**Proyecto de Machine Learning**

**Problema:**

Dado un chat saber si en algún momento uno de los usuarios comienza a escribir diferente ya sea por una razón u otra.

Como sub-problema se tiene que dado un texto saber cuál de los integrantes del chat escribió dicho texto.

**Modelación del problema:**

Dado un chat se seleccionaron una cantidad de mensajes que comienzan en algún momento. En un diccionario se guarda por cada usuario todos sus mensajes. Haciendo uso de la biblioteca nltk de Python se realiza un procesamiento de texto donde por cada mensaje se va a eliminar las stopwords, o sea, las palabras que no aportan ningún significado, los signos de puntuación y se lematiza el texto, se llevan las palabras a su raíz gramatical. Luego de este procesamiento se vectorizará cada mensaje usando el TfidfVectorizer de la biblioteca sklearn y así se obtendrán los vectores de cada usuario por cada mensaje.

**Algoritmos de aprendizaje no supervisado usados en la resolución del problema:**

**- One Class SVM:**

Aprende una función de decisión para la detección de novedades: clasificar nuevos datos como similares o diferentes al conjunto de entrenamiento.

**- Robust Covariance:**

Detecta valores atípicos en un conjunto de datos distribuidos de Gauss.

**- Isolation Forest:**

Detecta anomalías mediante el aislamiento, en lugar de modelar los puntos normales.

**Algoritmos de aprendizaje supervisado usados en la resolución del problema:**

**-KNN**

**-Naive-Bayes**

**-Decision Tree**

**-Random Forest**

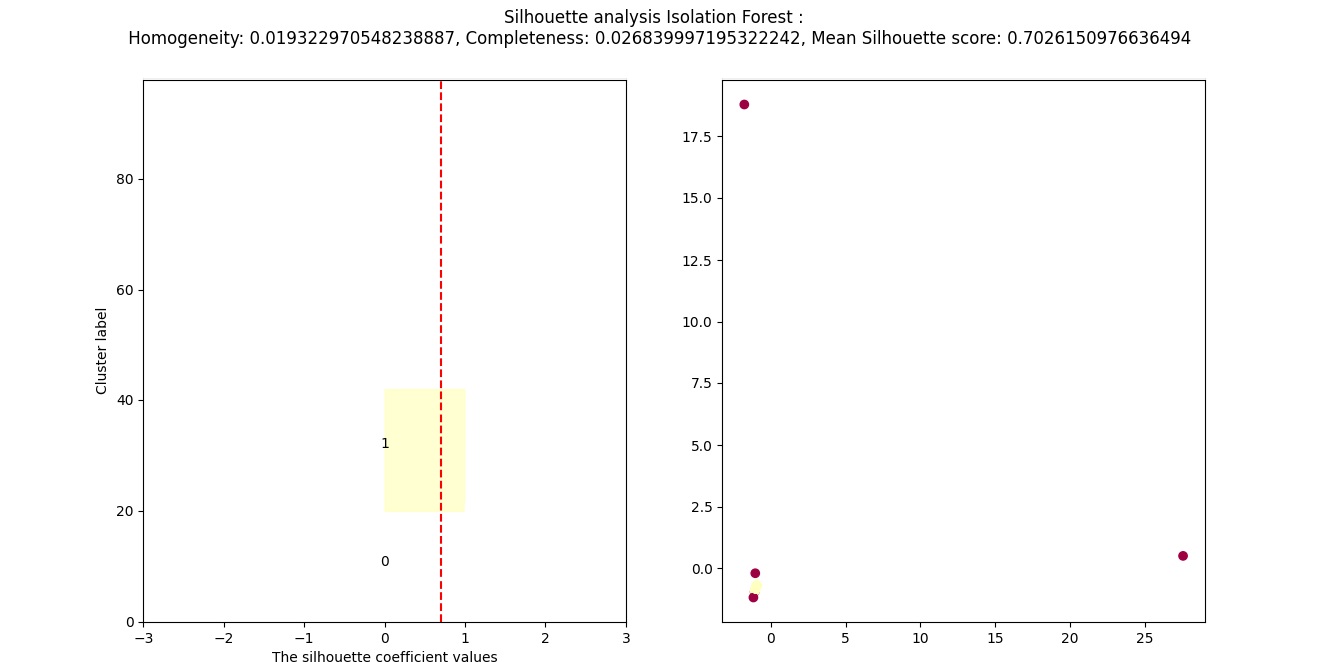
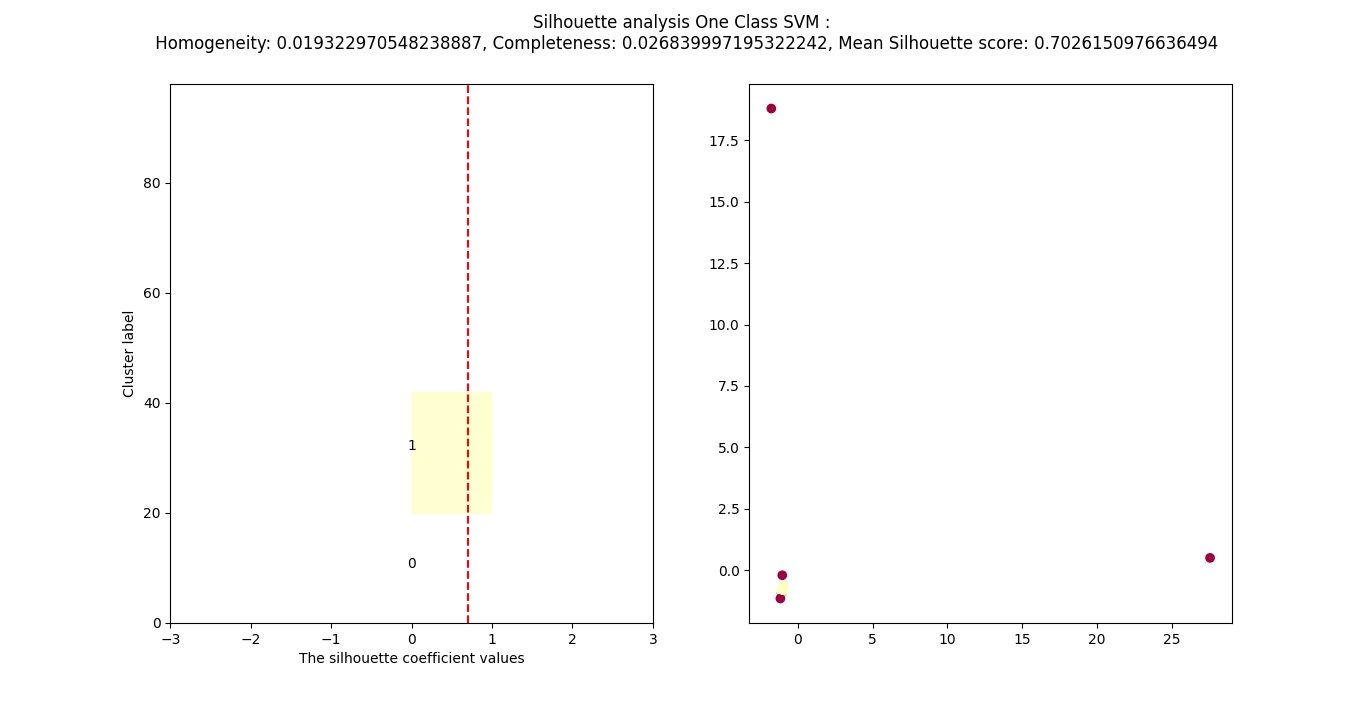
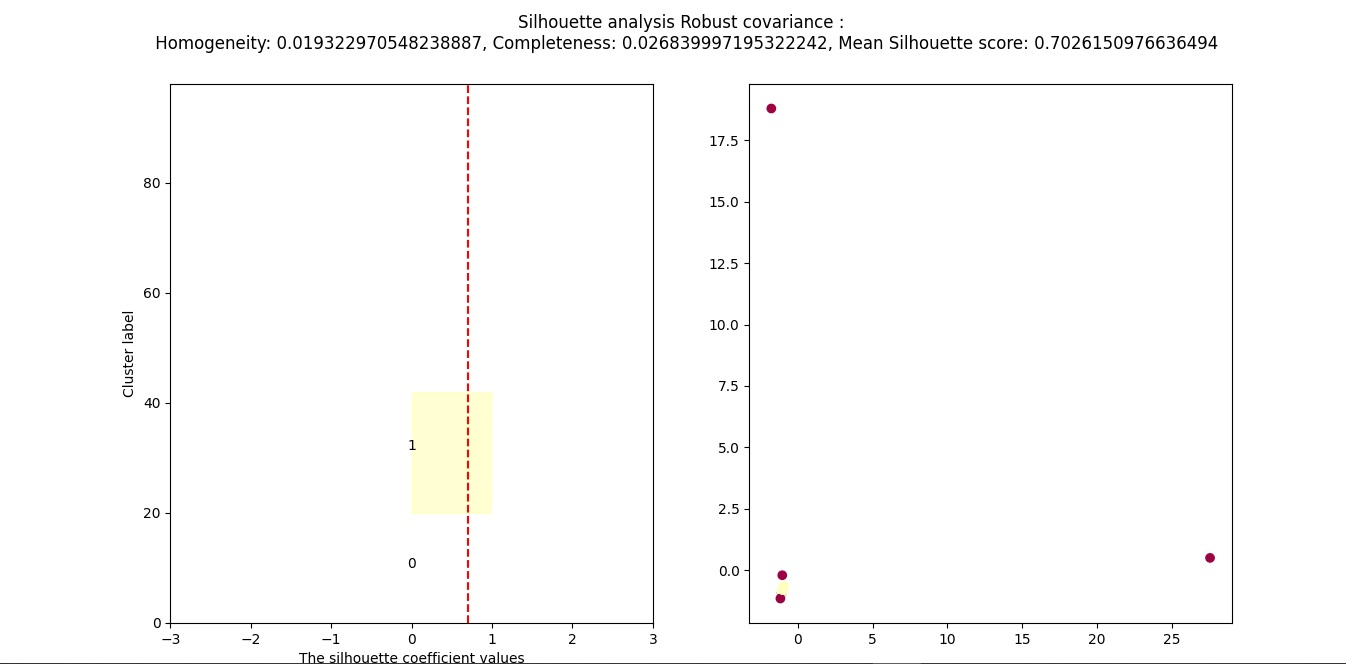
Estos algoritmos fueron los que estudiamos en clase. Escogimos KNN y Naive-Bayes porque trabajamos un problema similar en la CP2, aparte de que la idea de cercanía entre los conjuntos que tienen ambos algoritmos se asemejan a la idea que utiliza en un principio Novelty Detection, aunque enfocado en el aprendizaje supervisado por supuesto.

Por otro lado, utilizamos los modelos de Decision Tree y Random Forest porque demostraron ser de los modelos que mejores resultados obtenían en jeugos de datos de este estilo, siempre que se cuidara bastante el no realizar overfitting.

**Análisis de los resultados:**

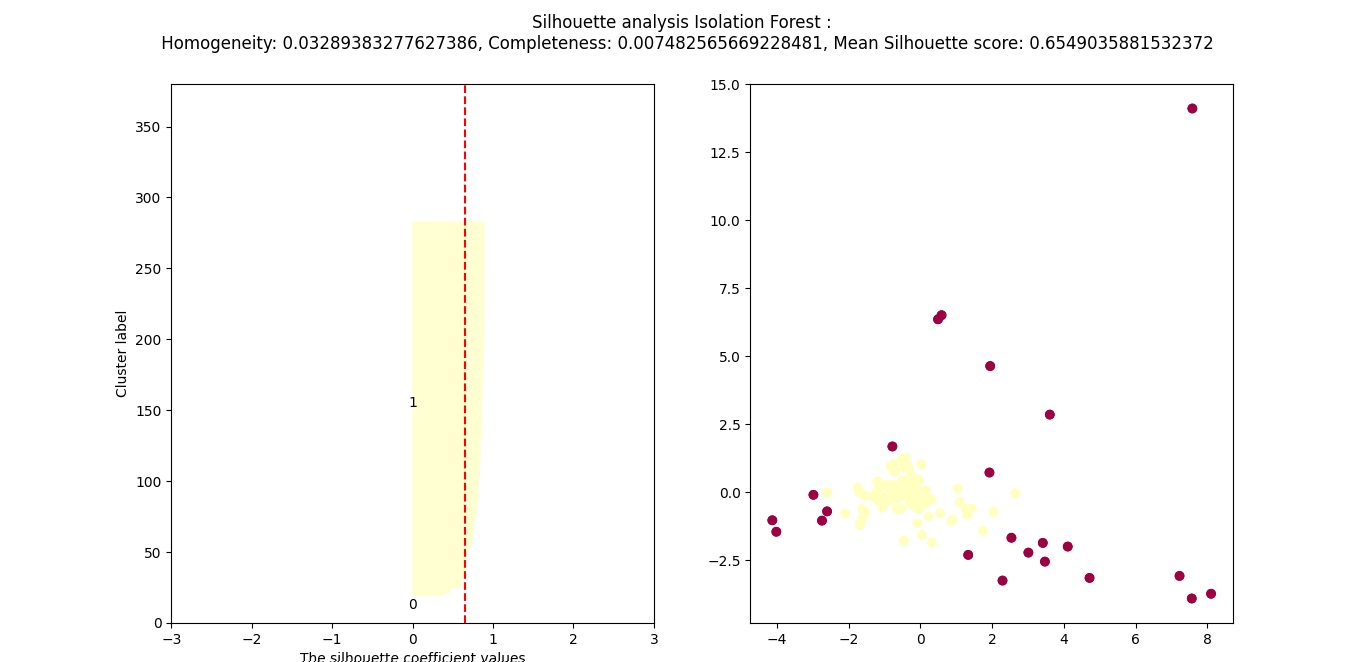
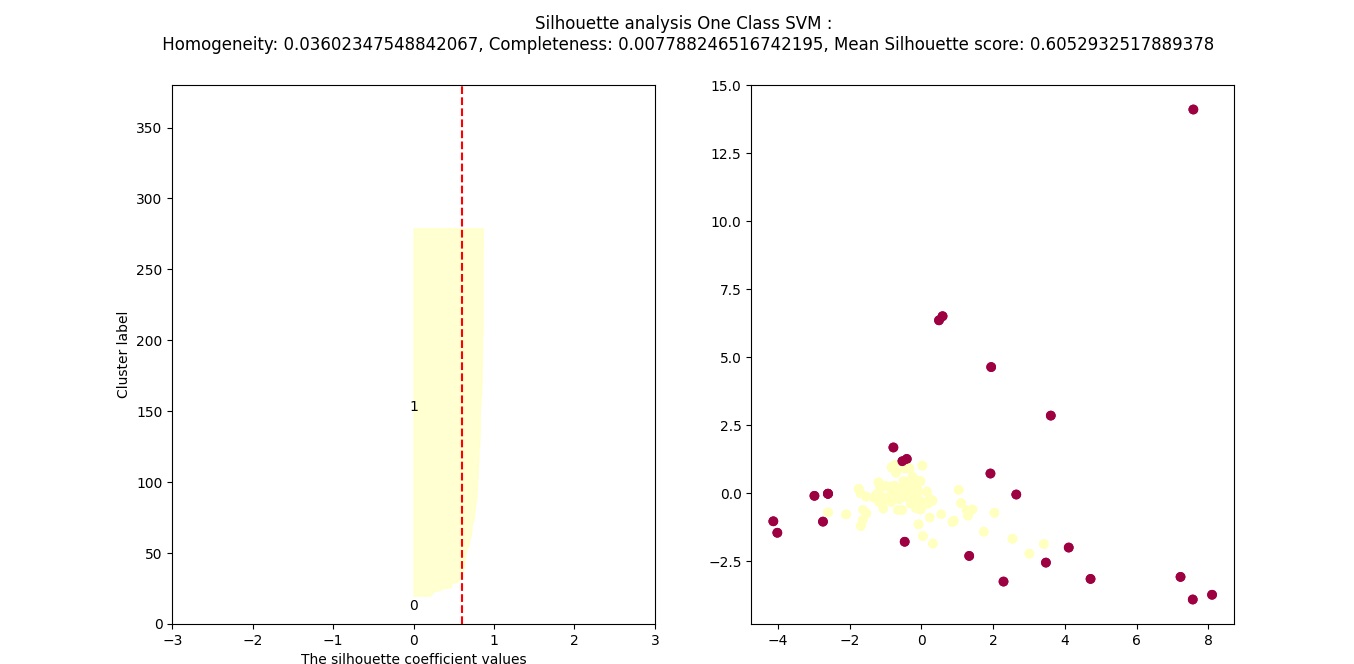
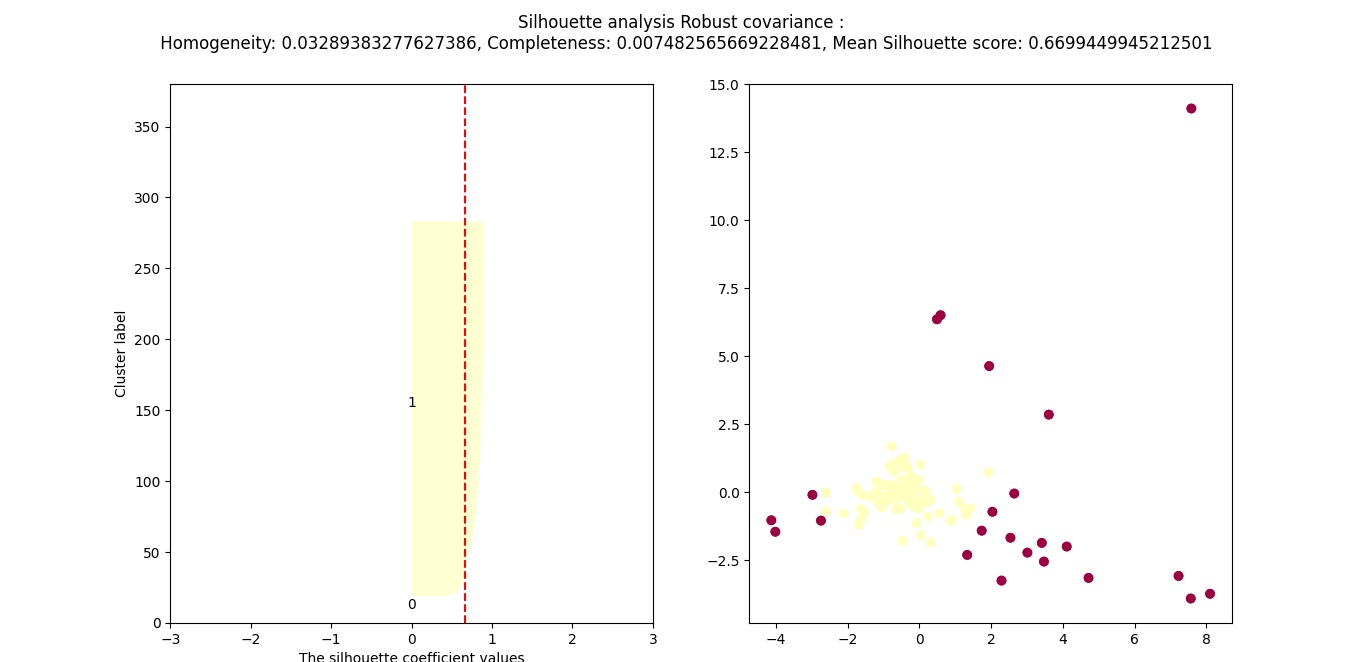
De las diferentes muestras escogimos una en particular de una conversación algo extensa entre 3 personas, y vimos utilizando los algoritmos no supervisados los siguientes resultados.

Para los datos de Airelys, quién participó poco en la conversación, los tres modelos obtuvieron los mismos resultados:



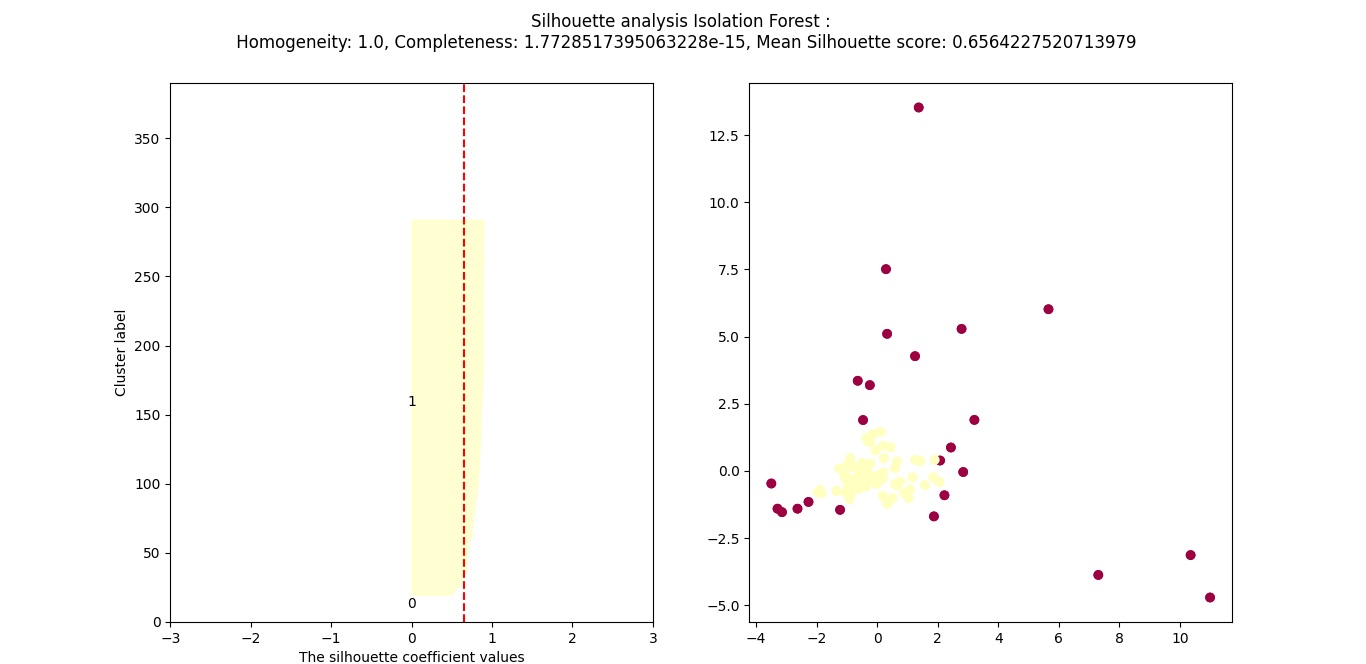
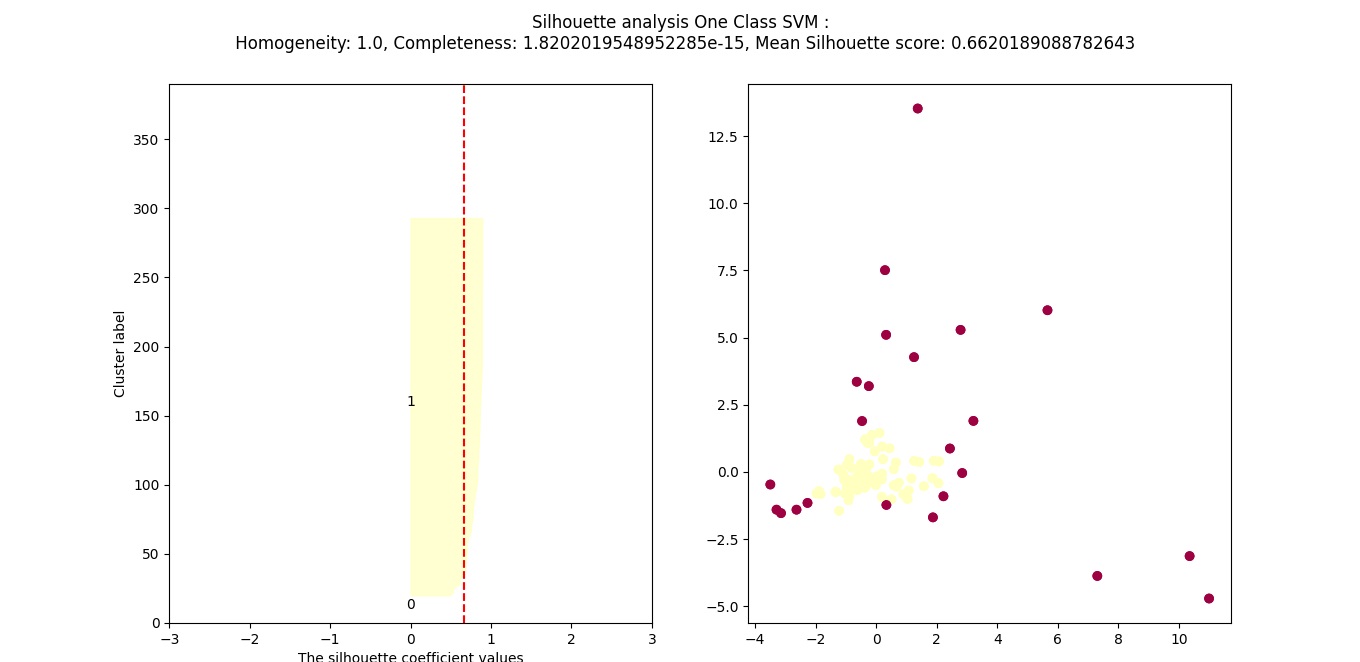
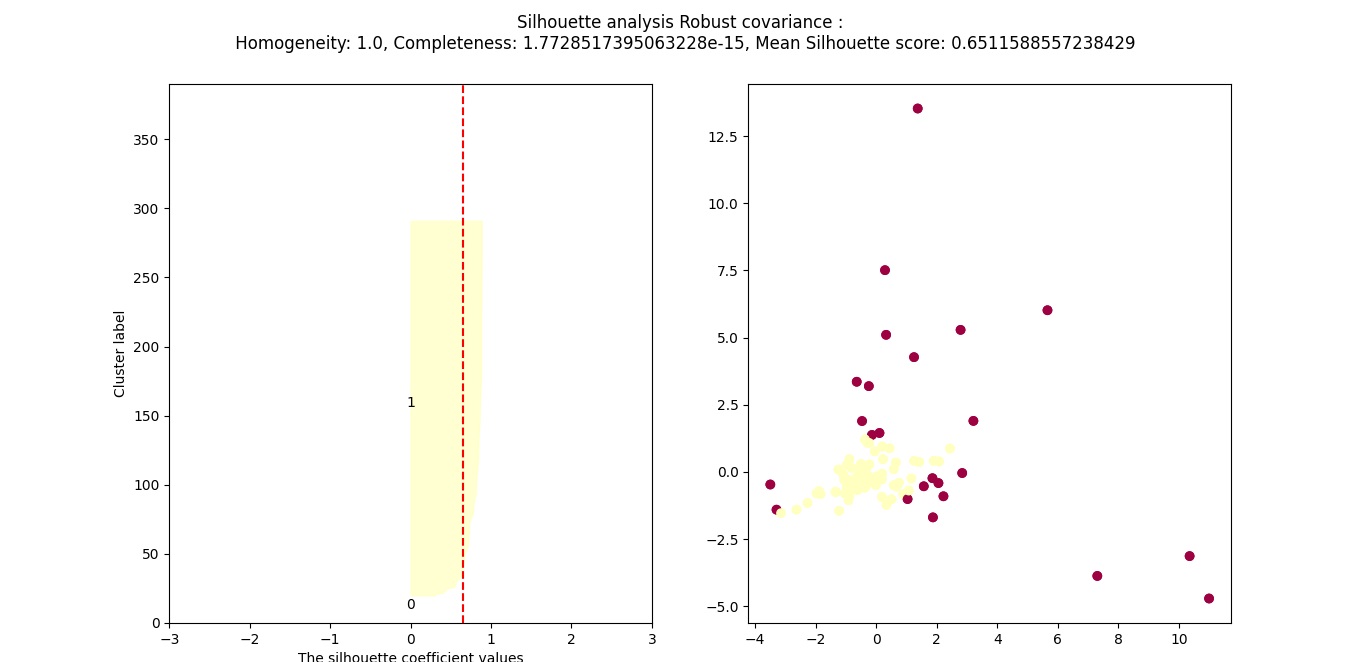
Por lo que, al menos para los algoritmos no supervisados, se necesitan una mayor cantidad de datos para que tengan un funcionamiento más disperso y no caigan en underfitting.

Analizemos los Datos para Alejandro:



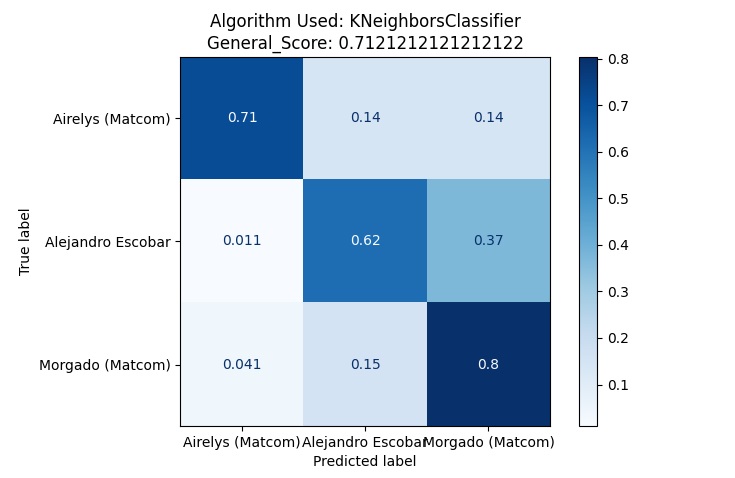
Ya estos datos se evidencian en mayores cantidades y Podemos notar que tienen un conjunto de clúster más grande, aparte de que los tres modelos toman decisiones distintas en cada caso.

Por ultimo veamos el caso de Morgado:

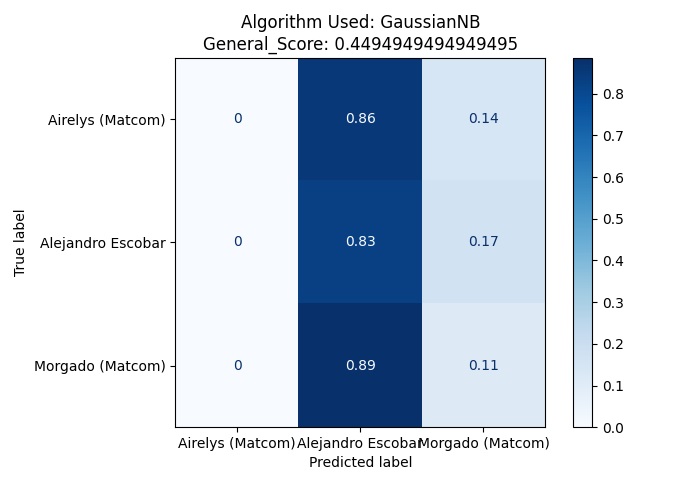


Los datos son similares en cantidad a los de Alejandro, pero Podemos notar que los modelos tienen menos diferencias en este caso a la hora de seleccionar qué puntos pertenecen al clúster.

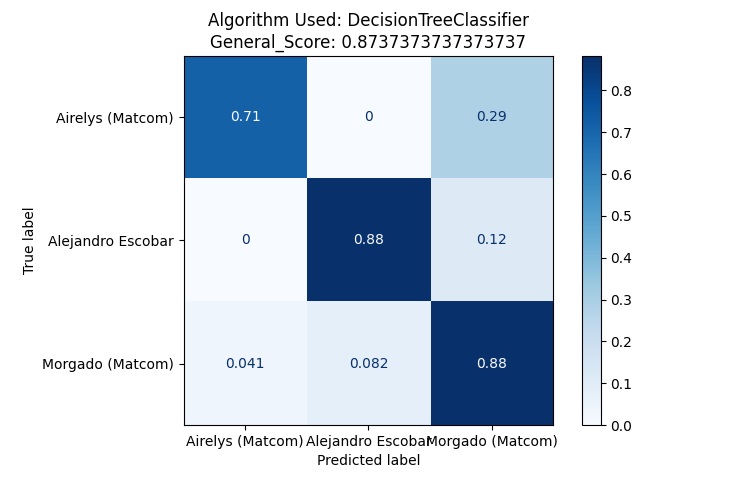
Por otro lado, probamos los algoritmos supervisados para, dado un mensaje, ver a quién podía pertenecer. En este caso los resultados fueron mucho más variados, siendo el más difícil siempre de clasificar aquellos mensajes que pertenecen a Airelys, los cuales por lo general en varias corridas terminaban siendo asignados mayormente a que eran mensajes de Morgado o Alejandro.



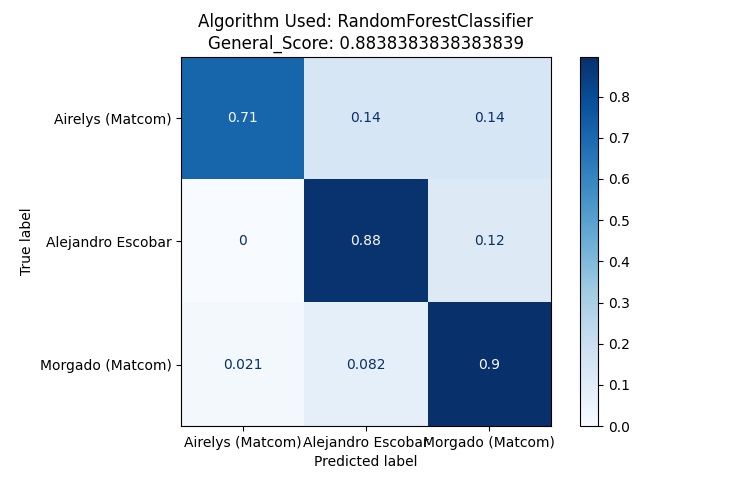
En el caso de KNN Podemos notar que la mayor parte de los datos de Morgado fueron clasificados correctamente con respect a los anteriores, incluso, que fue a quién más veces se le otorgó mensajes de forma errónea.



Naïve Bayes fue en todas las corridas el algoritmo con peores resultados (casi siempre por debajo de 0.5), en este caso no pudo clasificar ningún mensaje como que pertenecía a Airelys, en cambio, prácticamente todos los mensajes fueron otorgados a Alejandro (en algunas otras corridas sucedía lo mismo con Morgado, pero siempre inclinándose ferozmente sobre solo uno de los dos).



Decission Tree por su lado demostró buenos resultados mayormente, pudiendo tener altos resultados, e incluso 0 equivocaciones en algunas predicciones, donde nunca confundió un mensaje de Alejandro con Airelys ni viceversa, por tanto, aprendió bien de la diferencia entre estos dos. Aunque clasificó con igual exactitud a Alejandro y Morgado, Podemos notar que erróneamente clasificó muchos más mensajes como que eran de Morgado con respecto a los de Alejandro, y en una cantidad muy pequeña, confundió los mensajes de Morgado como mensajes de Airelys, por tanto, aprendió mejor de los datos de Morgado con respecto a los otros.



Por ultimo tenemos el algoritmo de Random Forest, quien casi siempre mostró los mejores resultados (llegando a sobrepasar el 9.0 de resultados). En este caso se ven resultados similares a Random Forest con respecto a Morgado, aunque disminuyendo las clasificaciones erróneas, sin contar unos mensajes de Airelys que clasificó como de Alejandro.