En la actualidad el poder manejar y obtener información de un conjunto de datos es de mucho valor para cualquier industria, y muchas otras aplicaciones. Es por esto que el saber utilizar modelos de inteligencia artificial es de alta importancia. En este documento se trabajó el famoso dataset de iris con el modelo de clasificación de K-Nearest Neighbors para analizar su desempeño. Este trabajo no se enfoca en la parte de preprocesamiento de los datos, si no en el entrenamiento y validación del mencionado modelo.

El dataset de iris cuenta con 150 instancias, se trata de un dataset de poco tamaño pero con la cantidad suficiente de registros para poder implementar un modelo. Es un conjunto de datos ideal pues los valores objetivo son categóricos cuyas características son de tipo numérico, lo cual es mejor para K-Nearest Neighbors puesto a que trabajar con las distancias entre puntos.

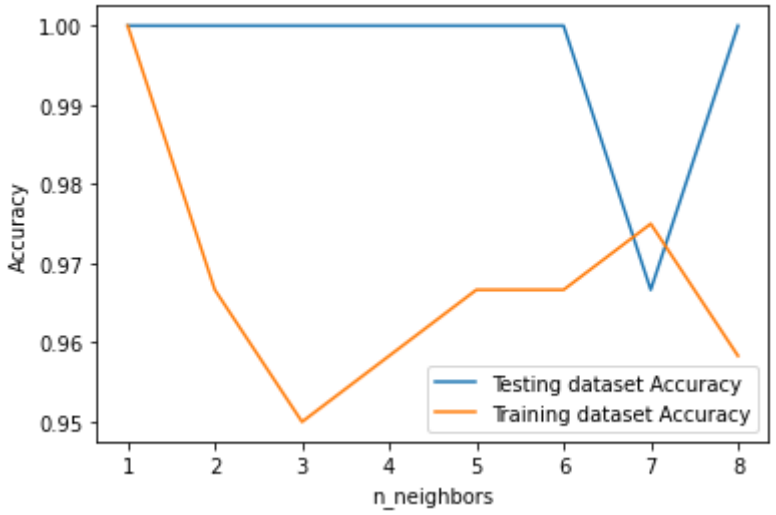
Una vez escogido el conjunto de datos, se debe separar primero los valores objetivo para trabajarlos por separado. Siento estos los datos que se usarán para el modelo, se separa el set completo en un conjunto de entrenamiento, prueba y validación. Los resultados son mejores cuando se hace esta división, pues el modelo entrena con ciertos datos, pero se valida el desempeño de este con los resultados que obtenga del set de prueba. De esta manera es posible conocer si el modelo funciona con daots nuevos y no se está acostumbrando a únicamente trabajar con los datos de entrenamiento. En ocasiones también se puede agregar la división de validación. Esto para que en el momento en que está entrenando el modelo, al mismo tiempo vaya validando los resultados y mejorándolos para que al final se obtenga el desempeño definitivo con el conjunto de prueba. Un método utilizado es el de Cross-Validation, el cual consiste en hacer varias iteraciones del entrenamiento del modelo con la diferencia de que cada iteración separa de manera distinta el conjunto de datos para entrenar y probar. Al final de cada iteración se evalúa el desempeño del modelo, y todas las califcaciones que obtuvo el modelo se promedian. Si estas varían cerca de los mismo valores, es una indicación de que el modelo está logrando describir de manera correcta el comportamiento de los datos. Si los valores son muy distintos, lo más probable es que exista un overfitting en el entrenamiento del modelo.

Otra manera que existe de separar el conjunto de datos es con Stratified K-Fold. Esta técnica se encarga de mantener una separación balanceada entre las clases pues en ocasiones tenemos más datos con cierta clase que otra, lo que podría afectar al modelo. En esta ocasión no se necesita usar esta ténica pues los datos están equilibrados, se tienen 50 respuestas de cada clase.

Al finalizar el entrenamiento del modelo, se debe realizar una evaluación de su desempeño. Esto se lleva a cabo a través de las conocidas métricas de evaluación. Existen distintas métricas que dependiendo del modelo que se utilice y la meta que se desee alcanzar es la que se debe escoger. En este caso como se trata de un clasificador, las métricas de accuracy y las de matriz de confusión para conocer si la manera en que clasificó fue la correcta.

Lo que se busca es que las métricas de evaluación sean buenas para asegurar que tu modelo está llevando a cabo un buen trabajo. Para esto se puede ir modificando los parámetros del modelo, así este se irá acomodando al comportamiento de los datos. Para K-Nearest Neighbors se puede modificar el número de ‘vecinos’ que se desea tener. Para

escoger esto se hizo una gráfica comparando el valor de accuracy dependiendo de los vecinos que se eligen.



Como se puede observar, el valor de accuracy se ve afectado por el número de vecinos. En este caso se puede elegir un valor de 5 o 6 pues en el dataset de entrenamiento tiene un buen accuracy, y en el de prueba se mantiene.

Para uno de los intentos se modificó que el número de vecinos fuera 5 y se realizó un cross-validation con 5 splits también. Los resultados fueron estos:

```
Metrica del modelo 0.9666666666666667
Metricas de Accuracy cross_validation [0.95833333 0.95833333 0.95833333 0.91666667 1.        ]
Media de Accuracy cross_validation 0.9583333333333333

Metrica en Test 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      1.00      1.00         9
           2       1.00      1.00      1.00        11

    accuracy                           1.00        30
   macro avg       1.00      1.00      1.00        30
weighted avg       1.00      1.00      1.00        30
```

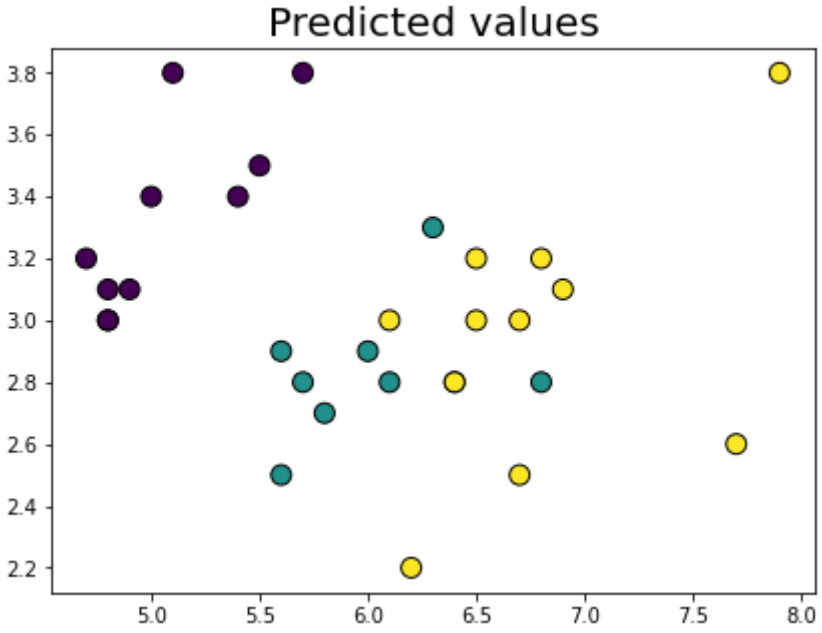
Las métricas de evaluación parecen estar muy bien, sin embargo las métricas en el dataset de prueba son perfectas, lo que puede levantar una alerta de que el modelo está haciendo overfitting. Esto puede estar pasando porque el tamaño del dataset no es muy grande por lo que con el cross-validation está tomando información de pocos datos.

En otro intento se eliminó el cross-validation y se modificó el número de vecinos a 7 y los resultados ya estuvieron más de acuerdo a las métricas que se obtuvo con el set de entrenamiento.

```
Metrica del modelo 0.975

Metrica en Test 0.9666666666666667
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        10
           1       1.00      0.89      0.94         9
           2       0.92      1.00      0.96        11

    accuracy                           0.97        30
   macro avg       0.97      0.96      0.97        30
weighted avg       0.97      0.97      0.97        30
```

Ahora el set de entrenamiento y en el de prueba tienen métricas parecidas, así que hay más razones para confiar que si se prueba el modelo en datos nuevos que jamás ha visto, los resultados de la clasificación serán buenos.

Los resultados de clasificación fueron los siguientes:



Para la clasificación de color amarillo y azul sí hay una separación, pero en el momento donde se juntan es dificil distinguir cuáles pertenecen a cada categoría. Pero el modelo logró hacerlo de una manera buena como se pudo ver en las métricas de evaluación.

La liga al código en Colaboratory se puede encontrar en el siguiente link: https://colab.research.google.com/drive/1xjMicJqIgKiGEfKbhhMnFJ5GUwy3ZLsH?usp=sharing