



# Tecnológico de Monterrey

Instituto Tecnológico de Estudios Superiores de  
Monterrey Campus Estado de México

Inteligencia artificial avanzada para la ciencia de datos  
I (Grupo 101)

Análisis y Reporte sobre el desempeño del modelo

Melissa Aurora Fadanelli Ordaz

A01749483

Profesor Jorge Adolfo Ramírez Uresti

## 1. Implementación de *dataset* de Iris

Para la implementación y el análisis se utilizó el *dataset* de iris que tiene un buen tamaño ya que tiene suficientes datos para el algoritmo de clasificación elegido (*Knn*) pero no tiene demasiados datos por lo que es bastante manejable. Tiene clases bien definidas y está disponible de manera fácil. Por lo que considero que es una base de datos ideal para el aprendizaje y las demostraciones necesarias.

## 2. Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (*Train/Test/Validation*).

Los datos se dividen en tres conjuntos, el de entrenamiento, el de prueba y el de validación.

Los datos de entrenamiento se utilizan para ajustar los parámetros.

Los datos de validación se utilizan para evaluar el ajuste del modelo.

Los datos de prueba se utilizan para evaluar el rendimiento final del modelo.

Dentro del código esto se puede ver con las siguientes líneas:

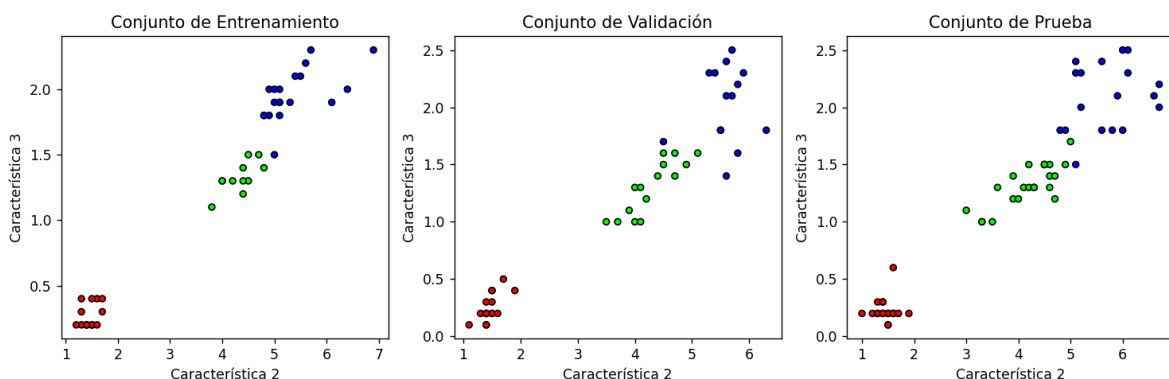
```
# División del conjunto de datos en un conjunto temporal y prueba
X_temp, X_test, y_temp, y_test = train_test_split(X, y, test_size=0.4, random_state=1234)

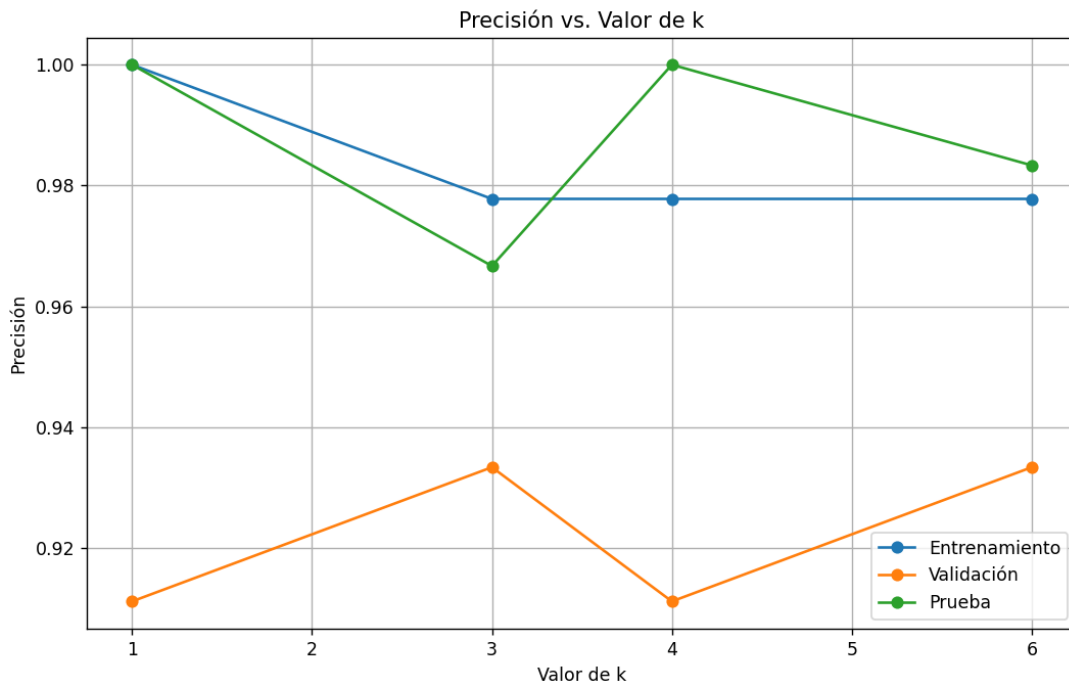
# Usando el conjunto temporal podemos dividirlo en dos para el entrenamiento y la validación
X_train, X_validation, y_train, y_validation = train_test_split(X_temp, y_temp, test_size=0.5, random_state=1234)
```

Aquí se divide primeramente en los datos de prueba y otro conjunto temporal que se será dividido nuevamente para poder sacar el conjunto de entrenamiento y el de validación.

Con los tres conjuntos podemos realizar el entrenamiento y la validación, para finalmente utilizar los datos de prueba. En la siguiente imagen se puede apreciar el uso de los conjuntos con los métodos de evaluación del modelo.

## 3. Gráficas comparativas





#### 4. Diagnóstico y explicación el grado de *bias* o sesgo: bajo medio alto

Considerando que las métricas de la primera evaluación, con valor de  $k=5$ , podemos definir que el modelo tiene un sesgo bajo, ya que la matriz contiene pocos errores, conforme el valor de  $k$  va aumentando se pueden notar más errores, en  $k=12$  podríamos considerar que el sesgo es medio, ya que la brecha aumenta y se muestran algunos errores, sin embargo en las últimas dos evaluaciones, donde  $k=26$  y  $k=35$ , la precisión disminuye altamente y la matriz de confusión aumenta en número de errores, por lo que el sesgo es alto.

Podemos notar que conforme aumenta el valor de  $k$  el sesgo va en aumento.

```

Usando valor de k=5
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 0, 0, 0, 2, 1, 0, 1, 1, 1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0, 2, 0, 0, 2]
0.9333333333333333
Matriz de confusión con datos de validación:
[[16 0 0]
 [ 0 13 2]
 [ 0 1 13]]
Puntaje F1 con datos de validación: 0.9333333333333333
Predicciones con datos de prueba:
[1, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 1, 1, 2, 0, 2, 1, 2, 0, 1, 2, 0, 2, 2, 0, 0, 0, 1, 0, 1, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 1, 1, 1, 1, 1, 0]
0.9833333333333333
Matriz de confusión on datos de prueba:
[[19 0 0]
 [ 0 22 1]
 [ 0 0 18]]
Puntaje F1 on datos de prueba: 0.9833333333333334

Usando valor de k=12
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 2, 2, 2, 2, 2, 1, 2, 2, 1, 1, 1, 0, 0, 0, 2, 2, 0, 1, 1, 1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0, 2, 0, 0, 2]
0.8888888888888888
Matriz de confusión con datos de validación:
[[16 0 0]
 [ 0 11 4]
 [ 0 1 13]]
Puntaje F1 con datos de validación: 0.888923934687376
Predicciones con datos de prueba:
[1, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 1, 1, 1, 2, 0, 2, 1, 2, 0, 1, 2, 0, 2, 2, 0, 0, 0, 1, 0, 1, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 1, 1, 1, 1, 2, 1, 0]
0.9666666666666667
Matriz de confusión on datos de prueba:
[[19 0 0]
 [ 0 21 2]
 [ 0 0 18]]
Puntaje F1 on datos de prueba: 0.9667862838915471

```

```

Usando valor de k=26
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 0, 2, 2, 1, 0, 2, 0, 0, 2, 2, 0, 2, 0, 2, 0, 2, 0, 0, 2]
0.7111111111111111
Matriz de Confusión con datos de validación:
[[16  0  0]
 [ 0  2 13]
 [ 0  0 14]]
Puntaje F1 con datos de validación: 0.6464530527658219
Predicciones con datos de prueba:
[2, 2, 2, 0, 2, 0, 0, 0, 2, 2, 2, 0, 2, 2, 0, 1, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2, 0, 0, 2, 0, 2, 2, 0, 0, 0, 0, 1, 0, 2, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 1, 1, 2, 2, 2, 0]
0.6833333333333333
Matriz de Confusión on datos de prueba:
[[19  0  0]
 [ 1  4 18]
 [ 0  0 18]]
Puntaje F1 on datos de prueba: 0.6221272554605888

Usando valor de k=35
Predicciones con datos de validación:
[2, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 0, 2, 2, 1, 0, 2, 0, 0, 2, 2, 2, 0, 2, 0, 2, 2, 0, 2, 0, 0, 2]
0.6888888888888889
Matriz de Confusión con datos de validación:
[[16  0  0]
 [ 0  1 14]
 [ 0  0 14]]
Puntaje F1 con datos de validación: 0.6046296296296296
Predicciones con datos de prueba:
[2, 2, 2, 0, 2, 0, 0, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 2, 2, 0, 2, 2, 0, 0, 2, 0, 2, 2, 0, 0, 0, 0, 0, 0, 2, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 0, 2, 2, 2, 2, 0]
0.6166666666666667
Matriz de Confusión on datos de prueba:
[[19  0  0]
 [ 3  0 20]
 [ 0  0 18]]
Puntaje F1 on datos de prueba: 0.48635307781649245

```

## 5. Diagnóstico y explicación el grado de varianza: bajo medio alto

Considerando los resultados mostrados en el anterior punto, podemos notar que en el valor de  $k=5$  la varianza no es significativa, los valores son consistentes. Viendo los valores de  $k=12$  podemos notas que la varianza aumenta, ya que el rendimiento en validación y prueba cambia con el valor de  $k$ . En valores de  $k=26$  y  $k=35$  la varianza es muy alta y de acuerdo con el *dataset* que se utilice puede cambiar el rendimiento.

## 6. Diagnóstico y explicación el nivel de ajuste del modelo: *underfitt fitt overfitt*

Para el diagnóstico del ajuste notamos que en  $k=5$ , siendo consistente también con los puntos anteriores, el rendimiento es buen en los conjuntos de validación y prueba, sin embargo conforme va aumentando el valor de  $k$ , podemos notar una tendencia hacia la eficiencia del modelo, en los valores más altos de  $k$  (26 y 35) podemos ver que no se generaliza de forma adecuada, por lo que los ajustes adecuados para hacer sería encontrar el punto de los valores de  $k$  en los que se equilibre el sesgo con la varianza

## 7. Mejorar el desempeño del modelo

Para este punto y de acuerdo con los diagnósticos que se hicieron podemos empezar notando que se necesita un ajuste en los valores de  $k$ , siendo los valores utilizados en la primera parte  $k= [5,12,26,35]$  y resultando en el valor de 5 siendo el mejor dentro de los aspectos de la evaluación del rendimiento, usaremos para una nueva evaluación los

valores de  $k = [1, 3, 4, 6]$ . Dejaremos de lado el cinco ya que lo utilizamos anteriormente y mostró resultados adecuados.

Utilizaremos el ajuste de hiperparámetros incluido en la librería *scikit-learn* para poder encontrar de forma más eficiente los valores de  $k$ , así podemos hacerlo de forma manual (tanteando los valores de  $k$ ) y de forma automática (con la librería).

Para nuestros valores manuales los resultados muestran mejor rendimiento que cuando aumenta el valor de  $k$ , viendo estos resultados considero que el mejor valor de  $k$  para usar es  $k=4$ .

```
Usando valor de k=1
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 0, 0, 0, 2, 2, 0, 1, 1, 1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0, 2, 0, 0, 2]
0.9111111111111111
Matriz de Confusión con datos de validación:
[[16  0  0]
 [ 0 12  3]
 [ 0  1 13]]
Puntaje F1 con datos de validación: 0.9108994708994709
Predicciones con datos de prueba:
[1, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 1, 1, 2, 0, 2, 1, 2, 0, 1, 2, 0, 2, 1, 0, 0, 0, 0, 1, 0, 1, 0, 2, 2, 0, 2, 2, 2, 2, 0, 2, 2, 1, 1, 1, 1, 1, 1, 0]
1.0
Matriz de Confusión on datos de prueba:
[[19  0  0]
 [ 0 23  0]
 [ 0  0 18]]
Puntaje F1 on datos de prueba: 1.0

Usando valor de k=3
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 0, 0, 0, 2, 1, 0, 1, 1, 1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0, 2, 0, 0, 2]
0.9333333333333333
Matriz de Confusión con datos de validación:
[[16  0  0]
 [ 0 13  2]
 [ 0  1 13]]
Puntaje F1 con datos de validación: 0.9333333333333333
Predicciones con datos de prueba:
[1, 2, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 1, 1, 1, 2, 0, 2, 1, 2, 0, 1, 2, 0, 2, 1, 0, 0, 0, 0, 1, 0, 2, 0, 2, 2, 0, 2, 2, 2, 2, 0, 2, 2, 1, 1, 1, 1, 1, 1, 0]
0.9666666666666667
Matriz de Confusión on datos de prueba:
[[19  0  0]
 [ 0 21  2]
 [ 0  0 18]]
Puntaje F1 on datos de prueba: 0.9667862838915471

Usando valor de k=4
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 2, 2, 1, 2, 2, 1, 2, 2, 1, 1, 1, 0, 0, 0, 2, 2, 0, 1, 1, 1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0, 2, 0, 0, 2]
0.9111111111111111
Matriz de Confusión con datos de validación:
[[16  0  0]
 [ 0 12  3]
 [ 0  1 13]]
Puntaje F1 con datos de validación: 0.9108994708994709
Predicciones con datos de prueba:
[1, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 1, 1, 1, 2, 0, 2, 1, 2, 0, 1, 2, 0, 2, 1, 0, 0, 0, 0, 1, 0, 1, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 1, 1, 1, 1, 1, 1, 0]
1.0
Matriz de Confusión on datos de prueba:
[[19  0  0]
 [ 0 23  0]
 [ 0  0 18]]
Puntaje F1 on datos de prueba: 1.0

Usando valor de k=6
Predicciones con datos de validación:
[2, 1, 0, 2, 2, 0, 0, 0, 2, 2, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0, 2, 1, 0, 1, 1, 1, 0, 1, 0, 0, 2, 2, 2, 0, 1, 0, 1, 1, 0, 2, 0, 0, 2]
0.9333333333333333
Matriz de Confusión con datos de validación:
[[16  0  0]
 [ 0 13  2]
 [ 0  1 13]]
Puntaje F1 con datos de validación: 0.9333333333333333
Predicciones con datos de prueba:
[1, 1, 2, 0, 1, 0, 0, 0, 1, 2, 1, 0, 2, 1, 0, 1, 2, 0, 2, 1, 1, 1, 1, 1, 2, 0, 2, 1, 2, 0, 1, 2, 0, 2, 2, 0, 0, 0, 0, 1, 0, 1, 0, 2, 2, 0, 2, 2, 2, 0, 2, 2, 1, 1, 1, 1, 1, 1, 0]
0.9833333333333333
Matriz de Confusión on datos de prueba:
[[19  0  0]
 [ 0 22  1]
 [ 0  0 18]]
Puntaje F1 on datos de prueba: 0.9833733733733734
```

Para el ajuste utilizando la librería, los resultados son los siguientes:

```
Precisiones antes de la búsqueda de hiperparámetros:  
Validación con características seleccionadas: 0.3111111111111111  
  
Precisiones después de la búsqueda de hiperparámetros:  
Validación con k óptimo (3): 0.9555555555555556
```

Se utilizó también la técnica de normalización, demostrando como cambia y la importancia de normalizar los datos que llegan a tener escalas diferentes:

```
Precisiones antes de la normalización:  
Entrenamiento: 0.9777777777777777  
Validación: 0.9333333333333333  
Prueba: 0.9833333333333333  
  
Precisiones después de la normalización:  
Entrenamiento: 0.9777777777777777  
Validación: 0.9111111111111111  
Prueba: 0.9833333333333333
```

Y también se utilizó la técnica de selección de características, que nos ayuda a reducir las características irrelevantes dentro del conjunto de datos:

```
Precisiones antes de la selección de características:  
Validación: 0.9111111111111111  
  
Precisiones después de la selección de características:  
Validación con características seleccionadas: 0.9333333333333333
```

Podemos considerar entonces que los valores altos de k van empeorando el rendimiento del modelo, se aumenta el sesgo y la varianza, aumentando el valor de vecinos es también menos capaz de generalizar y de discriminar entre clases, provocando más errores en la clasificación.

La técnicas utilizadas nos demuestran la importancia de mejorar el conjunto de datos, o de preprocesarlo antes de realizar predicciones, pues estas predicciones pueden verse afectada con los datos crudos o sin preprocesar.