UNIVERSIDAD DE CÓRDOBA

Departamento de Informática y Análisis Numérico

Ingeniería del Software, Conocimiento y Bases de Datos

MASTERES UNIVERSITARIOS-UNIVERSIDAD DE CÓRDOBA

ASIGNATURA:

Análisis, Diseño y Procesamiento de Datos Aplicados a las Ciencias y a las Tecnologías

Nombre : Lebbihi

Apellido : Mohammed

Profesores : 1- Gonzalo Cerreuila Garcia

2- Domingo Ortiz Boyer

3- Juan Antonio Romero

PRÁCTICAS (APARTADO III)

Etapa 1: Diseño de la Estructura de Datos

Objetivo: Definir cómo se almacenarán y organizarán los datos dentro de MongoDB.

Acciones:

Crear una colección departments para los departamentos, con campos para el número del departamento, nombre y localización.

Crear una colección employees para los empleados, con campos para el número de empleado, nombre, puesto, fecha de contratación, salario, comisión y el número del departamento.

Considerar las relaciones entre los departamentos y los empleados para determinar si se necesitan referencias o documentos embebidos.

Etapa 2: Implementación de la Estructura de Datos

Objetivo: Establecer las colecciones y sus relaciones en la base de datos MongoDB.

Acciones:

Usar comandos de MongoDB para crear las colecciones departments y employees.

Definir los índices necesarios para optimizar las consultas, especialmente para campos que se usarán frecuentemente en búsquedas y relaciones.

Etapa 3: Inserción de Datos

Objetivo: Poblar las colecciones con datos de prueba que sean representativos de los datos reales.

Acciones:

Generar o insertar datos de prueba en la colección departments.

Generar o insertar datos de prueba en la colección employees, asegurándose de que los datos reflejen las relaciones adecuadas con la colección departments.

Etapa 5: Optimización y Pruebas

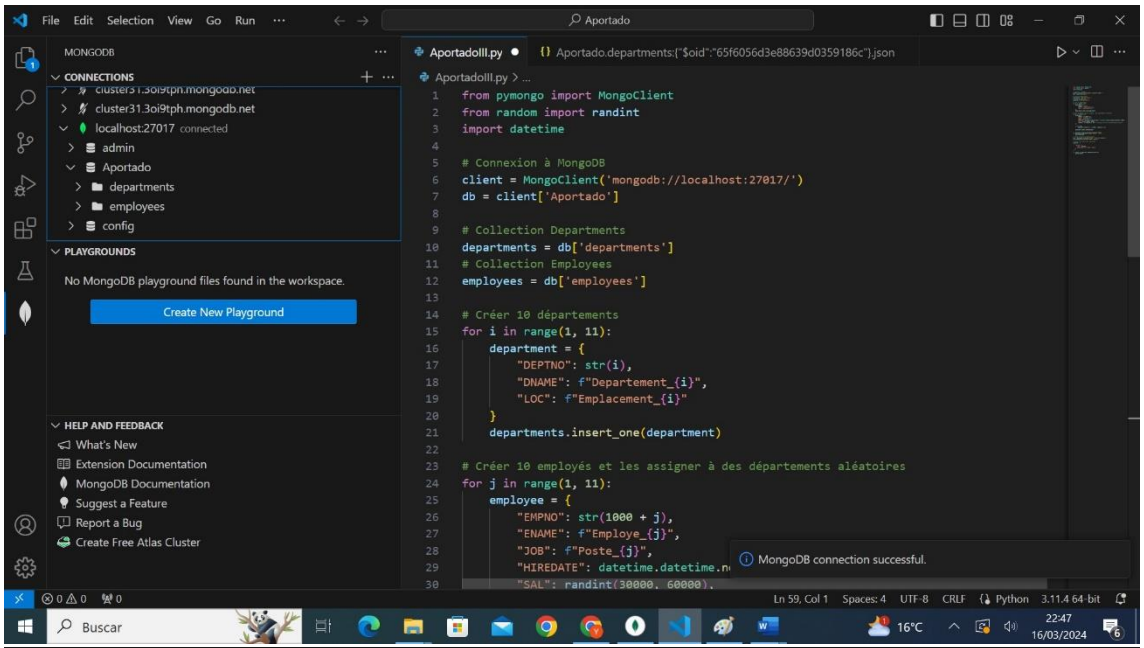
Objetivo: Asegurarse de que la solución sea eficiente y se ajuste a las necesidades de la aplicación.

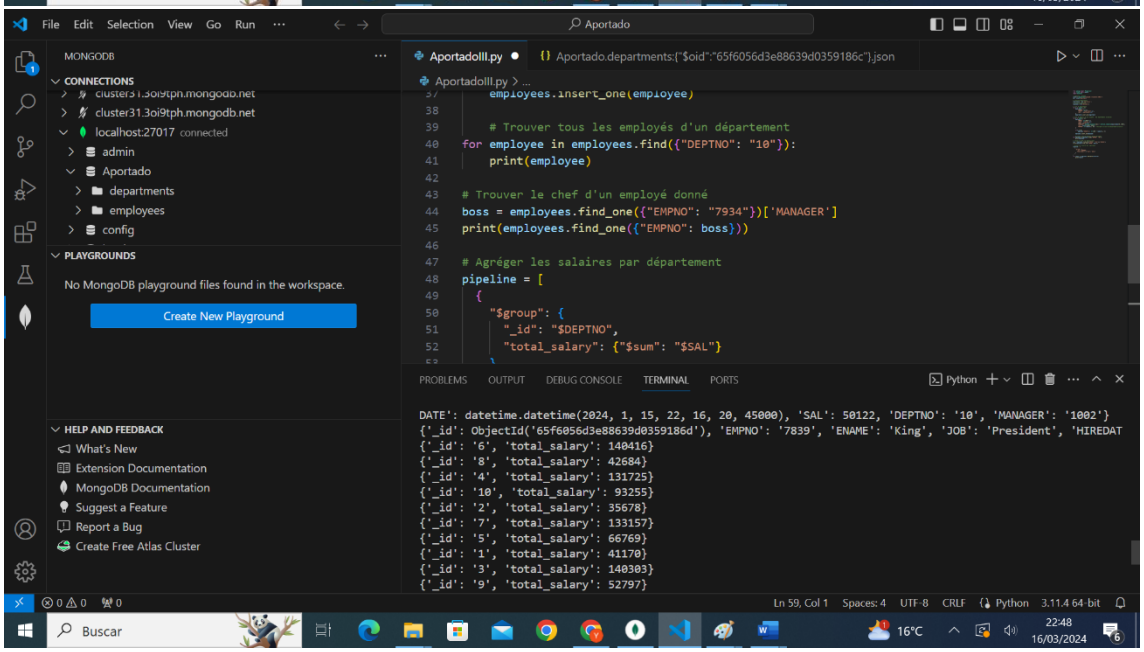
Acciones:

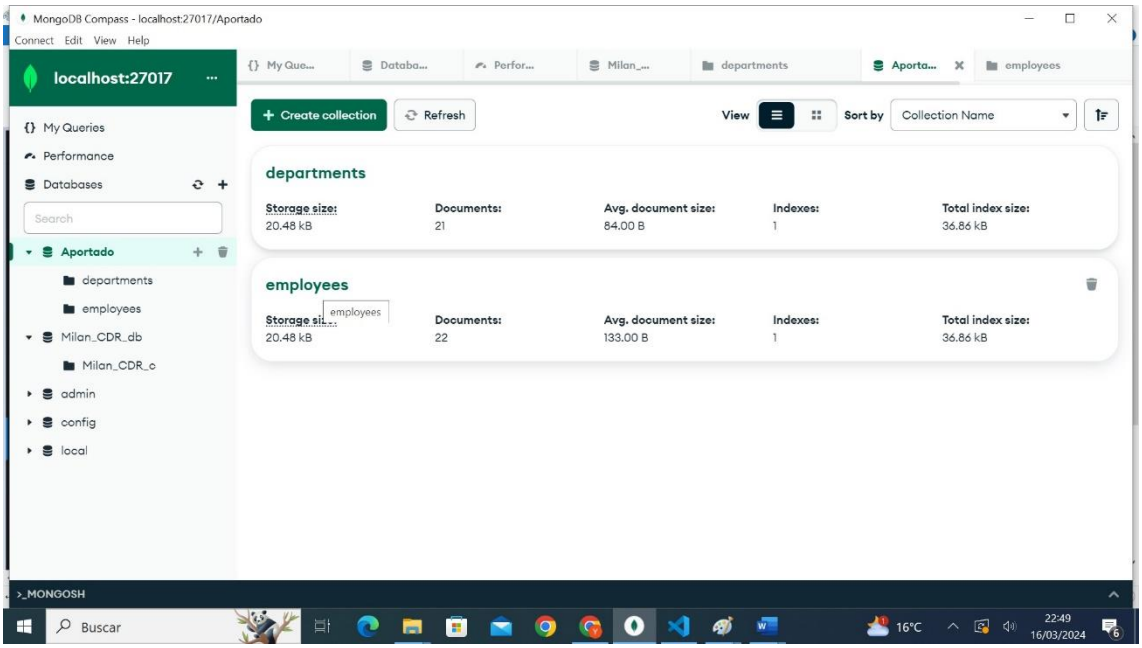
Analizar el rendimiento de las consultas y realizar ajustes en los índices si es necesario.

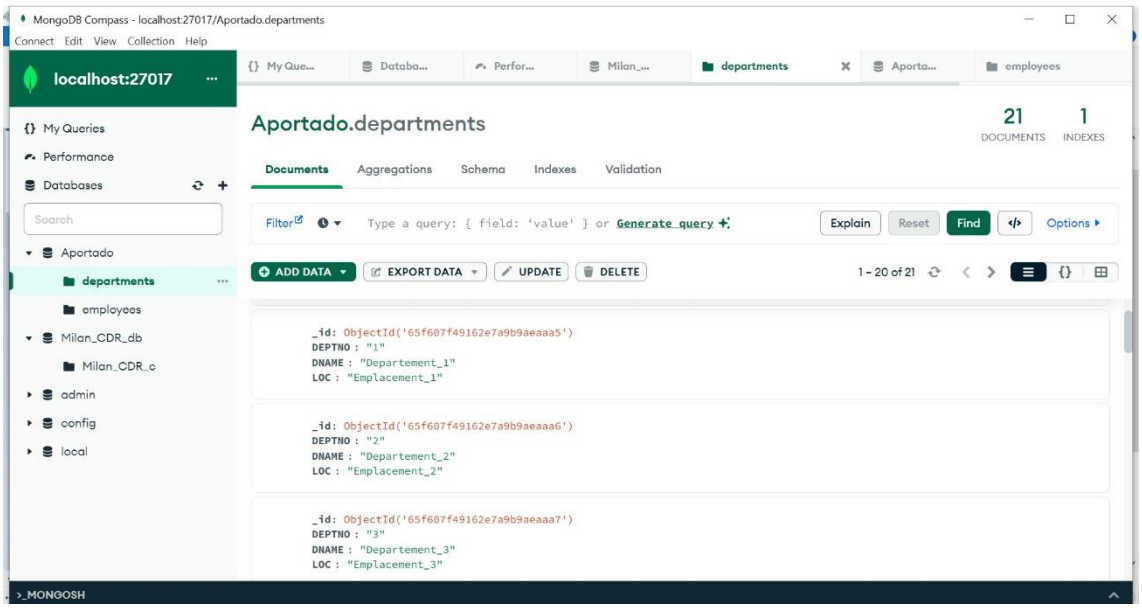
Revisar el esquema de datos y realizar cualquier ajuste para mejorar la estructura basándose en los patrones de acceso a los datos.

Estas etapas cubren el proceso de diseño e implementación de una solución de base de datos usando MongoDB para la parte práctica de un curso, con un enfoque en el aprendizaje teórico y la aplicación práctica de estos conocimientos.


File Edit Selection View Go Run
Aportado
MONGODB
CONNECTIONS
cluster31.3oi9tph.mongodb.net
localhost:27017 connected
admin
Aportado
departments
employees
config
PLAYGROUNDS
No MongoDB playground files found in the workspace.
Create New Playground
HELP AND FEEDBACK
What's New
Extension Documentation
MongoDB Documentation
Suggest a Feature
Report a Bug
Create Free Atlas Cluster
AportadoIII.py
Aportado.departments:{"$oid":"65f6056d3e88639d0359186c"}.json
from pymongo import MongoClient
from random import randint
import datetime
# Connexion à MongoDB
client = MongoClient('mongodb://localhost:27017/')
db = client['Aportado']
# Collection Departments
departments = db['departments']
# Collection Employees
employees = db['employees']
# Créer 10 départements
for i in range(1, 11):
department = {
"DEPTNO": str(i),
"DNAME": f"Departement_{i}",
"LOC": f"Emplacement_{i}"
}
departments.insert_one(department)
# Créer 10 employés et les assigner à des départements aléatoires
for j in range(1, 11):
employee = {
"EMPNO": str(1000 + j),
"ENAME": f"Employe_{j}",
"JOB": f"Poste_{j}",
"HIREDATE": datetime.datetime.n
"SAL": randint(30000, 60000),
MongoDB connection successful.
Ln 59, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit
Buscar
16°C
22:47
16/03/2024


File Edit Selection View Go Run
Aportado
MONGODB
CONNECTIONS
cluster31.3oi9tph.mongodb.net
localhost:27017 connected
admin
Aportado
departments
employees
config
PLAYGROUNDS
No MongoDB playground files found in the workspace.
Create New Playground
HELP AND FEEDBACK
What's New
Extension Documentation
MongoDB Documentation
Suggest a Feature
Report a Bug
Create Free Atlas Cluster
AportadoIII.py
Aportado.departments:{"$oid":"65f6056d3e88639d0359186c"}.json
employees.insert_one(employee)
# Trouver tous les employés d'un département
for employee in employees.find({"DEPTNO": "10"}):
print(employee)
# Trouver le chef d'un employé donné
boss = employees.find_one({"EMPNO": "7934"})['MANAGER']
print(employees.find_one({"EMPNO": boss}))
# Agréger les salaires par département
pipeline = [
{
"$group": {
"_id": "$DEPTNO",
"total_salary": {"$sum": "$SAL"}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
DATE': datetime.datetime(2024, 1, 15, 22, 16, 20, 45000), 'SAL': 50122, 'DEPTNO': '10', 'MANAGER': '1002'}
{'_id': ObjectId('65f6056d3e88639d0359186d'), 'EMPNO': '7839', 'ENAME': 'King', 'JOB': 'President', 'HIREDAT
{'_id': '6', 'total_salary': 140416}
{'_id': '8', 'total_salary': 42684}
{'_id': '4', 'total_salary': 131725}
{'_id': '10', 'total_salary': 93255}
{'_id': '2', 'total_salary': 35678}
{'_id': '7', 'total_salary': 133157}
{'_id': '5', 'total_salary': 66769}
{'_id': '1', 'total_salary': 41170}
{'_id': '3', 'total_salary': 140303}
{'_id': '9', 'total_salary': 52797}
Ln 59, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit
Buscar
16°C
22:48
16/03/2024


MongoDB Compass - localhost:27017/Aportado
Connect Edit View Help
localhost:27017
My Queries
Performance
Databases
Search
Aportado
departments
employees
Milan_CDR_db
Milan_CDR_c
admin
config
local
Create collection
Refresh
View
Sort by
Collection Name
departments
Storage size:
20.48 kB
Documents:
21
Avg. document size:
84.00 B
Indexes:
1
Total index size:
36.86 kB
employees
Storage size:
20.48 kB
Documents:
22
Avg. document size:
133.00 B
Indexes:
1
Total index size:
36.86 kB
>_MONGOSH
Buscar
16°C
22:49
16/03/2024


MongoDB Compass - localhost:27017/Aportado.departments
Connect Edit View Collection Help
localhost:27017
My Queries
Performance
Databases
Search
Aportado
departments
employees
Milan_CDR_db
Milan_CDR_c
admin
config
local
Aportado.departments
21
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' } or Generate query
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
UPDATE
DELETE
1 - 20 of 21
_id: ObjectId('65f607f49162e7a9b9aeaaa5')
DEPTNO : "1"
DNAME : "Departement_1"
LOC : "Emplacement_1"
_id: ObjectId('65f607f49162e7a9b9aeaaa6')
DEPTNO : "2"
DNAME : "Departement_2"
LOC : "Emplacement_2"
_id: ObjectId('65f607f49162e7a9b9aeaaa7')
DEPTNO : "3"
DNAME : "Departement_3"
LOC : "Emplacement_3"
>_MONGOSH

2: APLICACIONES CIENTÍFICAS Y EMPRESARIALES (I)

Etapa 1: Recopilación de Datos

Extraer los datos de compuestos de una base de datos, donde cada compuesto tiene un conjunto de características (fingerprint molecular) y una etiqueta que indica su actividad biológica (activa o inactiva).

Etapa 2: Limpieza y Preparación de Datos

Limpiar los datos para la modelización. Esto podría incluir la eliminación de columnas que sólo contienen ceros y la transformación de datos categóricos en numéricos si es necesario.

Etapa 3: División de Datos

Dividir el conjunto de datos en partes de entrenamiento y prueba para poder evaluar el modelo de manera justa.

Etapa 4: Desarrollo del Modelo

Desarrollar un modelo de clasificación que utilice las características de los compuestos para predecir su actividad biológica.

Etapa 5: Entrenamiento del Modelo

Entrenar el modelo con el conjunto de datos de entrenamiento, ajustando los parámetros según sea necesario para mejorar el rendimiento.

Etapa 6: Pruebas y Validación

Probar el modelo con el conjunto de datos de prueba para evaluar su capacidad de generalización y su rendimiento con datos no vistos durante el entrenamiento.

Etapa 7: Evaluación de Rendimiento

Evaluar el rendimiento del modelo utilizando métricas como la precisión, la sensibilidad, la especificidad y el área bajo la curva ROC (AUC-ROC).

Etapa 8: Ajuste Fino

Realizar ajustes finos en el modelo según sea necesario para mejorar el rendimiento, lo cual puede incluir la selección de características, la ingeniería de características o la optimización de hiperparámetros.

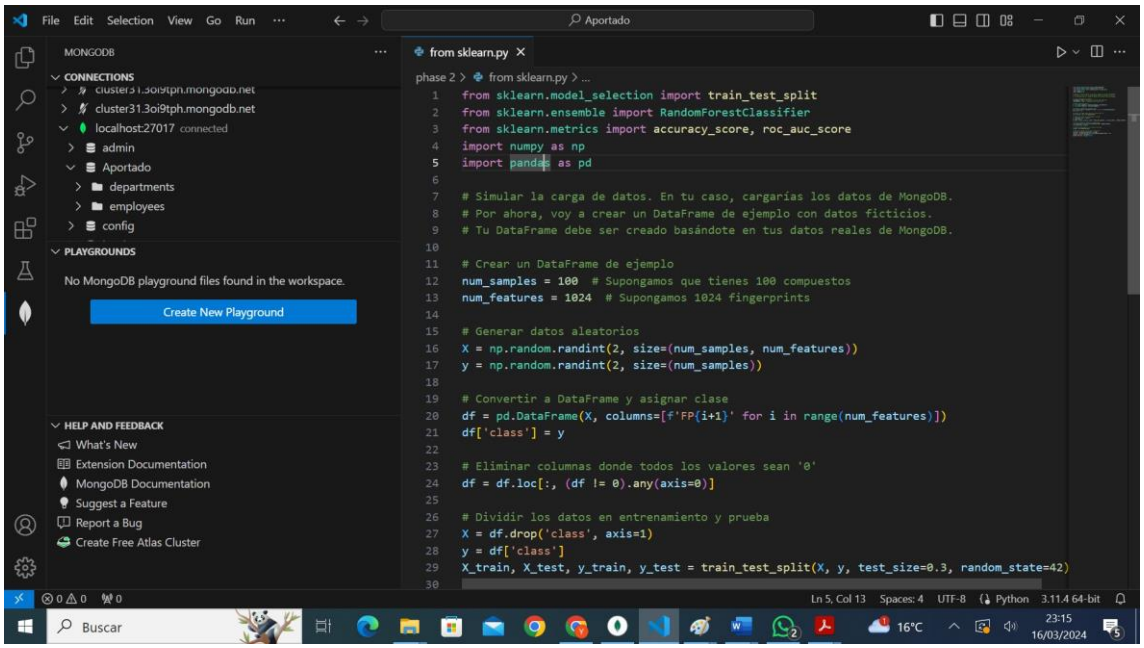
Etapa 9: Interpretación y Conclusión

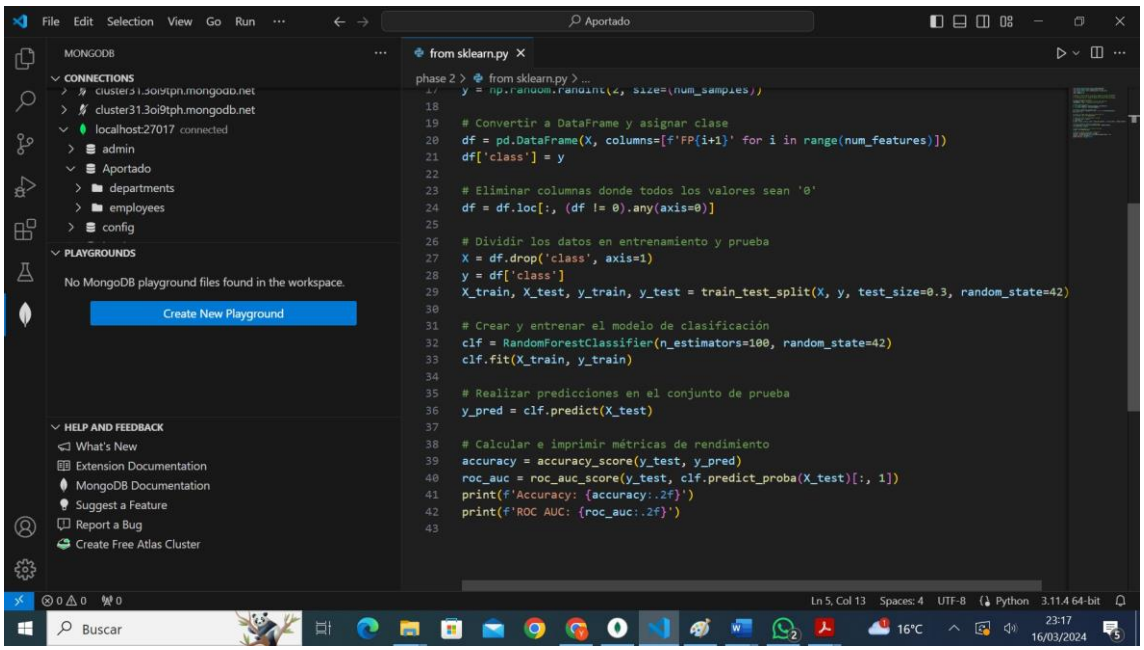
Interpretar los resultados del modelo, entender la importancia de las diferentes características y sacar conclusiones que puedan aplicarse en contextos científicos o empresariales.

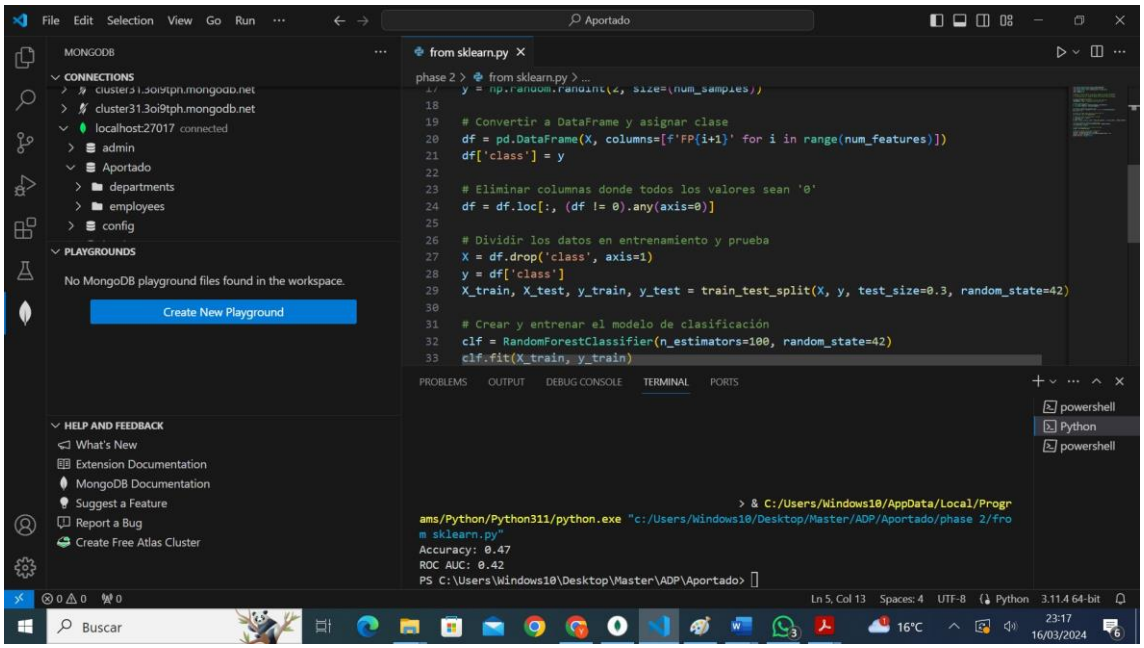
Etapa 10: Documentación y Reporte

Documentar el proceso y los hallazgos y preparar un informe detallado para los interesados o para la publicación de los resultados de la investigación.

Este resumen cubre el proceso general de desarrollo y evaluación de un modelo de clasificación en el contexto de aplicaciones científicas y empresariales, desde la recopilación de datos hasta la documentación final y la generación de informes.


File Edit Selection View Go Run
Aportado
MONGODB
CONNECTIONS
cluster31.3oi9tph.mongodb.net
localhost:27017 connected
admin
Aportado
departments
employees
config
PLAYGROUNDS
No MongoDB playground files found in the workspace.
Create New Playground
HELP AND FEEDBACK
What's New
Extension Documentation
MongoDB Documentation
Suggest a Feature
Report a Bug
Create Free Atlas Cluster
from sklearn.py
phase 2 > from sklearn.py > ...
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, roc_auc_score
import numpy as np
import pandas as pd
# Simular la carga de datos. En tu caso, cargarías los datos de MongoDB.
# Por ahora, voy a crear un DataFrame de ejemplo con datos ficticios.
# Tu DataFrame debe ser creado basándote en tus datos reales de MongoDB.
# Crear un DataFrame de ejemplo
num_samples = 100 # Supongamos que tienes 100 compuestos
num_features = 1024 # Supongamos 1024 fingerprints
# Generar datos aleatorios
X = np.random.randint(2, size=(num_samples, num_features))
y = np.random.randint(2, size=(num_samples))
# Convertir a DataFrame y asignar clase
df = pd.DataFrame(X, columns=[f'FP{i+1}' for i in range(num_features)])
df['class'] = y
# Eliminar columnas donde todos los valores sean '0'
df = df.loc[:, (df != 0).any(axis=0)]
# Dividir los datos en entrenamiento y prueba
X = df.drop('class', axis=1)
y = df['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
Ln 5, Col 13 Spaces: 4 UTF-8 Python 3.11.4 64-bit
Buscar
16°C
23:15
16/03/2024


File Edit Selection View Go Run
Aportado
MONGODB
CONNECTIONS
cluster31.3oi9tph.mongodb.net
localhost:27017 connected
admin
Aportado
departments
employees
config
PLAYGROUNDS
No MongoDB playground files found in the workspace.
Create New Playground
HELP AND FEEDBACK
What's New
Extension Documentation
MongoDB Documentation
Suggest a Feature
Report a Bug
Create Free Atlas Cluster
from sklearn.py
phase 2 > from sklearn.py > ...
# Convertir a DataFrame y asignar clase
df = pd.DataFrame(X, columns=[f'FP{i+1}' for i in range(num_features)])
df['class'] = y
# Eliminar columnas donde todos los valores sean '0'
df = df.loc[:, (df != 0).any(axis=0)]
# Dividir los datos en entrenamiento y prueba
X = df.drop('class', axis=1)
y = df['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Crear y entrenar el modelo de clasificación
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
# Realizar predicciones en el conjunto de prueba
y_pred = clf.predict(X_test)
# Calcular e imprimir métricas de rendimiento
accuracy = accuracy_score(y_test, y_pred)
roc_auc = roc_auc_score(y_test, clf.predict_proba(X_test)[:, 1])
print(f'Accuracy: {accuracy:.2f}')
print(f'ROC AUC: {roc_auc:.2f}')
Ln 5, Col 13 Spaces: 4 UTF-8 Python 3.11.4 64-bit
Buscar
16°C
23:17
16/03/2024


File Edit Selection View Go Run
Aportado
MONGODB
CONNECTIONS
cluster31.3oi9tph.mongodb.net
localhost:27017 connected
admin
Aportado
departments
employees
config
PLAYGROUNDS
No MongoDB playground files found in the workspace.
Create New Playground
HELP AND FEEDBACK
What's New
Extension Documentation
MongoDB Documentation
Suggest a Feature
Report a Bug
Create Free Atlas Cluster
from sklearn.py
phase 2 > from sklearn.py > ...
# Convertir a DataFrame y asignar clase
df = pd.DataFrame(X, columns=[f'FP{i+1}' for i in range(num_features)])
df['class'] = y
# Eliminar columnas donde todos los valores sean '0'
df = df.loc[:, (df != 0).any(axis=0)]
# Dividir los datos en entrenamiento y prueba
X = df.drop('class', axis=1)
y = df['class']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
# Crear y entrenar el modelo de clasificación
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
Python
powershell
> & C:/Users/Windows10/AppData/Local/Programs/Python/Python311/python.exe "c:/Users/Windows10/Desktop/Master/ADP/Aportado/phase 2/from sklearn.py"
Accuracy: 0.47
ROC AUC: 0.42
PS C:\Users\Windows10\Desktop\Master\ADP\Aportado>
Ln 5, Col 13 Spaces: 4 UTF-8 Python 3.11.4 64-bit
Buscar
16°C
23:17
16/03/2024