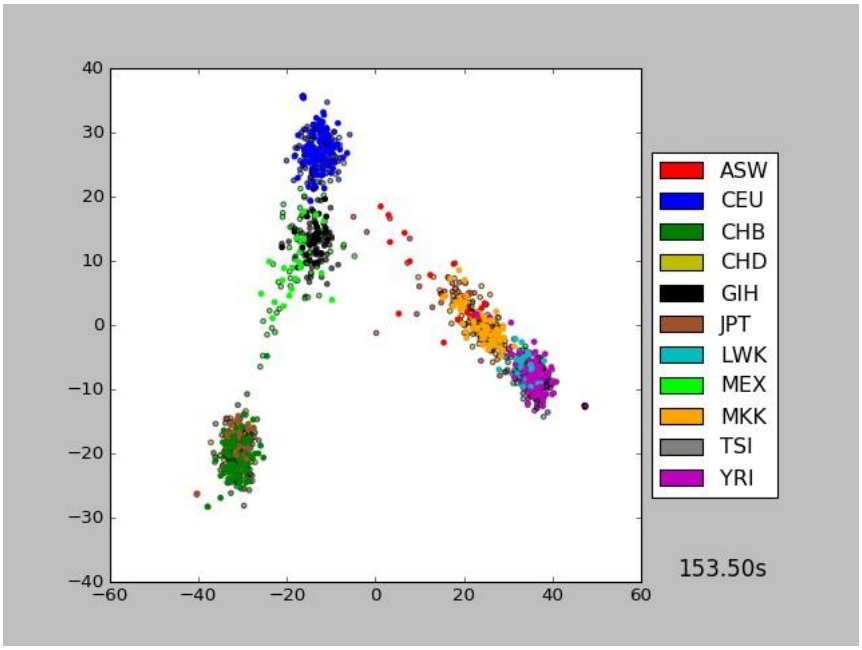
Afinity Propagation



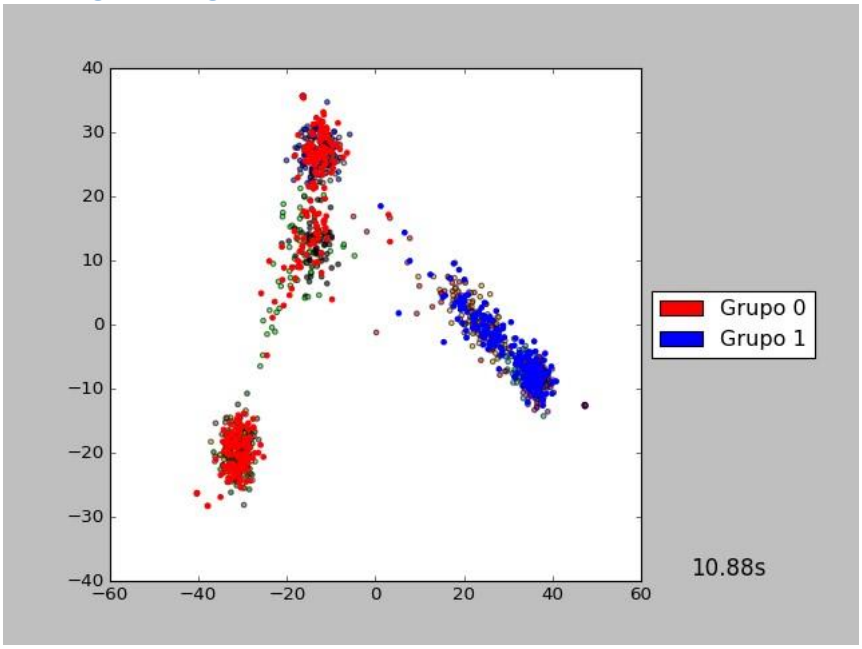
Random Forest



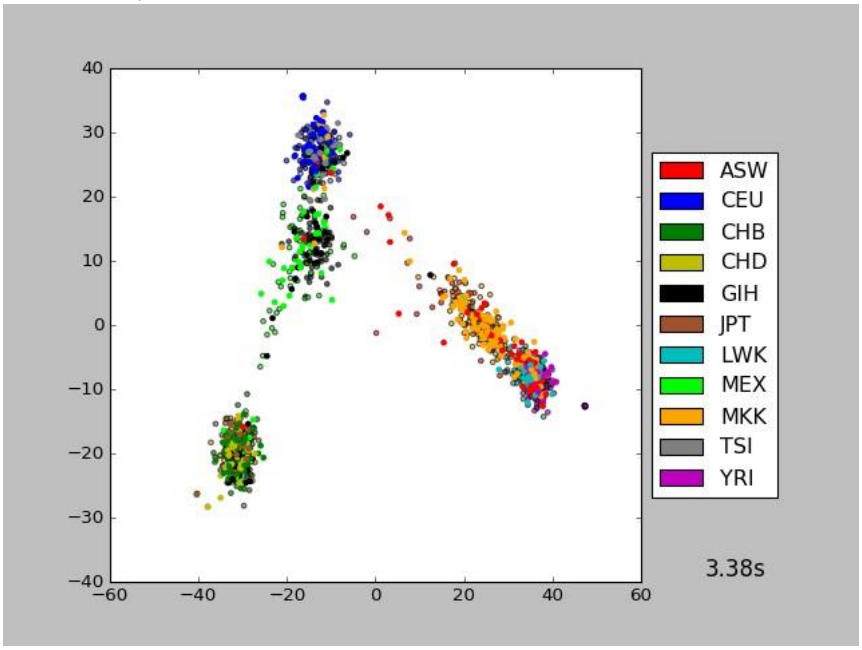
RBF SVM



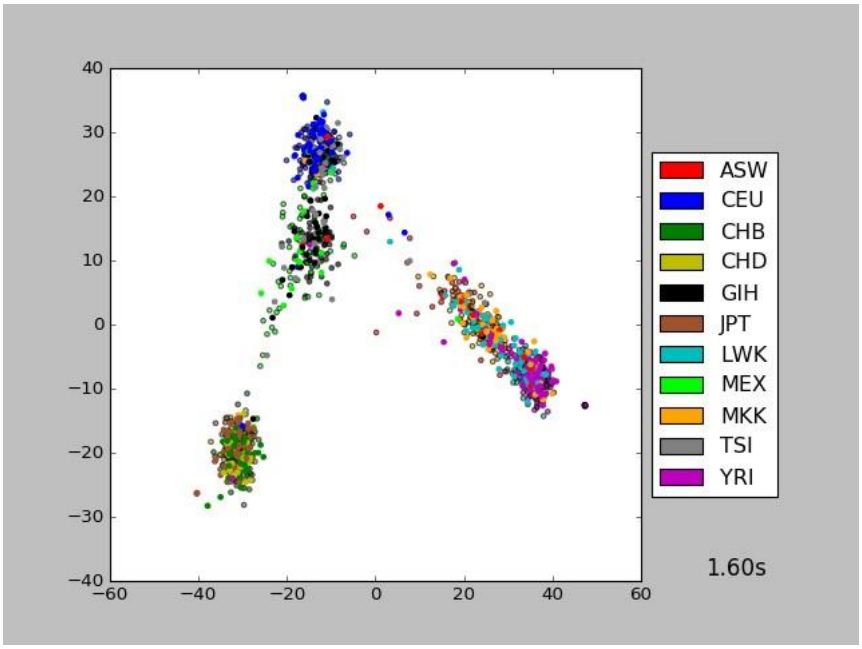
Average linkage



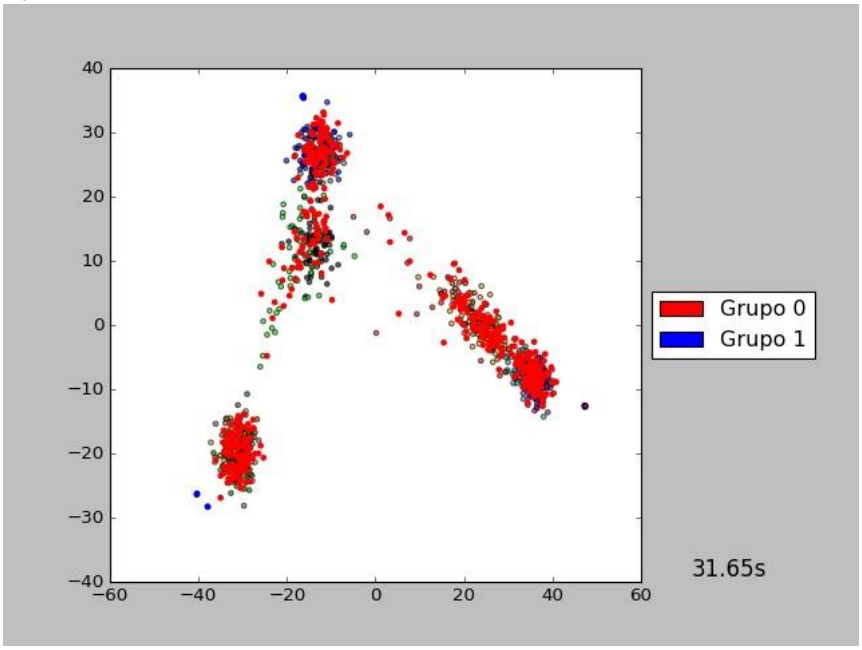
Naive Bayes



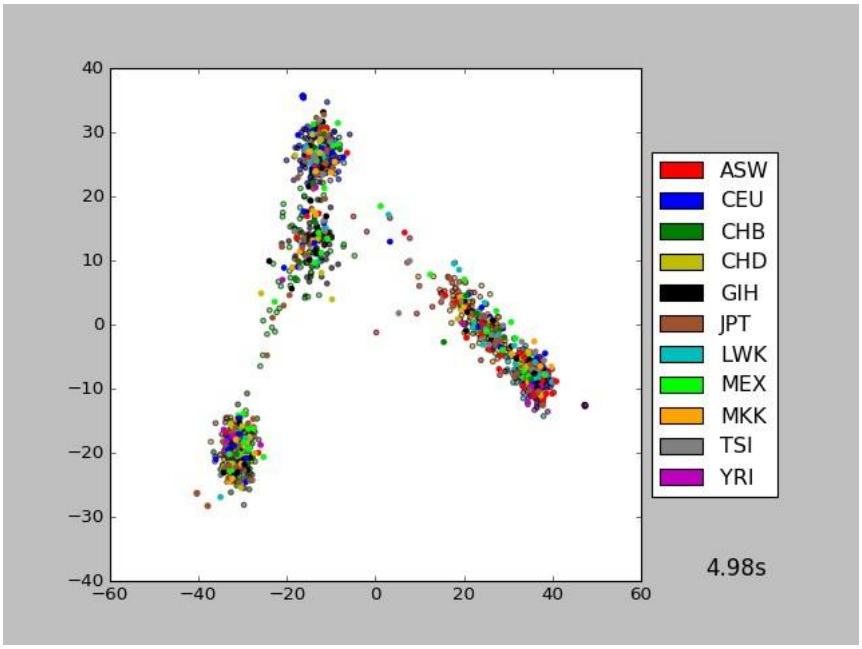
Decision Tree



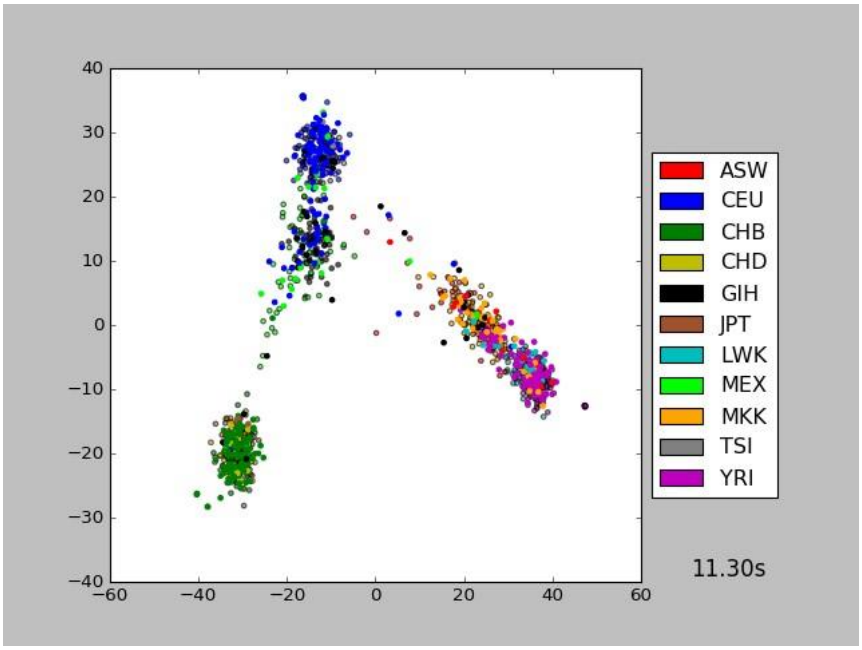
Spectral



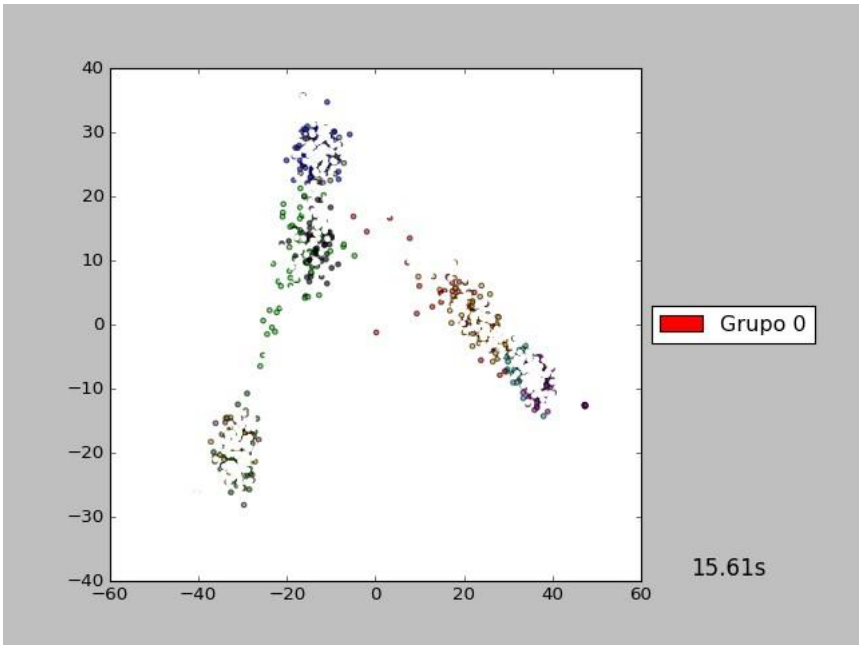
QDA



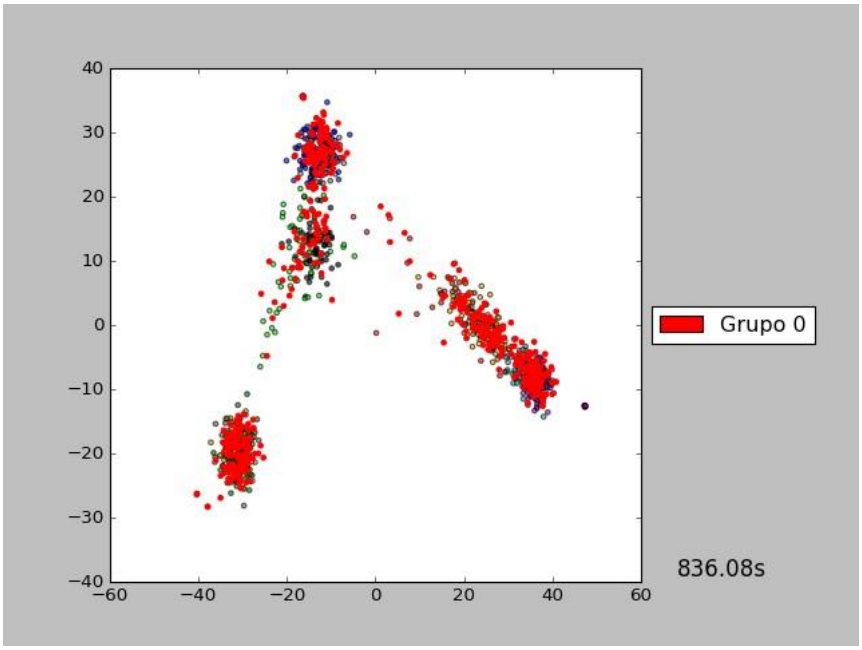
AdaBoost



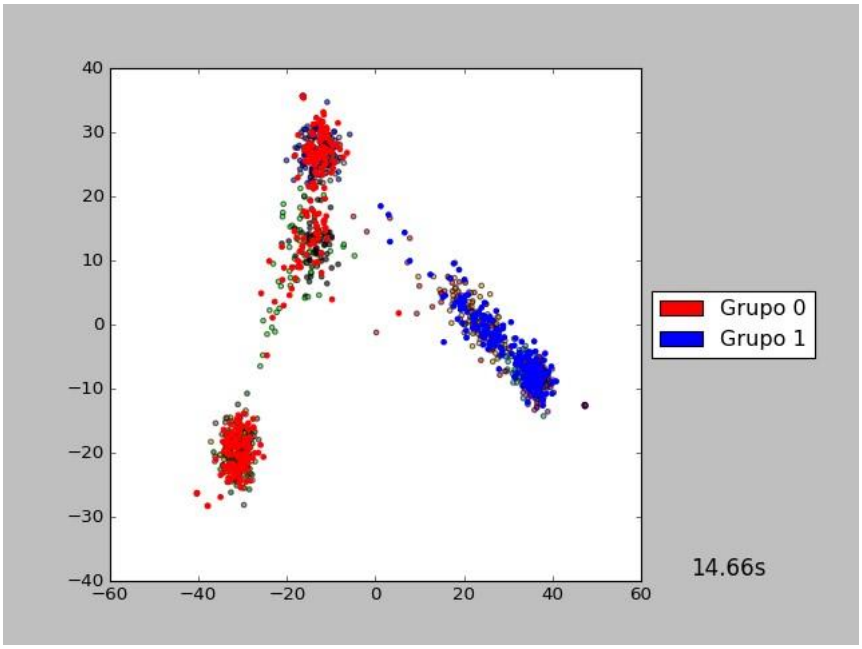
DBScan



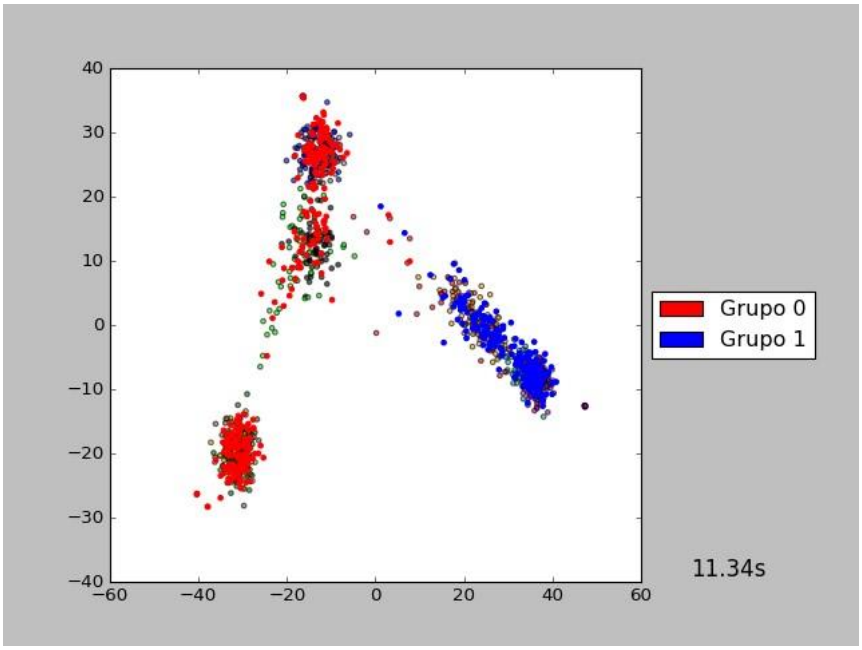
MeanShift



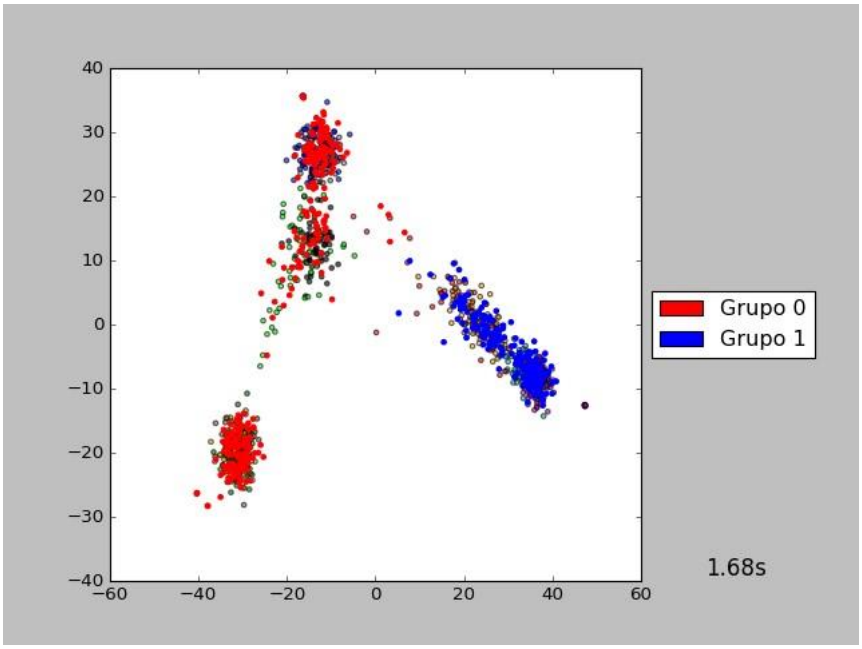
Birch



Ward



MiniBatchKMeans



ROC (sólo para clasificadores)