Análisis de datos para aplicaciones médicas

Santiago Mora Cruz |
Gabriel Reynoso Escamilla | A01643561
Guillermo Villegas Morales | A01637169

Junio de 2024

Descripción de la aplicación

Es una aplicación que utiliza un modelo de clasificación Support Vector Machine linear para identificar seis comandos de voz de manera *online* “Pausa”, “Continúa”, “Siguiente”, “Anterior”, “Más”, “Menos”. Se espera que el usuario diga la palabra y la aplicación sea capaz de identificarla correctamente.
Ya existen aplicaciones similares a estas ya sea integradas dentro de altavoces inteligentes o asistentes virtuales como lo son Alexa de Amazon o Siri de Apple [1].
La información que procesa el modelo no es el sonido que da el usuario sino las características extraídas con los Mel Frequency Cepstral Coefficients (MFCC).

Descripción de las características extraídas de los datos

Las características extraídas de los datos fueron mediante los MFCC, que se basan en la forma en la que los humanos percibimos el sonido de manera no lineal y somos más sensibles a los cambios en frecuencias bajas que en altas. Esto se hace aplicando a los datos Transformada de Fourier, Mel-Escalamiento (no lineal), Logaritmo en cada banda de frecuencia de Mel y, finalmente, Transformada Discreta de Coseno a los Logaritmos obtenidos.
Las características resultantes son, por lo tanto, las características más relevantes de la señal de audio en un formato compacto, y son valores numéricos continuos.

Resultados de la evaluación de clasificadores

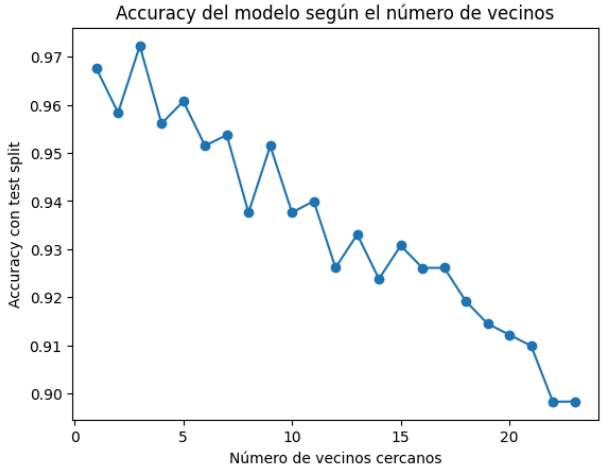
Primero se evaluó la metodología de diez modelos usando Validación Cruzada, a continuación se muestra la exactitud de cada uno y el *Recall* por clase, siendo estas, 1: Siguiente, 2: Anterior, 3: Más, 4: Menos, 5: Pausa, 6: Continúa.

	Accuracy	Recall 1	Recall 2	Recall 3	Recall 4	Recall 5	Recall 6
SVM Linear Kernel	0.95	1.00	1.00	1.00	1.00	0.89	0.80
SVM Radial Kernel	0.89	1.00	0.88	1.00	1.00	1.00	0.73
Gradient	0.84	0.95	0.94	0.67	0.95	0.89	0.87

Boosting							
Stochastic Gradient Descent	0.91	0.95	1.00	0.94	1.00	0.89	0.73
K-Nearest Neighbors	0.92	1.00	1.00	0.94	1.00	0.94	0.73
Random Forest	0.87	1.00	0.88	0.94	0.95	1.00	0.87
Passive Aggressive	0.94	0.95	1.00	0.94	1.00	0.89	0.80
Quadratic Discriminant	0.16	0.00	0.12	0.11	0.26	0.06	0.13
MLP Classifier	0.70	0.91	1.00	0.94	1.00	0.89	0.93
Linear Discriminant Analysis	0.93	1.00	0.94	0.89	1.00	1.00	0.80

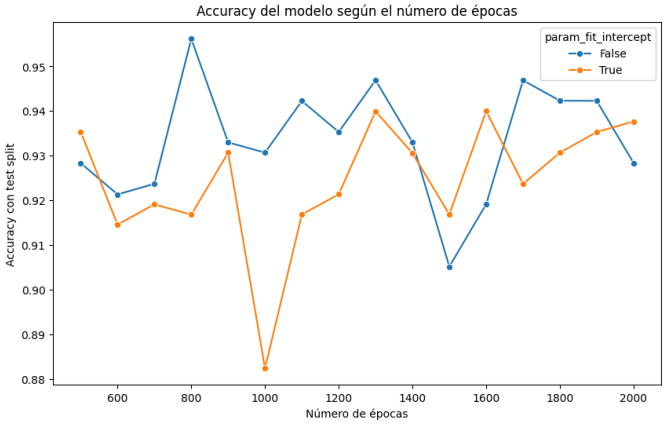
Con estos resultados decidimos usar el modelo Support Vector Machine con kernel lineal para la aplicación, esto porque tiene los mejores resultados en el *recall* de cada clase.

Después seguimos con la optimización de hiper parámetros, en este caso los parámetros de los clasificadores K-Nearest Neighbors (KNN) y Stochastic Gradient Descent (SGD) Classifier, esto porque aunque dieron buenos resultados aún contaban con buen rango de mejora.
La función *KNeighborsClassifier* cuenta con los parámetros *algorithm, leaf_size, metric, metric_params, n_jobs, n_neighbors, p,* y *weights.* El parámetro *n_neighbors* es el que indica el número de vecinos cercanos a la observación que entran para la clasificación. Haremos un *grid search* para este parámetro con un rango de 2 hasta 23, esto por ser cercano a la raíz cuadrada de nuestro número de observaciones total, para encontrar el mejor utilizamos los promedios de *accuracies* para cada valor.[4]



Determinamos que el mejor valor para el híper parámetro *n_neighbors* es 3 con un *accuracy* de 0.97.

Para el clasificador SGD evaluamos el parámetro *max_iter* este limita el número de épocas con las que se entrenará el modelo, por default son 1000. Tiene un rango de $[1,\infty)$. fit_intercept es un parámetro con valores True o False, define si el modelo buscará el intercepto o si asume que ya están centrados los datos. Haremos un *grid search* donde iteramos para max_iter desde 500, hasta 2000 con pasos de 100 épocas y a su vez con *fit_intercept* activado o desactivado.[3]

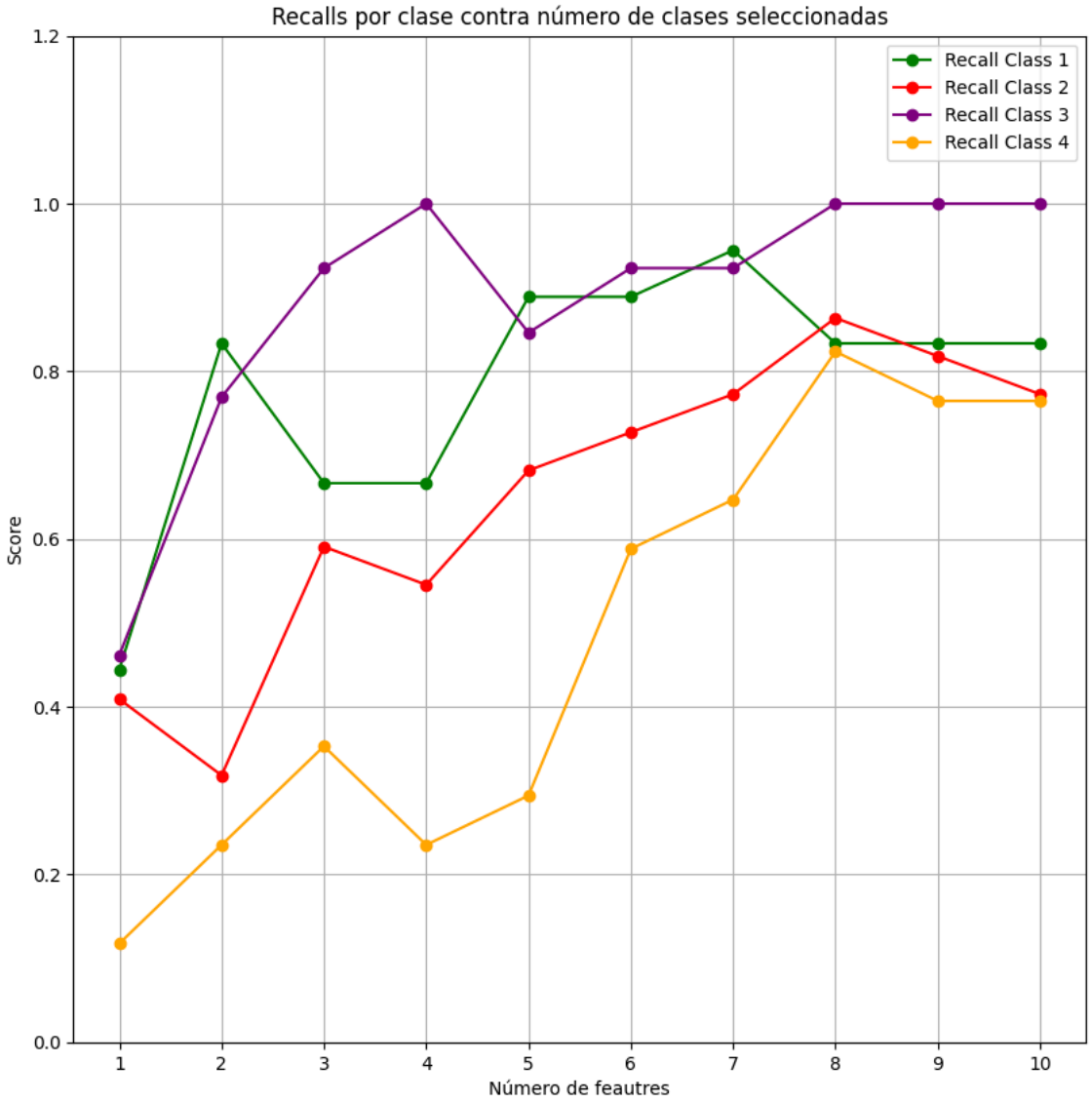


Como podemos observar el modelo alcanza su mejor accuracy cuando el *fit_intercept* está apagado y el número de épocas es 800. Esto nos dice que no hace falta entrenar con muchas más épocas y que el modelo se ajusta mejor si asume que los datos ya están centrados.

Por último realizamos selección de características a los dos modelos optimizados anteriormente.Para el modelo de K-Nearest-Neighbors utilizamos un wrapper. Escogimos este ya que el wrapper es más tardado de calcular, pero compensaba el hecho de que el hiperpara metro de vecinos cercanos es 3. Se iteró el wrapper escogiendo de 1 a 10 características y evaluando el accuracy, las características escogidas y sus accuracies se muestran a continuación

Características	Accuracy
['1966']	0.3
['1463' '1966']	0.45
['1463' '1708' '1966']	0.60
['1338' '1463' '1708' '1966']	0.61
['809' '1338' '1463' '1708' '1966']	0.68
['809' '1338' '1463' '1523' '1708' '1966']	0.77
['809' '1338' '1463' '1523' '1708' '1966' '2303']	0.76

['809' '1282' '1338' '1463' '1523' '1708' '1966' '2303']	0.81
['809' '1282' '1338' '1463' '1523' '1708' '1966' '2303' '2427']	0.84
['809' '1282' '1338' '1463' '1523' '1708' '1966' '2303' '2427' '2431']	0.84



Como se observa en la tabla, existe una mejora notable con cada iteración y cada variable nueva en consideración. Es muy interesante el hecho de que con 11 variables ya haya 0.84 de accuracy.

Por cuestiones de tiempo, decidimos correr un filter para seleccionar características evaluando con el clasificador Stochastic Gradient Descent. Las iteraciones van

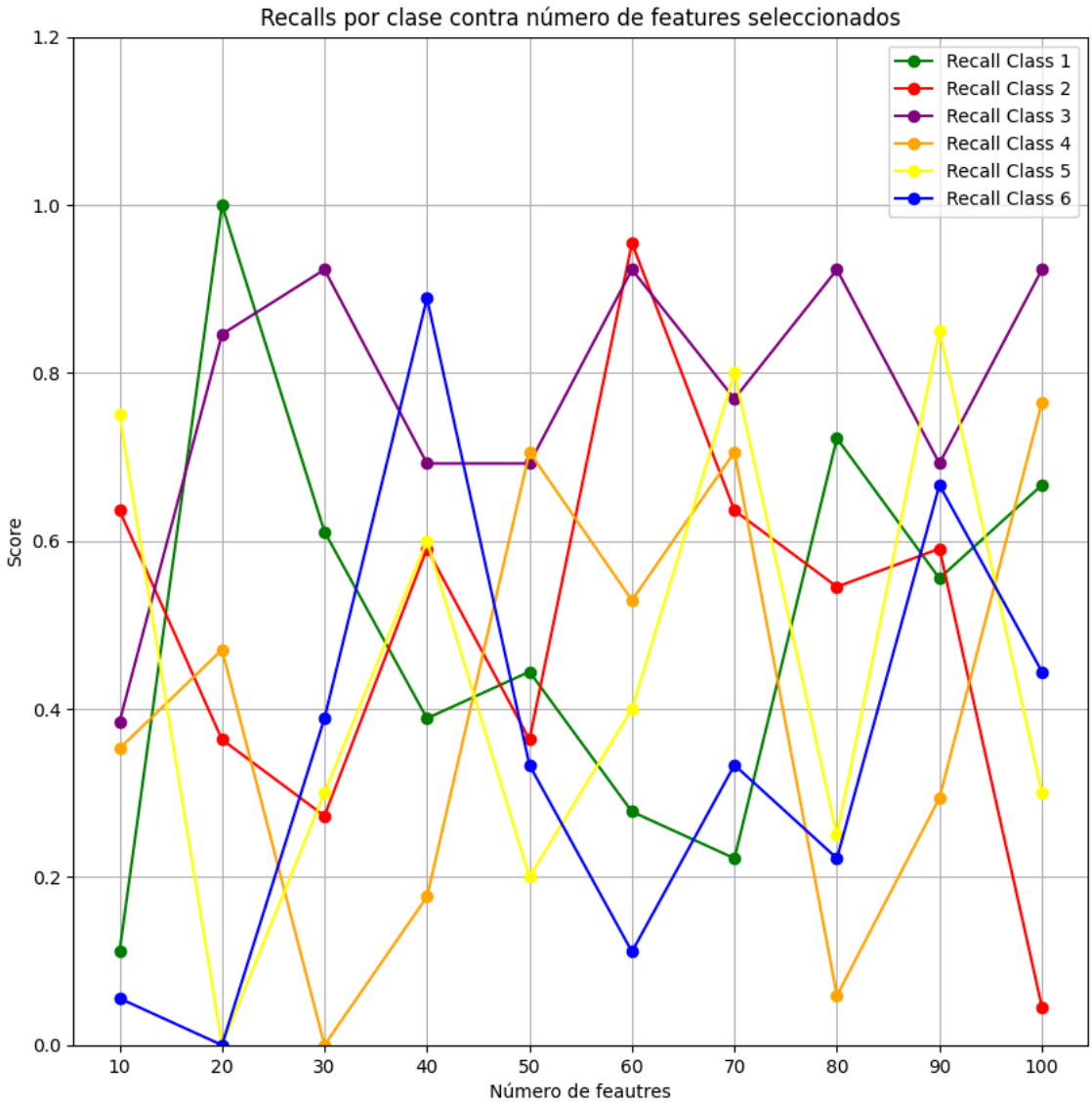
considerando de 10 hasta 100 variables con un tamaño de paso 10, con los hiper parámetros obtenidos de pasos anteriores, la tabla se muestra a continuación

Características	Accuracy
['1554' '1875' '1877' '1888' '1914' '1953' '1966' '1979' '1992' '2005']	0.40
['1489' '1515' '1554' '1619' '1864' '1875' '1877' '1888' '1914' '1916' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1992' '2005' '2252']	0.42
['1450' '1463' '1489' '1515' '1554' '1568' '1619' '1632' '1684' '1864' '1875' '1877' '1888' '1890' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1992' '1994' '2005' '2096' '2213' '2252']	0.39
['1450' '1463' '1489' '1515' '1541' '1554' '1568' '1619' '1632' '1684' '1763' '1776' '1838' '1841' '1864' '1875' '1877' '1888' '1890' '1903' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1992' '1994' '2005' '2007' '2010' '2096' '2174' '2213' '2252' '2356']	0.56
['1450' '1463' '1489' '1502' '1515' '1541' '1542' '1554' '1567' '1568' '1619' '1632' '1659' '1684' '1737' '1763' '1776' '1828' '1838' '1841' '1864' '1875' '1877' '1888' '1890' '1901' '1903' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1992' '1994' '2005' '2007' '2010' '2031' '2057' '2072' '2096' '2174' '2213' '2252' '2356']	0.44
['1437' '1450' '1463' '1476' '1489' '1502' '1515' '1541' '1542' '1554' '1567' '1568' '1606' '1619' '1632' '1633' '1659' '1684' '1737' '1750' '1763' '1776' '1828' '1838' '1841' '1849' '1864' '1875' '1877' '1888' '1890' '1901' '1903' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1992' '1994' '2005' '2007' '2010' '2018' '2031' '2057' '2072' '2096' '2174' '2187' '2213' '2226' '2239' '2252' '2356']	0.53
['1424' '1437' '1450' '1463' '1476' '1489'	0.57

'1502' '1515' '1516' '1541' '1542' '1554' '1567' '1568' '1606' '1619' '1632' '1633' '1659' '1672' '1684' '1710' '1737' '1750' '1763' '1776' '1828' '1836' '1838' '1841' '1849' '1851' '1864' '1875' '1877' '1880' '1888' '1890' '1901' '1903' '1906' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1992' '1994' '2005' '2007' '2010' '2018' '2031' '2057' '2070' '2072' '2096' '2135' '2174' '2187' '2213' '2226' '2239' '2252' '2356']	
['1424' '1437' '1450' '1463' '1476' '1489' '1502' '1515' '1516' '1541' '1542' '1554' '1567' '1568' '1606' '1619' '1632' '1633' '1659' '1672' '1684' '1697' '1698' '1710' '1724' '1737' '1750' '1762' '1763' '1776' '1815' '1828' '1836' '1838' '1841' '1849' '1851' '1864' '1875' '1877' '1880' '1888' '1890' '1901' '1903' '1906' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1984' '1992' '1994' '2005' '2007' '2010' '2018' '2031' '2033' '2057' '2070' '2072' '2083' '2096' '2135' '2161' '2174' '2187' '2213' '2226' '2239' '2252' '2265' '2356']	0.44
['1424' '1437' '1450' '1463' '1476' '1489' '1502' '1515' '1516' '1541' '1542' '1554' '1567' '1568' '1580' '1606' '1607' '1619' '1632' '1633' '1659' '1671' '1672' '1684' '1697' '1698' '1710' '1723' '1724' '1737' '1750' '1762' '1763' '1776' '1799' '1815' '1828' '1836' '1838' '1841' '1849' '1851' '1864' '1875' '1877' '1880' '1888' '1890' '1893' '1901' '1903' '1906' '1914' '1916' '1919' '1927' '1940' '1953' '1955' '1966' '1968' '1979' '1984' '1992' '1994' '2005' '2007' '2010' '2018' '2031' '2033' '2036' '2044' '2057' '2070' '2072' '2083' '2096' '2109' '2135' '2161' '2174' '2187' '2213' '2226' '2239' '2252' '2265' '2356' '2369']	0.61
['1398' '1424' '1437' '1450' '1463' '1476' '1489' '1502' '1515' '1516' '1528' '1529' '1541' '1542' '1554' '1567' '1568' '1580' '1606' '1607'	0.48

'1619' '1632' '1633' '1658' '1659' '1671' '1672' '1684' '1685' '1697' '1698' '1710' '1723' '1724' '1734' '1737' '1750' '1762' '1763' '1776' '1799' '1815' '1828' '1836' '1838' '1841' '1849' '1851' '1861' '1864' '1875' '1877' '1880' '1888' '1890' '1893' '1901' '1903' '1906' '1914' '1916' '1919' '1927' '1929' '1940' '1953' '1955' '1966' '1968' '1979' '1984' '1992' '1994' '2005' '2007' '2009' '2010' '2018' '2031' '2033' '2036' '2044' '2057' '2070' '2072' '2083' '2096' '2109' '2135' '2161' '2174' '2187' '2200' '2213' '2226' '2239' '2252' '2265' '2356' '2369']	

Tanto en la tabla como en la gráfica es perceptible el hecho de que incrementar las el número de variables no tiende a la mejora del accuracy, podemos decir que el modelo se confunde.



Resultados de la aplicación

Los resultados difirieron entre los miembros del equipo, con Santiago Mora la aplicación identificaba correctamente todos los comandos con excepción de “Pausa” en una ocasión, en el caso de Gabriel Reynoso ocasionalmente cometía errores en “Pausa” y el comando “Continúa” no lo identificaba, esto coincide con el *recall* de la clase continúa fue el menor al evaluar metodología con 0.80.

Conclusiones

Gabriel Reynoso Escamilla:
Este proyecto nos permitió explorar más la evaluación de metodologías de modelos de clasificación con información más “real” que la que se trabajaba en clase, además pudimos usar el modelo para clasificar de manera online.

También fue útil comenzar a relacionar la funcionalidad del modelo en relación con los datos, al ser lineales las características del MFCC fue claro porque el SVM de kernel lineal fue el adecuado para la clasificación.

Santiago Mora Cruz:
Al desarrollar un proyecto que compara distintos modelos de clasificación y comparar con las actividades de clasificación hechas anteriormente, fuimos capaces de observar que no necesariamente un clasificador es siempre bueno o malo (basado en el recall), sino que hay clasificadores que sirven para diferentes proyectos y que funcionan mejor con datos de diferente naturaleza. En este caso, por ejemplo, para trabajar con voz y con características extraídas de ella, los modelos que, generalmente, dieron mejores resultados son los lineales; aunque para algunas tareas realizadas anteriormente estos clasificadores tenían un rendimiento más bajo que otros. Además de esto, nos dió un acercamiento a lo que es trabajar con datos que uno mismo genera, y como es todo el proceso de recolección, preprocesamiento y entrenamiento. En general, nos permitió tener un acercamiento al procesamiento de señales y el desarrollo de modelos que funcionen *online*, pero además nos ayudó a reforzar conocimiento previamente visto y a comparar modelos tanto que ya conocíamos como que no.

Guillermo Villegas Morales
Durante el desarrollo del trabajo presente tuvimos la oportunidad de aplicar las técnicas y conceptos aprendidos en el transcurso de Modelación del Aprendizaje con Inteligencia Artificial. Las estrategias empleadas y los errores cometidos nos hicieron reforzar nuestro conocimiento de aprendizaje supervisado. Por los resultados obtenidos podemos reconocer la importancia de metodologías como la división de la base de datos en train y test, el cross validation, feature selection, así como tuvimos la oportunidad de conocer un poco del procesamiento de señales y las interfaces para aplicar los modelos predictivos. Considero que ahora somos capaces de resolver problemas con aprendizaje supervisado y no supervisado, comparar diferentes modelos predictivos y evaluar la metodología empleada

Referencias

[1] Hoy, M. B. (2018). Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. *Medical Reference Services Quarterly*, 37(1), 81–88. https://doi.org/10.1080/02763869.2018.1404391

[2] Divy Dwivedi, Ashutosh Ganguly, V.V. Haragopal,6 - Contrast between simple and complex classification algorithms, *Statistical Modeling in Machine Learning,* 2023, Pages 93-110, ISBN 9780323917766, https://doi.org/10.1016/B978-0-323-91776-6.00016-6.

[3] Scikitlearn (2024). SGDClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

[4] Scikitlearn (2024). KNeighborsClassifier. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html