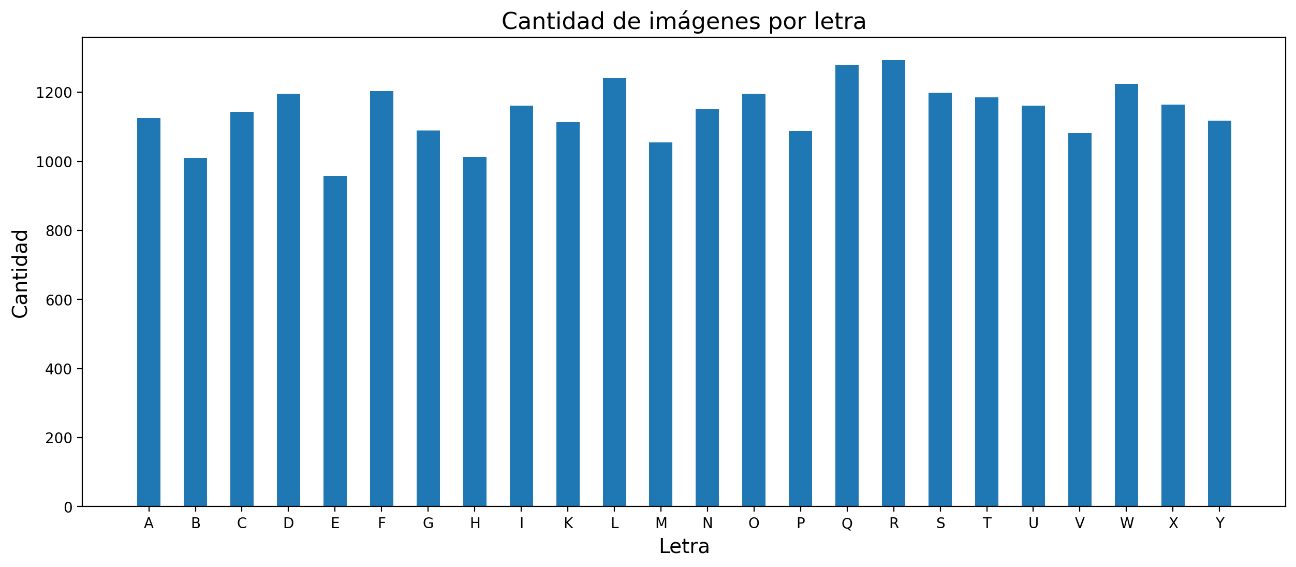
Introducción:

En este trabajo se realizaron implementaciones de modelos de calificación tanto binario como multiclase donde se crearon modelos de predicción para diferenciar a partir de imágenes de letras en lenguaje de señas .

Análisis Exploratorio de Datos:

Esbozamos el trabajo partir del DataSet “Sign Language MNIST”, en donde se encuentran imágenes de señas del abecedario, donde cabe destacar que no todas las letras se encuentran allí, ya que la “z” y la “j” no se producen a partir de señas, si no a partir de una gesto-forma. Esta base de datos posee como atributos una Etiqueta la cual indica de que letra se trata y luego atributos que indica un numero el cual se encuentran entre el 0 y el 255 los cuales son los pixeles de la imagen. En total cada imagen posee 728 pixeles, redimensionando estos pixeles a un matriz de 28 pixeles de ancho por 28 de altos y convirtiendo está en un mapa de color, obtenemos una imagen de la seña.

A partir del siguiente grafico observamos la gran cantidad se imágenes que obtenemos de este DataSet de cada letra:



A partir de este grafico podemos determinar que no existen una gran diferencia entre la cantidad de muestras por cada letra.

Con el objetivo de determinar a partir de una imagen, de que letra se trata, consideramos que hay pixeles mas importantes que otros. Para ello recreamos imágenes de algunas letras

Imagen que contiene Texto

Descripción generada automáticamenteImagen en blanco y negro

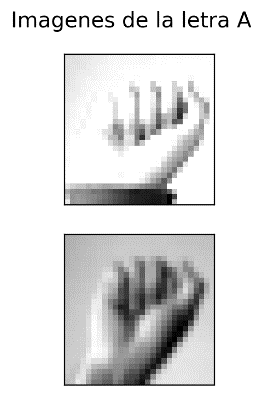
Descripción generada automáticamente con confianza bajaImagen que contiene Código QR

Descripción generada automáticamenteImagen que contiene Icono

Descripción generada automáticamente

En todas estas imágenes se puede observar algo en común, las señas suelen encontrarse en su mayoría en el centro de la imagen y luego hay un fondo que se aprecia diferente a la mano. Es por ello por lo que consideraremos como pixeles importantes los que representen una parte de la mano y como menos importante uno el cual represente el fondo. Por lo que si tuviéramos que descartar pixeles serían los del fondo.

Las imágenes por la pobre cantidad de pixeles que posee, a veces es difícil determinar una seña de otra. Por ejemplo, la letra A de la E o la letra M de la N

 Imagen en blanco y negro

Descripción generada automáticamente con confianza media Imagen que contiene Texto

Descripción generada automáticamenteImagen que contiene Código QR

Descripción generada automáticamente

Es por ello por lo que más adelante determinaremos un método para encontrar pixeles los cuales puedan determinar una letra de otra. Sin embargo, la mayoría poseen grandes diferencias.

Una ventaja que posee este DataSet es que las imágenes de las letras suelen ser todas iguales con sus pequeñas diferencias. Por ejemplo, aquí tenemos 25 imágenes de las letras A y L donde se diferencian por las sombras o inclinación de la mano:

 Imagen en blanco y negro

Descripción generada automáticamente con confianza media

Esta base de datos al poseer esta extraña forma genera que el proceso de exploración de los datos sea mas demandante, ya que estamos tratando con números los cuales representan una sombra en un mapa de calor donde sabemos algo de información a partir de cada “*label*”. Sin embargo, es imperante realizar este mapa para poder deducir algo de los datos.

Experimento Modelo AoL:

Con este experimento queremos lograr crear modelos KNN para predecir a partir de una imagen si se trata de una “L” o una “A”. Podemos observar las imágenes de estas letras en las figuras 4.1 y 4.2.

En primer lugar, obtuvimos todas las muestras correspondientes en un data set llamado “Muestras AL” que contenía a las letras de interés y determinamos a partir del siguiente grafico si las muestras estaban normalizadas.

Gráfico, Gráfico circular

Descripción generada automáticamente

Podemos observar en este gráfico, que las muestras dentro de este dataframe están normalizadas y no será un problema la diferencia entre la cantidad de “A” y “L”. A estas muestras las dividimos en 2 sets, uno de entrenamiento y otro de prueba para evaluar si el modelo es eficaz, siendo la prueba el 20% de la muestra original.

Modelo AoL Automático:

En primer lugar, quisimos generar el modelo KNN con solo 3 pixeles el cual toma como entrada la mejor cantidad de vecinos que arroje la evaluación “K-Fold Cross Validation” tienendo como hiperparametros la lista de números que va de 1 a 10. Los cuales no son seleccionado al azar. La elección fue a partir de sobreponer todas las “A” y todas las “L”, es decir sumar todos los pixeles para luego reescalar para que se encuentren entre 0 y 255. Luego restamos la sobreposición de “A” con la de “L” y viceversa. Generando estas imágenes:

Gráfico

Descripción generada automáticamente

En estas imágenes nos importa tomar los pixeles los cuales sean mas blancos o mas negros. Ya que son los que mayor variabilidad poseen y es por ello que será más fácil detectar de que letra se trata con estos pixeles. Mientras que los que se mantienen grises son pixeles que no poseen variabilidad luego estos no son tan relevantes por lo que dificultarían tomar una decisión acerca de que letra se trata; esto se corresponde a lo mencionado anteriormente, que un píxel en la mano tiene mayor importancia que a uno del fondo. Remarcamos estos pixeles relevantes:

Gráfico, Gráfico de burbujas

Descripción generada automáticamente

Por lo tanto, tomamos 2 sets de pixeles relevantes, {491,492,518} y { 301,274,246} , 2 set de pixeles no relevantes, {283,784,10} y {1,420,28} , y 2 aleatorios, {448,462,599} y {648,320,187}.

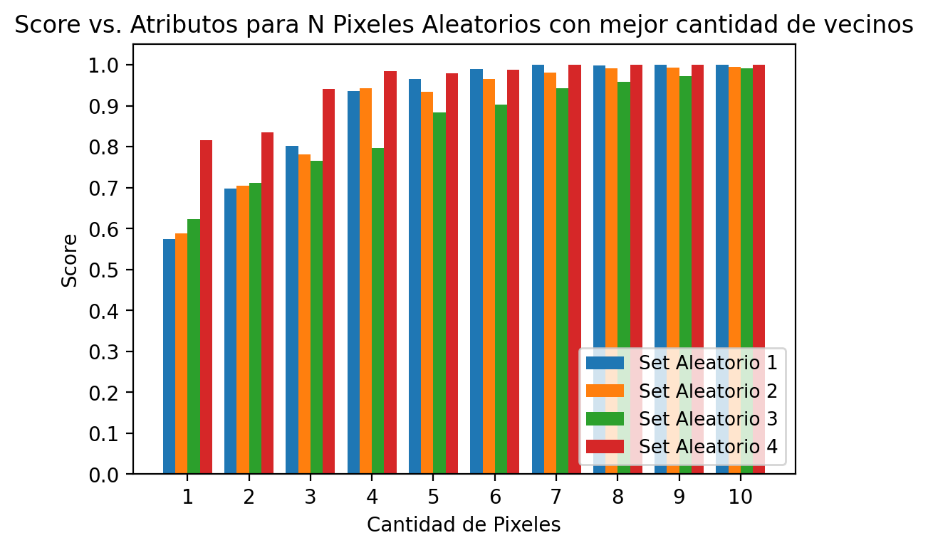
Evaluamos el modelo en base a la exactitud con nuestras muestras de prueba y estos fueron los resultados:

Gráfico, Gráfico de barras

Descripción generada automáticamente

A partir de esta imagen podemos concluir que existe una relación en la importancia de la elección de pixeles para obtener un mejor *score.* Sin embargo, el modelo automático con pixeles no relevantes obtiene un buen resultado también. Además, una peculiaridad de los modelos generados a partir de estos pixeles es que el mejor hipeparametro siempre era 1 vecino.

En base a este modelo automático quisimos ver que sucede cuando agregamos más pixeles por ello generamos 4 sets aleatorios de pixeles y a medida que agregábamos pixeles fuimos generando modelos y evaluándolos, generando este grafico como resultado:

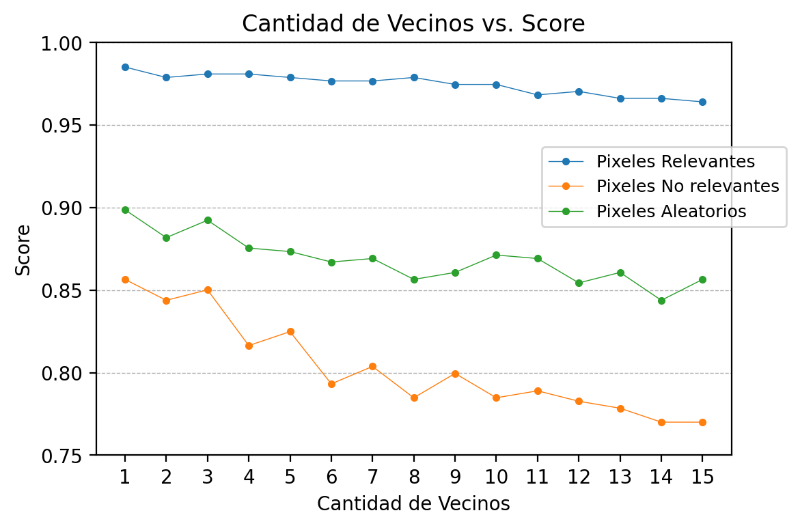


Se visualiza que existe un cierto tipo de patrón en donde el modelo predice mejor cuando posee mayor cantidad de pixeles. Esto nos llevo a generar el siguiente modelo no automático.

Modelo AoL No Automático:

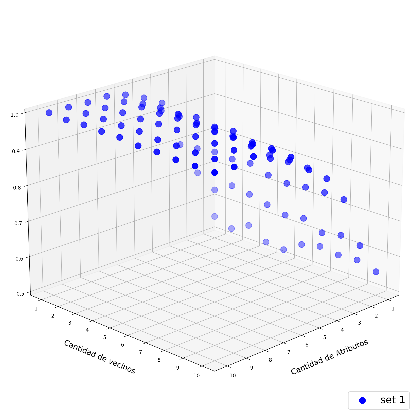
Este modelo corresponde a otro KNN en donde toma como entrada la cantidad de vecinos y pixeles deseada por el usuario.

En primer lugar, tomamos un set de pixeles por cada tipo de los mencionados anteriormente. Y los evaluamos para distinta cantidad de vecinos en base a la exactitud:



Se puede divisar que a medida que el modelo evalúa con mayor cantidad de vecinos obtiene un peor resultado. Y que cuando toma como vecinos igual a 1, siempre obtiene el mejor resultado.

Por último, para este tipo de modelo quisimos evaluar para distinta cantidad de atributos y vecinos el score. Generamos 4 sets de pixeles que a medida que agregábamos valores lo íbamos evaluando con distinta cantidad de vecinos:

Gráfico

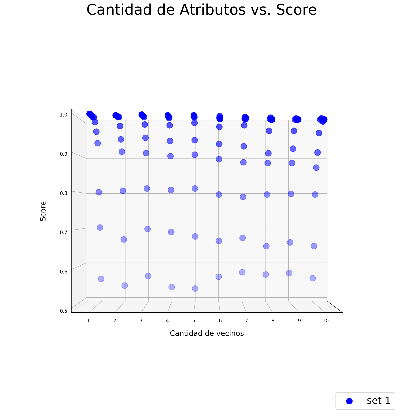
Descripción generada automáticamenteGráfico

Descripción generada automáticamenteGráfico

Descripción generada automáticamente

En esta perspectiva no se puede apreciar así que cambiamos la perspectiva a los dos lados:

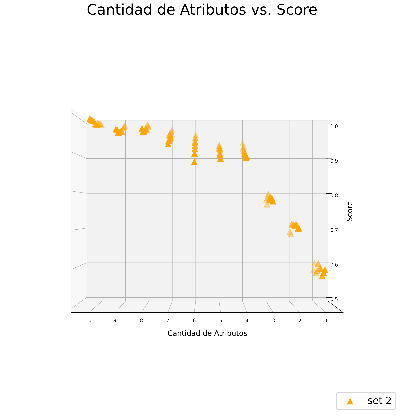
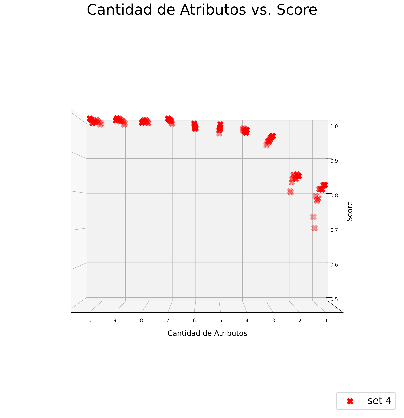
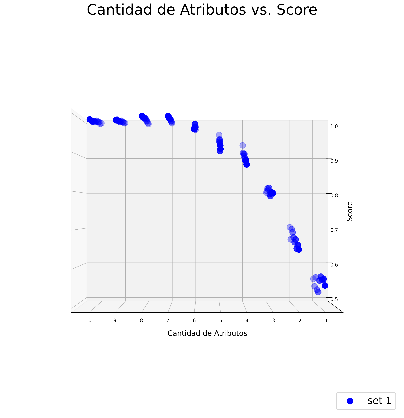
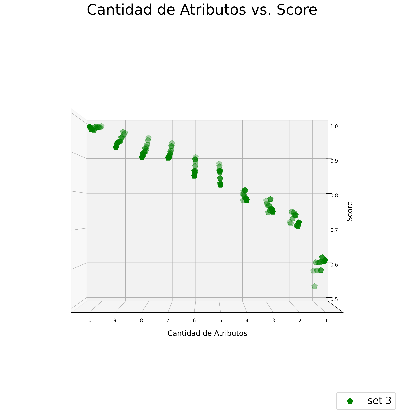
Gráfico, Gráfico de dispersión, Gráfico de cajas y bigotes

Descripción generada automáticamenteGráfico, Gráfico de dispersión, Gráfico de cajas y bigotes

Descripción generada automáticamenteGráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

En estos graficos no se aprecia alguna correlacion clara sin embargo, cuando cambiamos nuevamente la perspectiva:



Estos gráficos remarcan que mientras más pixeles se utilicen, sin importar su relevancia, más fácil será detecta si se trata de una A o una L.

Experimento Modelo Vocales:

En este experimento, queremos a partir de un modelo predictivo de árbol de decisión para calificación multiclase predecir que vocal es.

En primer lugar, obtuvimos en un solo dataframe todas las muestras de todas las vocales y tomamos todos los pixeles. Dividimos esta muestra en un 80% para el entrenamiento del modelo y el restante 20% como muestra de prueba para evaluar el modelo.

En primer lugar, evaluamos este modelo con varias profundidades. Estos fueron los resultados:

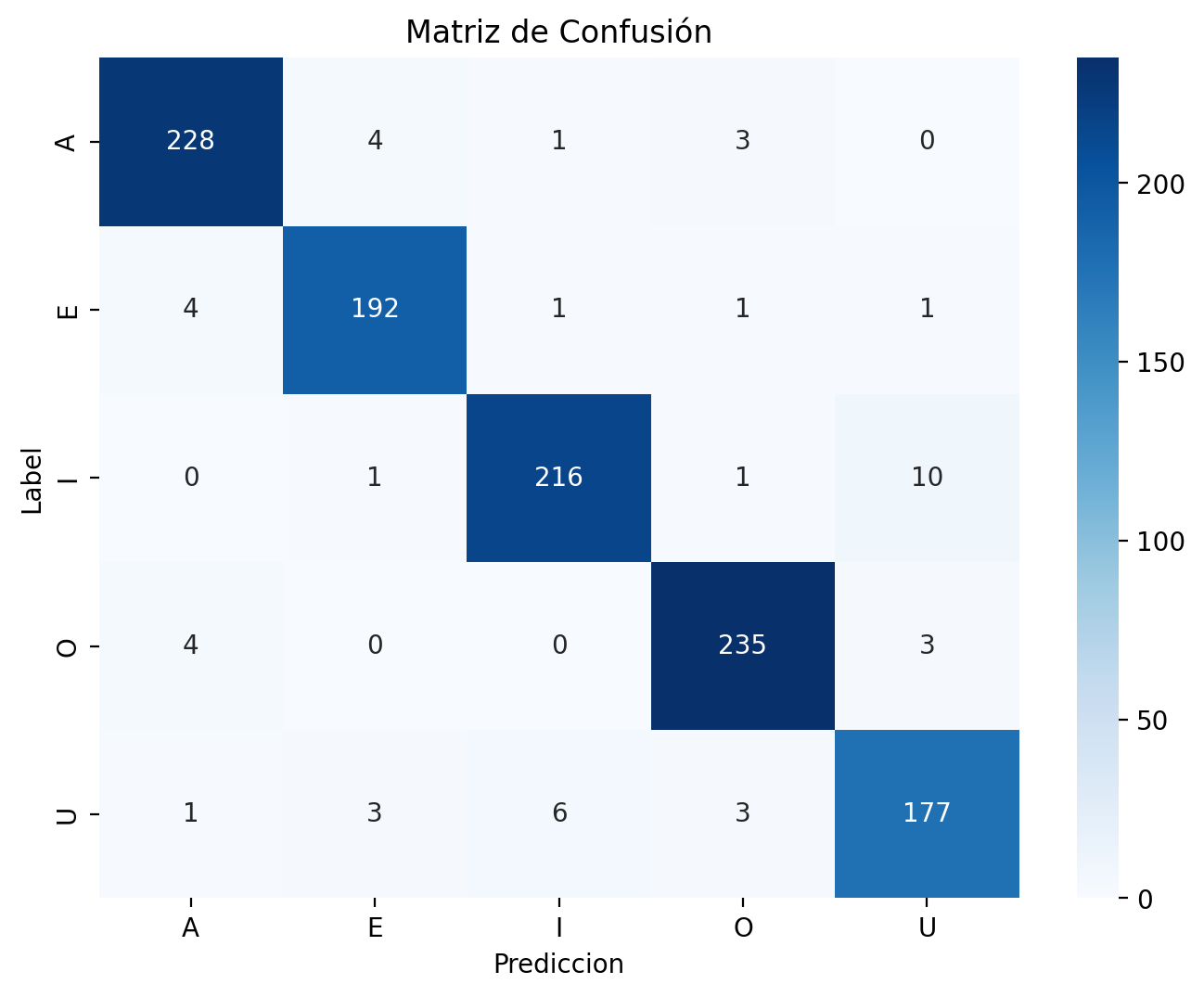
Gráfico

Descripción generada automáticamente

Se puede observar que a medida que hay mayor profundidad se obtiene un mejor score.

Luego creamos el árbol de decisión, donde ultizamos “*k-fold cross validation*” con criterios Gini y Entropia, además las porfundidades van de 7 a 20. Este va a ser el árbol con los mejores criterios donde arroja un score de 0.95 con una profundidad 11 y con el criterio Gini.

A partir de los datos generamos una matriz de confusión:



Conclusiones: