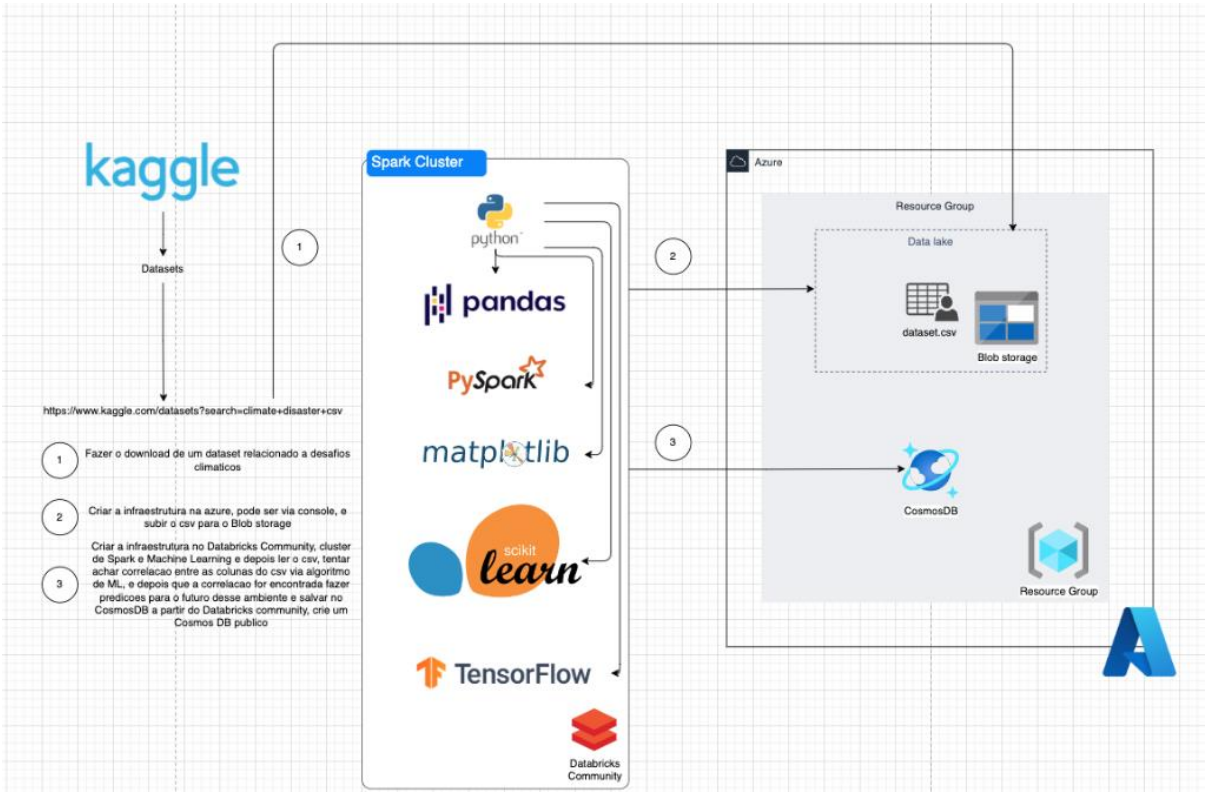
Global Solutions

RM	Nome
Caiky de Ávila Pereira Dutra	99274
Lucas Pinheiro de Souza	99061
Yann Dantas	550783

Arquitetura solicitada para tarefa



Passo a passo da criação da nossa infraestrutura

O dataset escolhido para essa tarefa refere-se ao consumo de energia disponibilizado através desse link do Kaggle:

https://www.kaggle.com/datasets/slawagrechaniy/powerconsumption/data

Para acesso do código fonte através do Github basta clicar neste link:

https://github.com/lucaspinheiro27/Globalsolutions_cloud

1. Criação de um Resource Group

O Resource Group é o contêiner lógico que agrupa todos os recursos da Azure para facilitar o gerenciamento. Ele permite organizar e gerenciar os recursos necessários para o projeto em um único lugar

```
Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
mr99061 [ ~ ]$ az group create --name GSResource --location brazilsouth
{
  "id": "/subscriptions/4c6283da-b703-409f-9293-b062565d0ccf/resourceGroups/GSResource",
  "location": "brazilsouth",
  "managedBy": null,
  "name": "GSResource",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
mr99061 [ ~ ]$
```

2. Configuração do Blob Storage

O Azure Blob Storage será usado como um data lake, onde o powerconsumption.csv será carregado. Este serviço oferece armazenamento econômico para grandes volumes de dados.

```
mr99061 [ ~ ]$ az storage account create --name globalsolutionstorage --resource-group GSResource --location brazilsouth --sku Standard_LRS
{
  "accessTier": "Hot",
  "accountMigrationInProgress": null,
  "allowBlobPublicAccess": false,
  "allowCrossTenantReplication": false,
  "allowSharedKeyAccess": null,
  "allowedCopyScope": null,
  "azureFilesIdentityBasedAuthentication": null,
  "blobRestoreStatus": null,
  "creationTime": "2024-11-21T00:56:12.633374+00:00",
  "customDomain": null,
  "defaultToOAuthAuthentication": null,
  "dnsEndpointType": null,
  "enableExtendedGroups": null,
  "enableHttpsTrafficOnly": true,
  "enableNfsV3": null,
  "encryption": {
    "encryptionIdentity": null,
    "keySource": "Microsoft.Storage",
    "keyVaultProperties": null,
    "requireInfrastructureEncryption": null,
    "services": {
      "blob": {
        "enabled": true,
        "keyType": "Account",
        "lastEnabledTime": "2024-11-21T00:56:12.727195+00:00"
      },
      "file": {
        "enabled": true,
        "keyType": "Account",
        "lastEnabledTime": "2024-11-21T00:56:12.727195+00:00"
      },
      "queue": null,
      "table": null
    }
  },
```

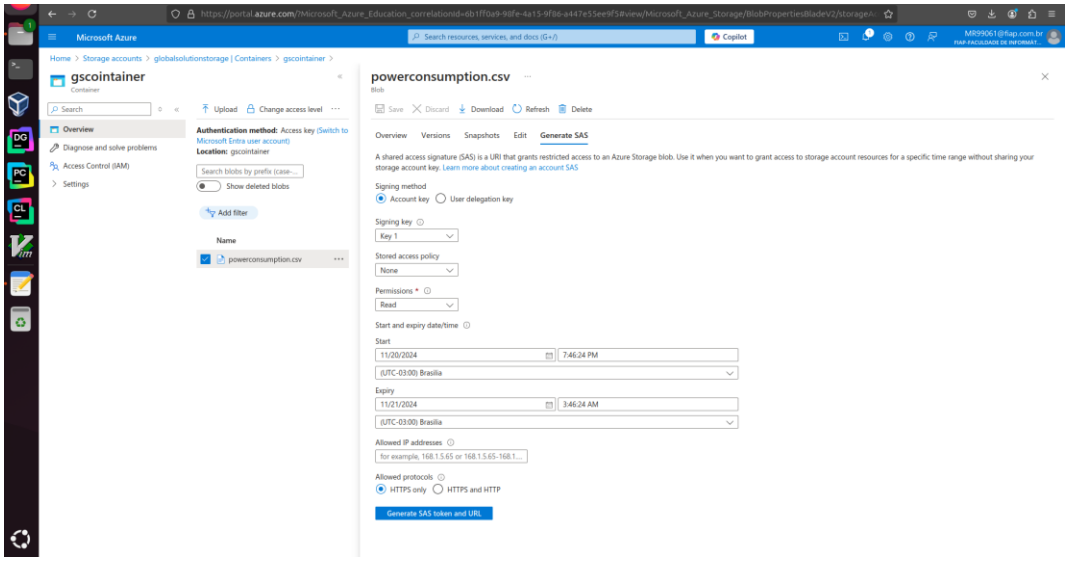
```
mr99061 [ ~ ]$ az storage container create --account-name globalsolutionstorage --name gscointainer

There are no credentials provided in your command and environment, we will query for account key for your storage account.
It is recommended to provide --connection-string, --account-key or --sas-token in your command as credentials.

You also can add `--auth-mode login` in your command to use Azure Active Directory (Azure AD) for authorization if your login account is assigned required RBAC roles.
For more information about RBAC roles in storage, visit https://docs.microsoft.com/azure/storage/common/storage-auth-aad-rbac-cli.

In addition, setting the corresponding environment variables can avoid inputting credentials in your command. Please use --help to get more information about environment variable usage.
{
  "created": true
}
mr99061 [ ~ ]$
```

Após o carregamento do arquivo, não esqueça de gerar o token SAS para o arquivo ser carregado no Databricks posteriormente.



3. Cosmos DB para Armazenamento de Resultados

O Azure Cosmos DB será utilizado para armazenar os dados processados e os resultados das análises. Ele permite que os dados estejam disponíveis para consultas em tempo real e oferece uma integração com MongoDB API, facilitando o acesso.

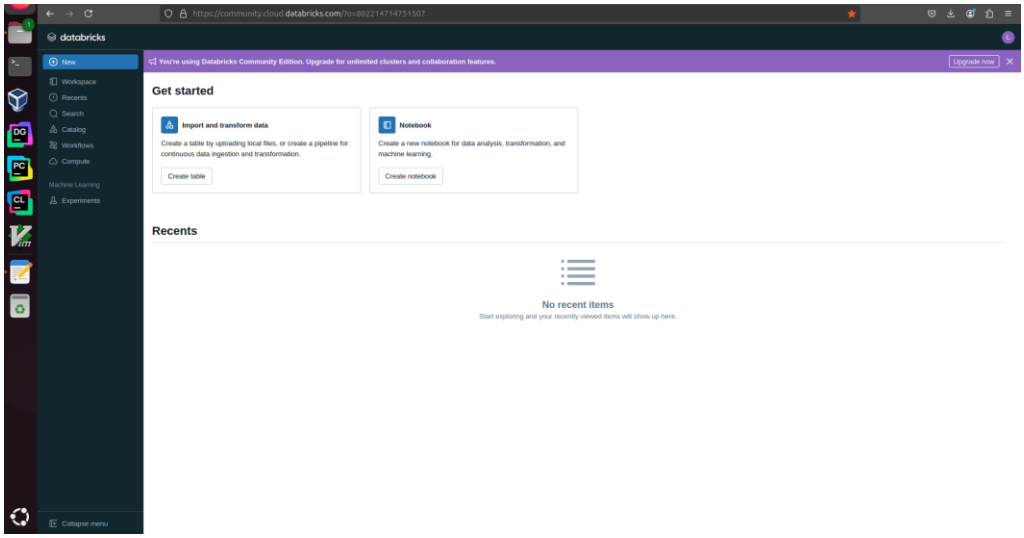
```
mr99061 [ ~ ]$ az cosmosdb create --name gscosmosdb --resource-group GSResource --kind MongoDB --locations regionName=brazilsouth failoverPriority=0 isZoneRedundant=False
{
  "analyticalStorageConfiguration": {
    "schemaType": "FullFidelity"
  },
  "apiProperties": {
    "serverVersion": "3.6"
  },
  "backupPolicy": {
    "migrationState": null,
    "periodicModeProperties": {
      "backupIntervalInMinutes": 240,
      "backupRetentionIntervalInHours": 8,
      "backupStorageRedundancy": "Geo"
    },
    "type": "Periodic"
  },
  "capabilities": [
    {
      "name": "EnableMongo"
    }
  ],
  "capacity": null,
  "connectorOffer": null,
  "consistencyPolicy": {
    "defaultConsistencyLevel": "Session",
    "maxIntervalInSeconds": 5,
    "maxStalenessPrefix": 100
  },
  "cors": [],
  "createMode": null,
  "customerManagedKeyStatus": null,
  "databaseAccountOfferType": "Standard",
  "defaultIdentity": "FirstPartyIdentity",
  "disableKeyBasedMetadataWriteAccess": false,
  "disableLocalAuth": false,
```

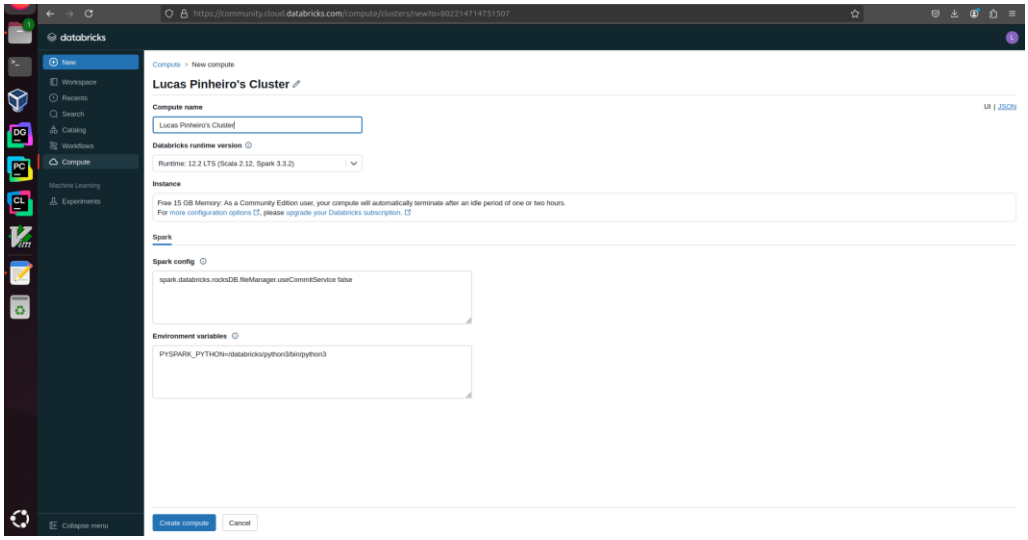
```
mr99061 [ ~ ]$ az cosmosdb mongodb database create \
--account-name  gscosmosdb \
--name mydatabase \
--resource-group GSResource
{
  "id": "/subscriptions/4c6283da-b703-409f-9293-b062565d0ccf/resourceGroups/GSResource/providers/Microsoft.DocumentDB/databaseAccounts/gscosmosdb/mongodbDatabases/mydatabase",
  "location": null,
  "name": "mydatabase",
  "options": null,
  "resource": {
    "createMode": null,
    "etag": null,
    "id": "mydatabase",
    "restoreParameters": null,
    "rid": null,
    "ts": null
  },
  "resourceGroup": "GSResource",
  "tags": null,
  "type": "Microsoft.DocumentDB/databaseAccounts/mongodbDatabases"
}
mr99061 [ ~ ]$
```

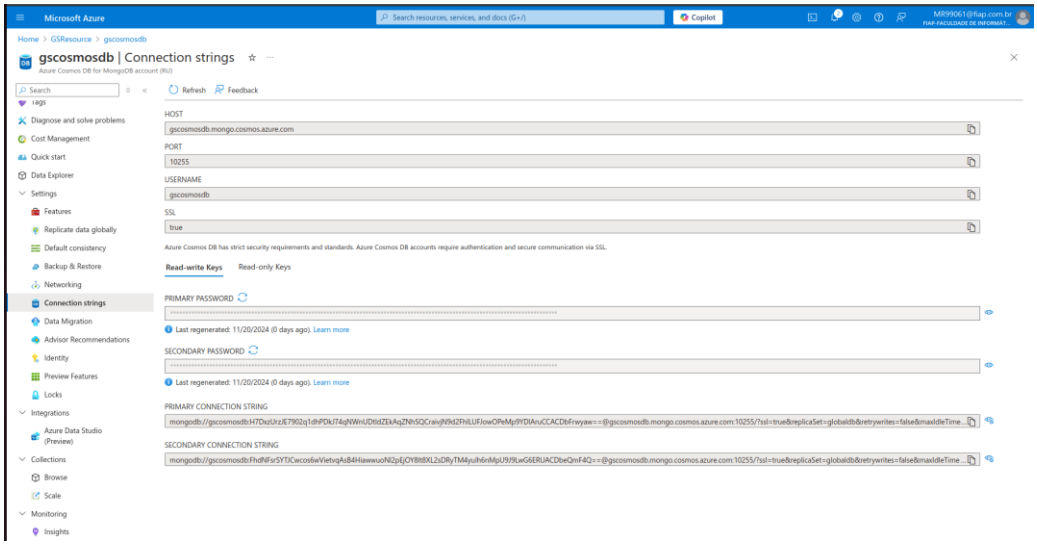
```
mr99061 [ ~ ]$ az cosmosdb mongodb collection create --account-name gscosmosdb --database-name mydatabase --name mycollection1 --resource-group GSResource --throughput 400
{
  "id": "/subscriptions/4c6283da-b703-409f-9293-b062565d0ccf/resourceGroups/GSResource/providers/Microsoft.DocumentDB/databaseAccounts/gscosmosdb/mongodbDatabases/mydatabase/collections/mycollection1",
  "location": null,
  "name": "mycollection1",
  "options": null,
  "resource": {
    "analyticalStorageTtl": null,
    "createMode": null,
    "etag": null,
    "id": "mycollection1",
    "indexes": [
      {
        "key": {
          "keys": [
            "_id"
          ]
        },
        "options": null
      }
    ],
    "restoreParameters": null,
    "rid": null,
    "shardKey": null,
    "ts": null
  },
  "resourceGroup": "GSResource",
  "tags": null,
  "type": "Microsoft.DocumentDB/databaseAccounts/mongodbDatabases/collections"
}
mr99061 [ ~ ]$
```

Fase 4 - Configuração do Databricks Communitypara o Cluster Spark:

Não esqueça de copiar a connection string do cosmosdb presente no container criado anteriormente

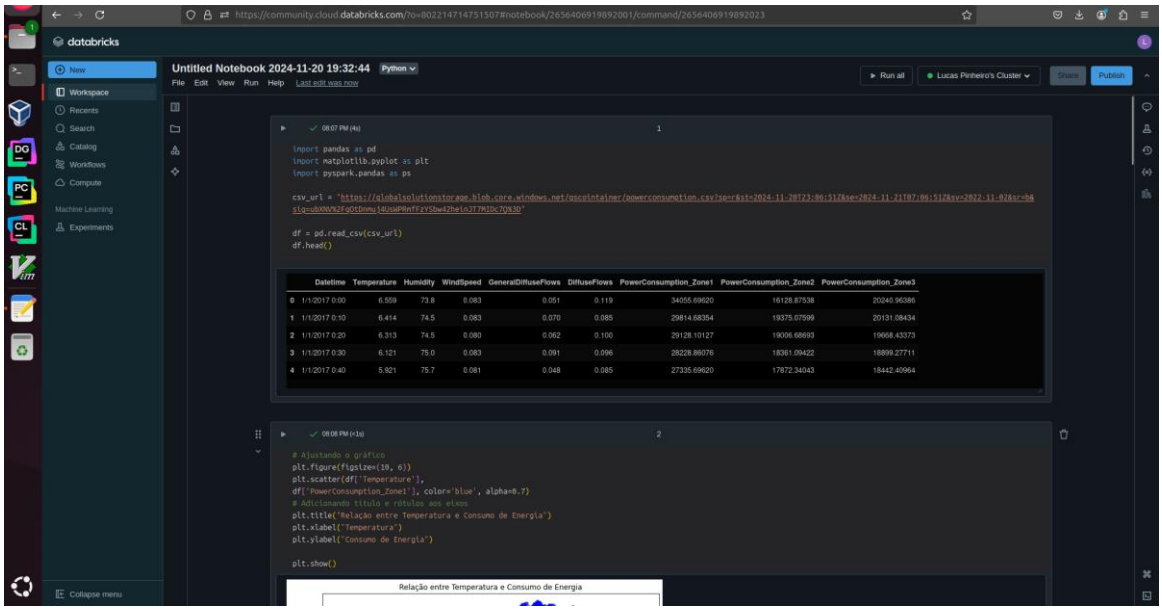


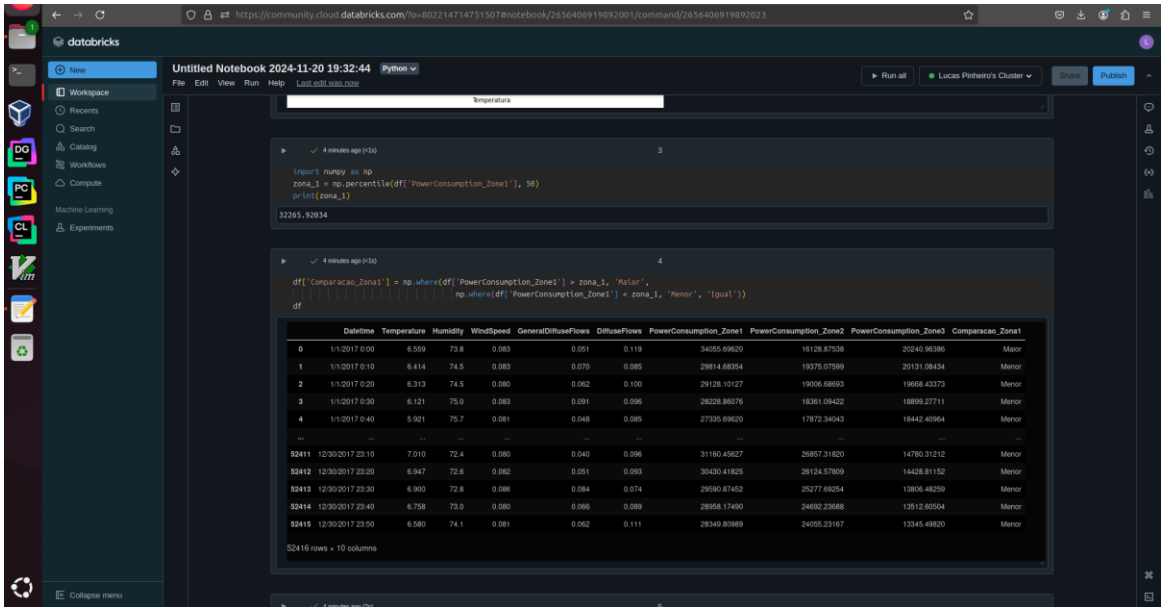


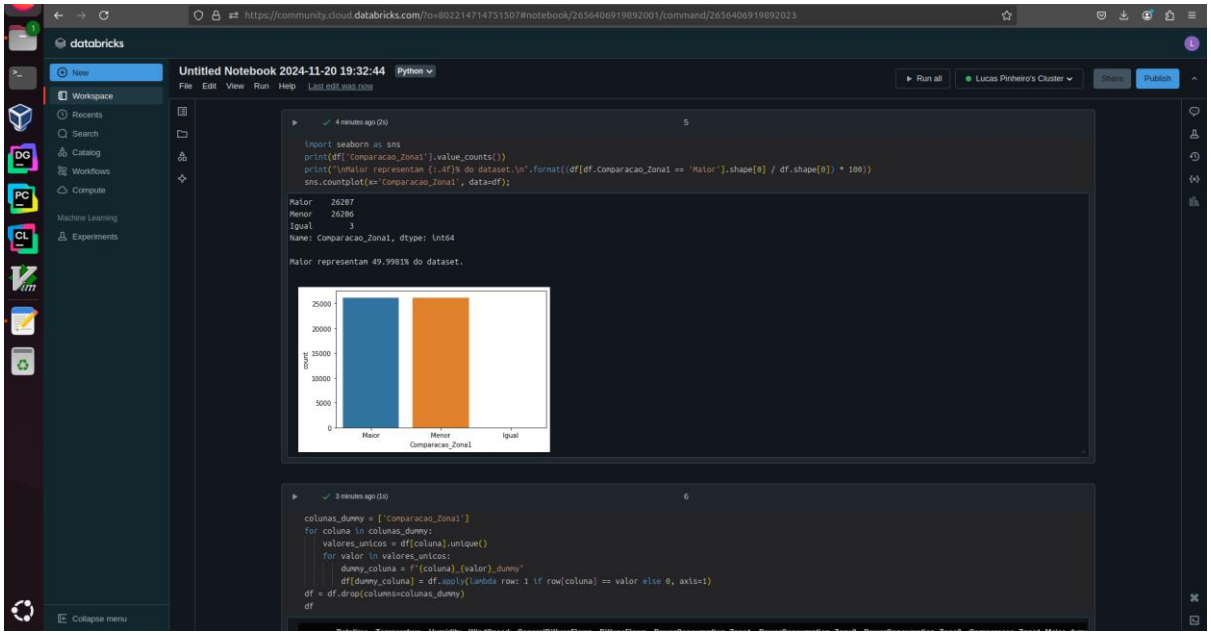


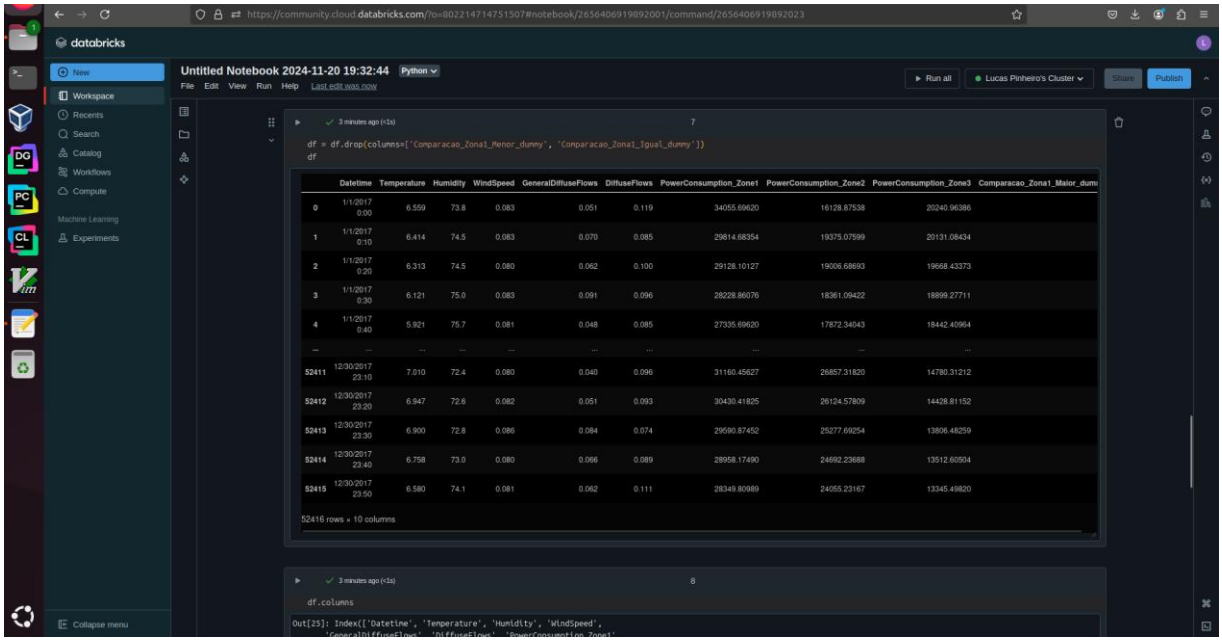
Iremos utilizar a URL do arquivo importado para um notebook do Databricks, depois iremos encontrar algumas análises, incluindo preparação e limpeza dos dados. Neste exemplo podemos notar que há uma correlação entre a temperatura e a Zona 1, a partir do percentil de 50, podemos

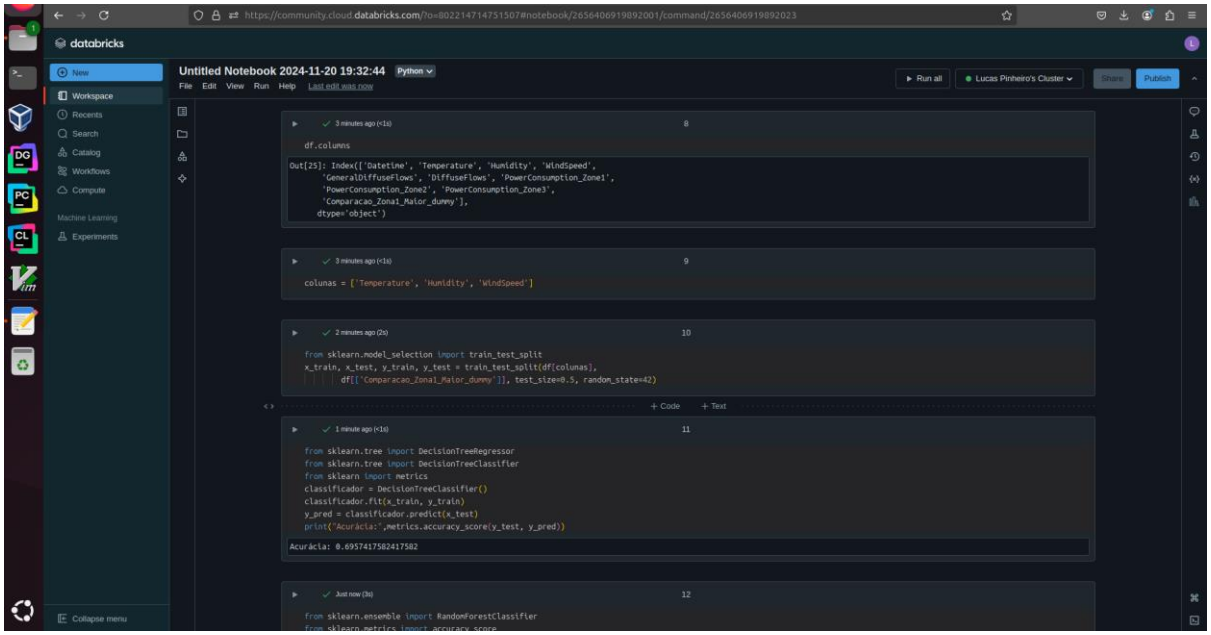
fazer a classificação dessa zona se é maior, menor ou igual. Verificamos se o há um balanceamento dos dados e faremos a dummização dessa coluna para futura predição do dataset. Nesta predição utilizamos as colunas ‘Temperatura’, ‘Humidade’ e ‘Velocidade do vento’. Utilizamos os algoritmos de árvore de decisão floresta aleatória

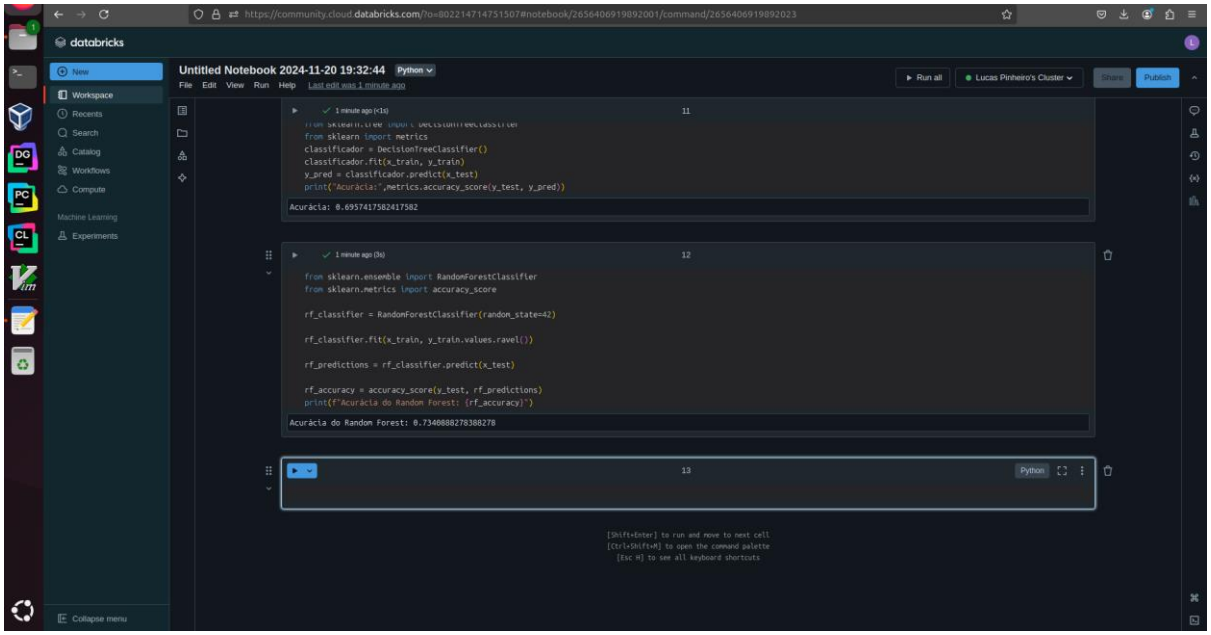


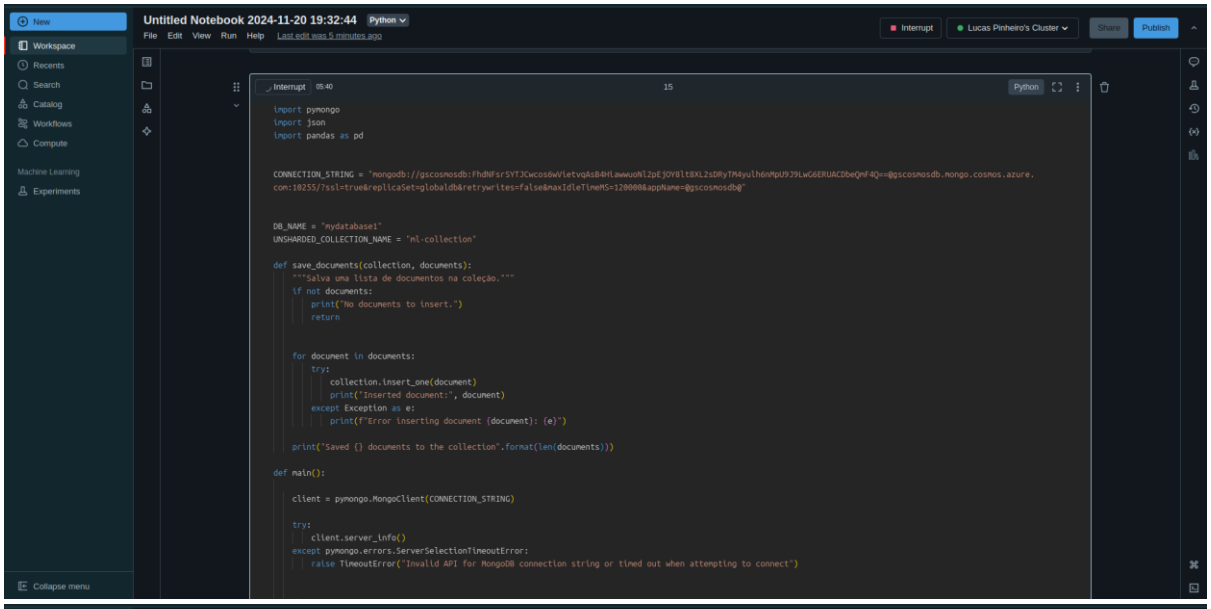


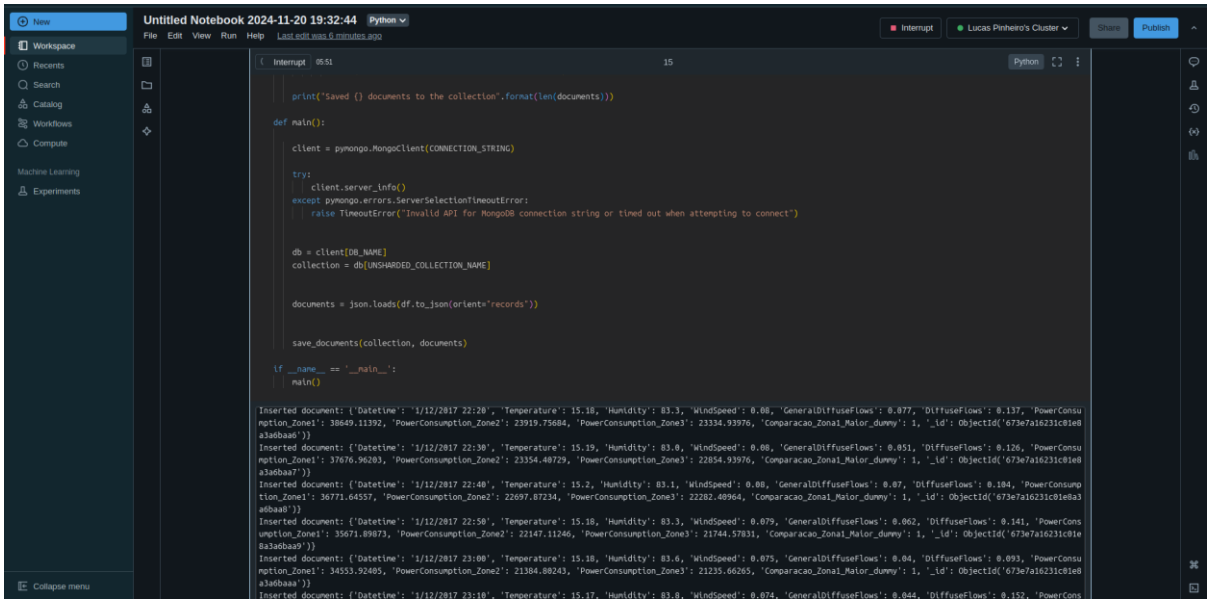

databricks
Untitled Notebook 2024-11-20 19:32:44
Python
File Edit View Run Help Last edit was now
Run all
Lucas Pinheiro's Cluster
Share
Publish
New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Collapse menu
4 minutes ago (2s)
5
import seaborn as sns
print(df['Comparacao_Zona1'].value_counts())
print("\nMaior representam {:.4f}% do dataset.\n".format((df[df.Comparacao_Zona1 == 'Maior'].shape[0] / df.shape[0]) * 100))
sns.countplot(x='Comparacao_Zona1', data=df);
Maior 26287
Menor 26206
Igual 3
Name: Comparacao_Zona1, dtype: int64
Maior representam 49.9981% do dataset.
count
Maior
Menor
Igual
Comparacao_Zona1
3 minutes ago (1s)
6
colunas_dummy = ['Comparacao_Zona1']
for coluna in colunas_dummy:
valores_unicos = df[coluna].unique()
for valor in valores_unicos:
dummy_coluna = f"{coluna}_{valor}_dummy"
df[dummy_coluna] = df.apply(lambda row: 1 if row[coluna] == valor else 0, axis=1)
df = df.drop(columns=colunas_dummy)
df


databricks
Untitled Notebook 2024-11-20 19:32:44
Python
File Edit View Run Help Last edit was now
Run all
Lucas Pinheiro's Cluster
Share
Publish
New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Collapse menu
3 minutes ago (<1s)
7
df = df.drop(columns=['Comparacao_Zona1_Menor_dummy', 'Comparacao_Zona1_Igual_dummy'])
df
Datetime Temperature Humidity WindSpeed GeneralDiffuseFlows DiffuseFlows PowerConsumption_Zone1 PowerConsumption_Zone2 PowerConsumption_Zone3 Comparacao_Zona1_Maior_dum
0 1/1/2017 0:00 6.559 73.8 0.083 0.051 0.119 34055.69620 16128.87538 20240.96386
1 1/1/2017 0:10 6.414 74.5 0.083 0.070 0.085 29814.68354 19375.07599 20131.08434
2 1/1/2017 0:20 6.313 74.5 0.080 0.062 0.100 29128.10127 19006.68693 19668.43373
3 1/1/2017 0:30 6.121 75.0 0.083 0.091 0.096 28228.86076 18361.09422 18899.27711
4 1/1/2017 0:40 5.921 75.7 0.081 0.048 0.085 27335.69620 17872.34043 18442.40964
52411 12/30/2017 23:10 7.010 72.4 0.080 0.040 0.096 31160.45627 26857.31820 14780.31212
52412 12/30/2017 23:20 6.947 72.6 0.082 0.051 0.093 30430.41825 26124.57809 14428.81152
52413 12/30/2017 23:30 6.900 72.8 0.086 0.084 0.074 29590.87452 25277.69254 13806.48259
52414 12/30/2017 23:40 6.758 73.0 0.080 0.066 0.089 28958.17490 24692.23688 13512.60504
52415 12/30/2017 23:50 6.580 74.1 0.081 0.062 0.111 28349.80989 24055.23167 13345.49820
52416 rows × 10 columns
3 minutes ago (<1s)
8
df.columns
Out[25]: Index(['Datetime', 'Temperature', 'Humidity', 'WindSpeed',


databricks
Untitled Notebook 2024-11-20 19:32:44
File Edit View Run Help Last edit was now
Run all
Lucas Pinheiro's Cluster
Share
Publish
New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Collapse menu
3 minutes ago (<1s)
8
df.columns
Out[25]: Index(['Datetime', 'Temperature', 'Humidity', 'WindSpeed',
'GeneralDiffuseFlows', 'DiffuseFlows', 'PowerConsumption_Zone1',
'PowerConsumption_Zone2', 'PowerConsumption_Zone3',
'Comparacao_Zonal_Maior_dummy'],
dtype='object')
3 minutes ago (<1s)
9
colunas = ['Temperature', 'Humidity', 'WindSpeed']
2 minutes ago (2s)
10
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(df[colunas],
df[['Comparacao_Zonal_Maior_dummy']], test_size=0.5, random_state=42)
+ Code + Text
1 minute ago (<1s)
11
from sklearn.tree import DecisionTreeRegressor
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
classificador = DecisionTreeClassifier()
classificador.fit(x_train, y_train)
y_pred = classificador.predict(x_test)
print("Acurácia:",metrics.accuracy_score(y_test, y_pred))
Acurácia: 0.6957417582417582
Just now (3s)
12
from sklearn.ensemble import RandomForestClassifier


databricks
Untitled Notebook 2024-11-20 19:32:44
File Edit View Run Help Last edit was 1 minute ago
Run all
Lucas Pinheiro's Cluster
Share
Publish
New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Collapse menu
1 minute ago (<1s)
11
from sklearn import metrics
classificador = DecisionTreeClassifier()
classificador.fit(x_train, y_train)
y_pred = classificador.predict(x_test)
print("Acurácia:",metrics.accuracy_score(y_test, y_pred))
Acurácia: 0.6957417582417582
1 minute ago (3s)
12
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
rf_classifier = RandomForestClassifier(random_state=42)
rf_classifier.fit(x_train, y_train.values.ravel())
rf_predictions = rf_classifier.predict(x_test)
rf_accuracy = accuracy_score(y_test, rf_predictions)
print(f"Acurácia do Random Forest: {rf_accuracy}")
Acurácia do Random Forest: 0.734088827838278
13
Python
[Shift+Enter] to run and move to next cell
[Ctrl+Shift+M] to open the command palette
[Esc H] to see all keyboard shortcuts


New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Collapse menu
Untitled Notebook 2024-11-20 19:32:44
Python
File Edit View Run Help Last edit was 5 minutes ago
Interrupt
Lucas Pinheiro's Cluster
Share
Publish
Interrupt 05:40
15
Python
import pymongo
import json
import pandas as pd
CONNECTION_STRING = "mongodb://gscosmosdb:FhdNFsrSYTJCwcos6wVletvqAsB4HiawwuoNl2pEJOY8lt8XL2sDRyTM4yulh6nMpU9J9LwG6ERUACDbeQmF4Q==@gscosmosdb.mongo.cosmos.azure.com:10255/?ssl=true&replicaSet=globaldb&retrywrites=false&maxIdleTimeMS=120000&appName=@gscosmosdb@"
DB_NAME = "mydatabase1"
UNSHARDED_COLLECTION_NAME = "nl-collection"
def save_documents(collection, documents):
"""Salva uma lista de documentos na coleção."""
if not documents:
print("No documents to insert.")
return
for document in documents:
try:
collection.insert_one(document)
print("Inserted document:", document)
except Exception as e:
print(f"Error inserting document {document}: {e}")
print("Saved {} documents to the collection".format(len(documents)))
def main():
client = pymongo.MongoClient(CONNECTION_STRING)
try:
client.server_info()
except pymongo.errors.ServerSelectionTimeoutError:
raise TimeoutError("Invalid API for MongoDB connection string or timed out when attempting to connect")


New
Workspace
Recents
Search
Catalog
Workflows
Compute
Machine Learning
Experiments
Collapse menu
Untitled Notebook 2024-11-20 19:32:44
Python
File Edit View Run Help Last edit was 6 minutes ago
Interrupt
Lucas Pinheiro's Cluster
Share
Publish
Interrupt 05:51
15
Python
print("Saved {} documents to the collection".format(len(documents)))
def main():
client = pymongo.MongoClient(CONNECTION_STRING)
try:
client.server_info()
except pymongo.errors.ServerSelectionTimeoutError:
raise TimeoutError("Invalid API for MongoDB connection string or timed out when attempting to connect")
db = client[DB_NAME]
collection = db[UNSHARDED_COLLECTION_NAME]
documents = json.loads(df.to_json(orient="records"))
save_documents(collection, documents)
if __name__ == '__main__':
main()
Inserted document: {'Datetime': '1/12/2017 22:20', 'Temperature': 15.18, 'Humidity': 83.3, 'WindSpeed': 0.08, 'GeneralDiffuseFlows': 0.077, 'DiffuseFlows': 0.137, 'PowerConsumption_Zone1': 38649.11392, 'PowerConsumption_Zone2': 23919.75684, 'PowerConsumption_Zone3': 23334.93976, 'Comparacao_Zona1_Maior_dummy': 1, '_id': ObjectId('673e7a16231c01e8a3a6baa6')}
Inserted document: {'Datetime': '1/12/2017 22:30', 'Temperature': 15.19, 'Humidity': 83.0, 'WindSpeed': 0.08, 'GeneralDiffuseFlows': 0.051, 'DiffuseFlows': 0.126, 'PowerConsumption_Zone1': 37676.96203, 'PowerConsumption_Zone2': 23354.40729, 'PowerConsumption_Zone3': 22854.93976, 'Comparacao_Zona1_Maior_dummy': 1, '_id': ObjectId('673e7a16231c01e8a3a6baa7')}
Inserted document: {'Datetime': '1/12/2017 22:40', 'Temperature': 15.2, 'Humidity': 83.1, 'WindSpeed': 0.08, 'GeneralDiffuseFlows': 0.07, 'DiffuseFlows': 0.104, 'PowerConsumption_Zone1': 36771.64557, 'PowerConsumption_Zone2': 22697.87234, 'PowerConsumption_Zone3': 22282.40964, 'Comparacao_Zona1_Maior_dummy': 1, '_id': ObjectId('673e7a16231c01e8a3a6baa8')}
Inserted document: {'Datetime': '1/12/2017 22:50', 'Temperature': 15.18, 'Humidity': 83.3, 'WindSpeed': 0.079, 'GeneralDiffuseFlows': 0.062, 'DiffuseFlows': 0.141, 'PowerConsumption_Zone1': 35671.89873, 'PowerConsumption_Zone2': 22147.11246, 'PowerConsumption_Zone3': 21744.57831, 'Comparacao_Zona1_Maior_dummy': 1, '_id': ObjectId('673e7a16231c01e8a3a6baa9')}
Inserted document: {'Datetime': '1/12/2017 23:00', 'Temperature': 15.18, 'Humidity': 83.6, 'WindSpeed': 0.075, 'GeneralDiffuseFlows': 0.04, 'DiffuseFlows': 0.093, 'PowerConsumption_Zone1': 34553.92405, 'PowerConsumption_Zone2': 21384.80243, 'PowerConsumption_Zone3': 21235.66265, 'Comparacao_Zona1_Maior_dummy': 1, '_id': ObjectId('673e7a16231c01e8a3a6baaa')}
Inserted document: {'Datetime': '1/12/2017 23:10', 'Temperature': 15.17, 'Humidity': 83.8, 'WindSpeed': 0.074, 'GeneralDiffuseFlows': 0.044, 'DiffuseFlows': 0.152, 'PowerCons