Tarea evaluable CE_5073 3.1

Programación de aprendizaje automático

David Ramírez Ruiz
1/12/24

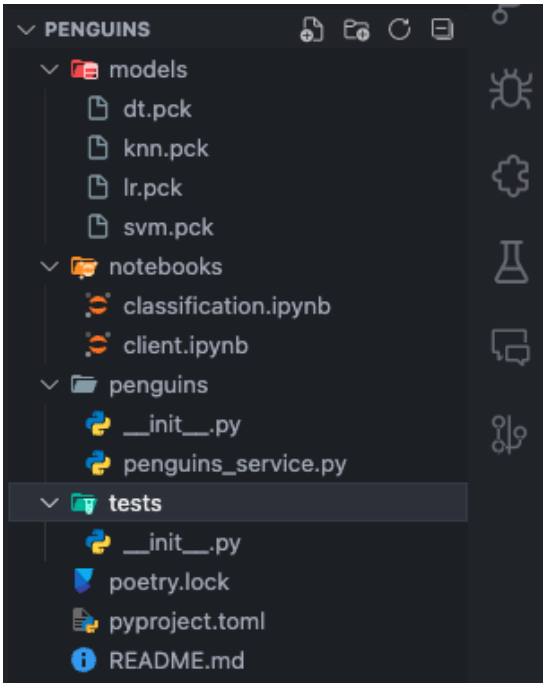
Índice

1 - REPOSITORIO GITHUB

Aquí adjunto el link para acceder a mi repositorio en GitHub. Se trata de un repositorio en el que estoy subiendo todos los proyectos del curso, por esa razón el link redirige directamente a la carpeta “penguins” que contiene todo el proyecto de esta tarea.

2 - PROYECTO

1 - Estructura de carpetas



2 - Archivo pyproject.toml

3 - Resultados y comentarios

Breve explicación sobre los datos enviados a los distintos modelos:

- **Datos A**: He creado unos datos ficticios basados en los valores en general que he podido encontrar en la tabla original. Aunque son inventados, he intentado que se mantengan dentro de los rangos más comunes, con la idea de comprobar cómo se comportan los modelos para casos genéricos.

- **Datos B:** Para este caso, he decidido experimentar un poco más utilizando los datos de un pingüino identificado en la tabla original:
  - **Island:** Biscoe
  - **Bill length (mm):** 50.4
  - **Bill depth (mm):** 15.7
  - **Flipper length (mm):** 222.0
  - **Body mass (g):** 5750.0
  - **Sex:** Male

Pero en el caso de las peticiones he modificado únicamente el campo "island" para observar cómo afecta este cambio al resultado.

- Datos B

```
!curl --request POST "http://127.0.0.1:8000/predict_lr" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Torgersen\",\
    \"bill_length_mm\": 50.4,\
    \"bill_depth_mm\": 15.7,\
    \"flipper_length_mm\": 222.0,\
    \"body_mass_g\": 5750.0,\
    \"sex\": \"Male\"\
}"
```

✓ 0.2s Python

```
{
  "penguin": "Gentoo",
  "probability": 0.5002252704381351
}
```

El modelo devuelve una probabilidad del 50%, parece que le ha costado clasificarlo en una especie de pingüino, posiblemente debido a su manera de clasificar. A través de la comparación con los otros modelos comprobaremos la fiabilidad que ofrecen.

SVM

- Datos A

```
!curl --request POST "http://127.0.0.1:8000/predict_svm" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Biscoe\",\
    \"bill_length_mm\": 33.0,\
    \"bill_depth_mm\": 18.0,\
    \"flipper_length_mm\": 140.0,\
    \"body_mass_g\": 3800.0,\
    \"sex\": \"Female\"\
}"
```

✓ 0.2s Python

```
{
  "penguin": "Adelie",
  "probability": 0.999997212470352
}
```

Siguiendo el ejemplo de la regresión logística aquí también conseguimos un resultado bastante claro con una probabilidad cercana a 1. Podemos prácticamente asegurar que el modelo está realizando una buena clasificación de este pingüino como Adelie.

- Datos B

```
!curl --request POST "http://127.0.0.1:8000/predict_svm" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Torgersen\",\
    \"bill_length_mm\": 50.4,\
    \"bill_depth_mm\": 15.7,\
    \"flipper_length_mm\": 222.0,\
    \"body_mass_g\": 5750.0,\
    \"sex\": \"Male\"\
}"
```

5] ✓ 0.2s Python


```
{
  "penguin": "Gentoo",
  "probability": 0.9900303720058639
}
```


El modelo SVM clasifica el pingüino en la misma especie que el modelo anterior y además con una probabilidad del 99%. Esto podría indicar que el modelo SVM está mejor adaptado a las características de los datos en este caso. Sin embargo sería importante analizar cómo se comportan los otros modelos con estos datos.

Decision tree

- Datos A

```
!curl --request POST "http://127.0.0.1:8000/predict_dt" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Biscoe\",\
    \"bill_length_mm\": 33.0,\
    \"bill_depth_mm\": 18.0,\
    \"flipper_length_mm\": 140.0,\
    \"body_mass_g\": 3800.0,\
    \"sex\": \"Female\"\
}"
```

6] ✓ 0.2s Python


```
{
  "penguin": "Adelie",
  "probability": 1.0
}
```


Una vez más, el pingüino de los datos A se clasifica como 'Adelie'. Además, obtenemos una probabilidad del 100% , lo que nos muestra la precisión de los modelos decision tree. Estos modelos dividen los datos en nodos basados en las características, y esto puede dar resultados muy precisos, incluso demasiado. Esta alta precisión puede causar un caso de overfitting si el modelo se ajusta demasiado a los datos de entrenamiento. Es difícil asegurarlo simplemente con dos ejemplos pero me parece una reflexión a tener en cuenta.

- Datos B

```
!curl --request POST "http://127.0.0.1:8000/predict_dt" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Torgersen\",\
    \"bill_length_mm\": 50.4,\
    \"bill_depth_mm\": 15.7,\
    \"flipper_length_mm\": 222.0,\
    \"body_mass_g\": 5750.0,\
    \"sex\": \"Male\"\
}"
```

7] ✓ 0.2s Python


```
{
  "penguin": "Chinstrap",
  "probability": 1.0
}
```


En el caso de los datos B el modelo decision tree nos retorna un resultado distinto a los dos anteriores. Su probabilidad es del 100%, pero como he comentado en el ejemplo de los datos A, los árboles de decisión pueden estar sobre ajustados en algunos casos por la manera de su funcionamiento. Por lo que no es recomendable confiar en este único resultado sin tener en cuenta los demás.

K-Nearest Neighbours (KNN)

- Datos A

```
!curl --request POST "http://127.0.0.1:8000/predict_knn" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Biscoe\",\
    \"bill_length_mm\": 33.0,\
    \"bill_depth_mm\": 18.0,\
    \"flipper_length_mm\": 140.0,\
    \"body_mass_g\": 3800.0,\
    \"sex\": \"Female\"\
}"
```

8] ✓ 0.2s Python


```
{
  "penguin": "Adelie",
  "probability": 1.0
}
```


Confirmamos con el cuarto modelo que la clasifiación de este pingüino altamente correcta.

- Datos B

```
!curl --request POST "http://127.0.0.1:8000/predict_knn" \
--header "Content-Type: application/json" \
--data-raw "{\
    \"island\": \"Torgersen\",\
    \"bill_length_mm\": 50.4,\
    \"bill_depth_mm\": 15.7,\
    \"flipper_length_mm\": 222.0,\
    \"body_mass_g\": 5750.0,\
    \"sex\": \"Male\"\
}"
```

✓ 0.2s Python

```
{
  "penguin": "Gentoo",
  "probability": 1.0
}
```

Por último conseguimos de nuevo el mismo resultado que los modelos regresión logística y SVM. Este resultado es interesante porque, a diferencia de los modelos anteriores, KNN clasifica al pingüino en función de los vecinos más cercanos. Este modelo se basa en la mayoría de las etiquetas de esos vecinos. Sin embargo, puede verse afectado por cómo están distribuidos los datos y por el número de vecinos (K), que puede influir en su precisión.