Azure Databricks Coding Challenge
Name : Mitushi Vishwakarma

Question 1 : Exploratory data analysis (EDA) in Databricks & Visualizing data in Databricks

Exploratory Data Analysis (EDA) in Databricks involves utilizing its robust tools for data exploration and visualization. Databricks provides a unified analytics platform that integrates Apache Spark for big data processing with SQL, Python, R, and Scala. Here's how we can perform EDA and visualize data in Databricks:

1. **Data Import:** First, import the dataset into Databricks. We can upload data files directly or connect to various data sources like Azure Data Lake Storage, AWS S3, or a database.
2. **Data Exploration**:
   - Visualization: Use Databricks' built-in visualization tools or integrate with popular libraries like Matplotlib, Seaborn, or Plotly for creating plots and charts.
3. **Visualizing Data**:
   - Databricks Visualizations: Databricks provides interactive visualization capabilities. You can create charts directly from DataFrame results using Databricks notebooks. Supported chart types include line plots, bar charts, scatter plots, and more.
4. **Advanced Analysis:**
   - Machine Learning: Databricks supports building machine learning models using Spark MLlib or integrating with popular ML frameworks.
   - Time Series Analysis: For time series data, use Databricks' time-series functions and visualization tools to analyze trends, seasonality, and anomalies.
5. **Collaboration and Sharing**: Databricks allows collaborative work through shared notebooks. You can share notebooks with team members for collaborative analysis and decision-making.
6. **Exporting Results**: Export your analysis results and visualizations in various formats like CSV, Excel, or images for sharing with stakeholders or further analysis.

Creating new visualizations:

In order to create visualizations, we need to have data.
Here I provided data and used the display() method.

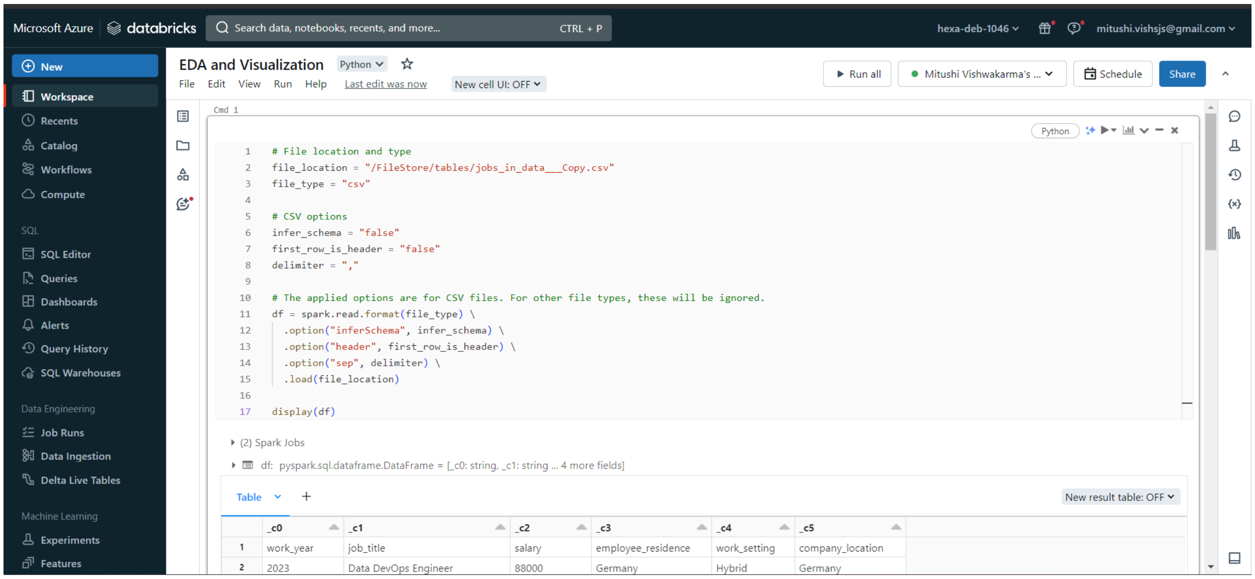
- After creating a table
- Click on + symbol
- Click on visualization.
- Select the type of visualization, here I selected “Pie” and “Bar”

In the visualization editor, We can select which type of chart we want in the drop down. There are various options then to groupBy, change colors,etc.

You can also change the color of the chart visualization by clicking “edit” on the bottom left corner.

You can edit , delete , view and download visualization charts.

Uploaded CSV file into DBFS and displaying it :

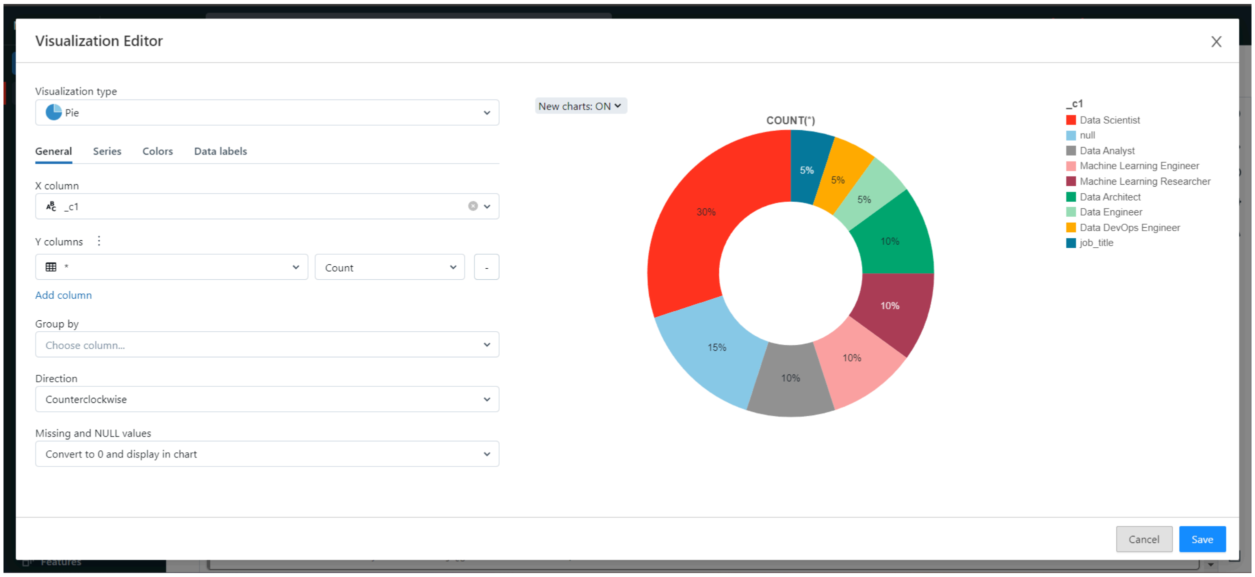


```
# File location and type
file_location = "/FileStore/tables/jobs_in_data___Copy.csv"
file_type = "csv"

# CSV options
infer_schema = "false"
first_row_is_header = "false"
delimiter = ","

# The applied options are for CSV files. For other file types, these will be ignored.
df = spark.read.format(file_type) \
  .option("inferSchema", infer_schema) \
  .option("header", first_row_is_header) \
  .option("sep", delimiter) \
  .load(file_location)

display(df)
```

```
(2) Spark Jobs
df:  pyspark.sql.dataframe.DataFrame = [_c0: string, _c1: string ... 4 more fields]
```

	_c0	_c1	_c2	_c3	_c4	_c5
1	work_year	job_title	salary	employee_residence	work_setting	company_location
2	2023	Data DevOps Engineer	88000	Germany	Hybrid	Germany

Creating Visualization 1 using “Pie” :



Visualization type: Pie

X column: _c1

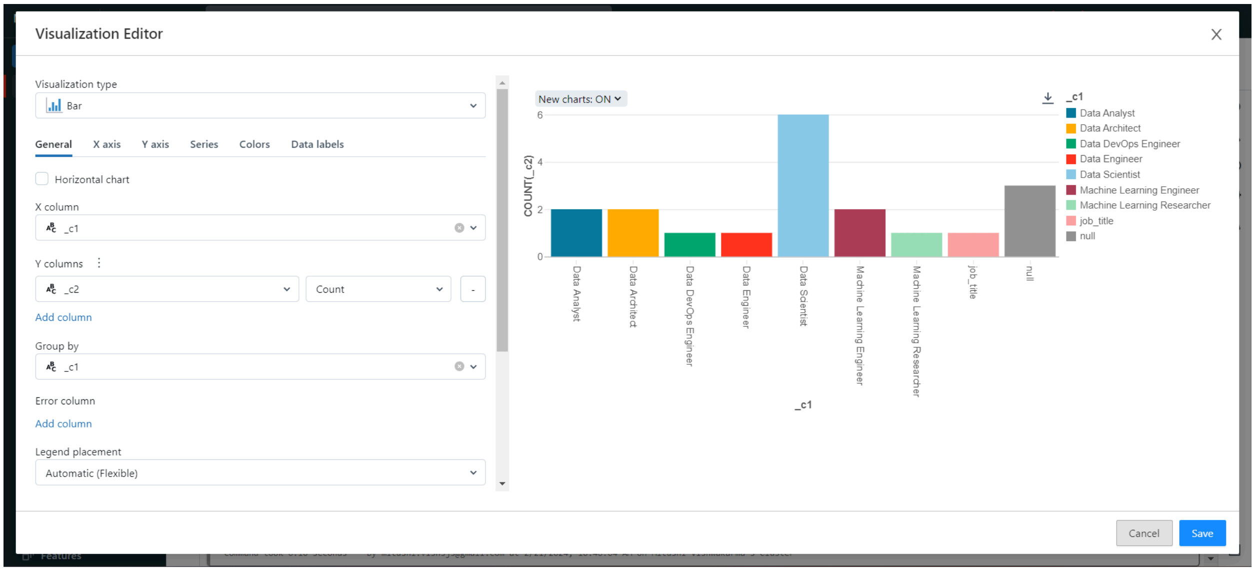
Y columns: * Count

Group by: Choose column...

Direction: Counterclockwise

Missing and NULL values: Convert to 0 and display in chart

Creating Visualization 2 using “Bar” :



Visualization Editor

Visualization type
Bar

General | X axis | Y axis | Series | Colors | Data labels

□ Horizontal chart

X column
_c1

Y columns
_c2 | Count

Add column

Group by
_c1

Error column
Add column

Legend placement
Automatic (Flexible)