

Configuration Manual

EVALUATION OF THE ROLE OF METADATA MANAGEMENT, ACCESS CONTROL AND DATA LINEAGE CAPABILITIES IN INDIAN IT SECTOR

MSc Research Project
MSc Cloud Computing

SRAVAN PEESU
Student ID: X23125721

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

**National College of Ireland
MSc Project Submission Sheet
School of Computing**



Student Name:	Sravan Peesu
Student ID:	X23125721
Programme:	MSc Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Mr. Vikas Sahni
Submission Due Date:	24/04/2025
Project Title:	EVALUATION OF THE ROLE OF METADATA MANAGEMENT, ACCESS CONTROL AND DATA LINEAGE CAPABILITIES IN INDIAN IT SECTOR
Word Count:	760
Page Count:	9

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Sravan Peesu
Date:	24/04/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
---	--------------------------

Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

1 Introduction

The guide contains detailed construction instructions for creating the necessary practical environment to execute the metadata management access control and data lineage analysis project through Python and Jupyter Notebook. The set of instructions targets technical employees who perform data science and machine learning work throughout the Indian IT industry..

2 Hardware and Software Requirements

Minimum hardware requirements:

- CPU: Intel i5 or equivalent (Quad-core)
- RAM: 8 GB minimum (16 GB recommended)
- Storage: 20 GB free disk space
- GPU: Optional, for parallel processing (e.g., NVIDIA GTX 1650 or above)

Software Requirements

- Operating System: Windows 10/11, macOS (10.15 or later), or Ubuntu 20.04+
- Python: Version 3.8 or later
- Jupyter Notebook: Installed via Anaconda or pip
- Git: Optional, for version control
- Web browser: Chrome/Firefox for Jupyter access

Gaining these Python packages occurs through pip or conda installations.

The necessary Python packages include pandas and numpy and matplotlib and seaborn and plotly and wordcloud and scikit-learn as well as jupyter.

The installed libraries enable data handling as well as visual representation and fundamental machine learning capabilities.

3 Environment Setup

- Install Anaconda
- Create a new environment:
 - `conda create --name datagov_env python=3.8`
 - `conda activate datagov_env`
- Install required libraries in the environment
- Launch Jupyter Notebook using:
 - `jupyter notebook`

4 Dataset Preparation

- Obtain the dataset (e.g., salary_data.csv)
- Ensure it is placed in the same directory as your notebook or provide the correct file path
- Dataset should be in CSV format, UTF-8 encoded, with no missing headers

5 Running the Code

- Open the Jupyter Notebook file (e.g., data_analysis.ipynb)
- Execute cells in order from top to bottom
- Modify file paths or configurations as necessary for your local setup
- Output visualizations and tables will be generated inline

6 Experiment Execution

- Adjust input filters such as experience range, location, and job title within the notebook
- Visual outputs will update dynamically using Plotly for interactivity
- Performance metrics such as runtime and memory usage can be measured using %timeit or memory_profiler
- Export final results to CSV or PNG using notebook functions

7 Troubleshooting

- **Jupyter not launching:** Ensure Jupyter is installed and PATH is set. Use jupyter notebook from Anaconda Prompt or terminal.
- **ModuleNotFoundError:** Use pip install <library> or ensure the correct environment is activated.
- **Data not loading:** Confirm file path and format (CSV), check for encoding issues.
- **Plotly charts not showing:** Ensure internet connectivity and import Plotly offline mode:
 - `from plotly.offline import init_notebook_mode;`
`init_notebook_mode(connected=True)`
- **Memory issues:** Work with smaller data samples or increase system RAM.
- This manual ensures a stable and reproducible setup to effectively run and interpret the data governance project.

8 Code Snippet

```
import pandas as pd
# import numpy as np
import matplotlib.pyplot as plt
import seaborn as sea
import plotly.express as pltex
from PIL import Image
from IPython.display import Image
from wordcloud import WordCloud, STOPWORDS

[ ] df = pd.read_csv("/content/Software_Professional_Salaries.csv")
df.head()

Rating      Company Name      Job Title      Salary      Salaries Reported      Location
0      3.8      Sasken      Android Developer      400000      3      Bangalore
1      4.5      Advanced Millennium Technologies      Android Developer      400000      3      Bangalore
2      4.0      Unacademy      Android Developer      1000000      3      Bangalore
3      3.8      SnapBizz Cloudlech      Android Developer      300000      3      Bangalore
4      4.4      Appoids Tech Solutions      Android Developer      600000      3      Bangalore

[ ] df.shape
(22774, 6)

[ ] df.columns
Index(['Rating', 'Company Name', 'Job Title', 'Salary', 'Salaries Reported',
      'Location'],
      dtype='object')

[ ] df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22774 entries, 0 to 22773
Data columns (total 6 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Rating      22774 non-null  float64
1   Company Name  22773 non-null  object
2   Job Title    22774 non-null  object
3   Salary       22774 non-null  int64
4   Salaries Reported  22774 non-null  int64
5   Location     22774 non-null  object
dtypes: float64(1), int64(2), object(3)
memory usage: 1.0+ MB
```

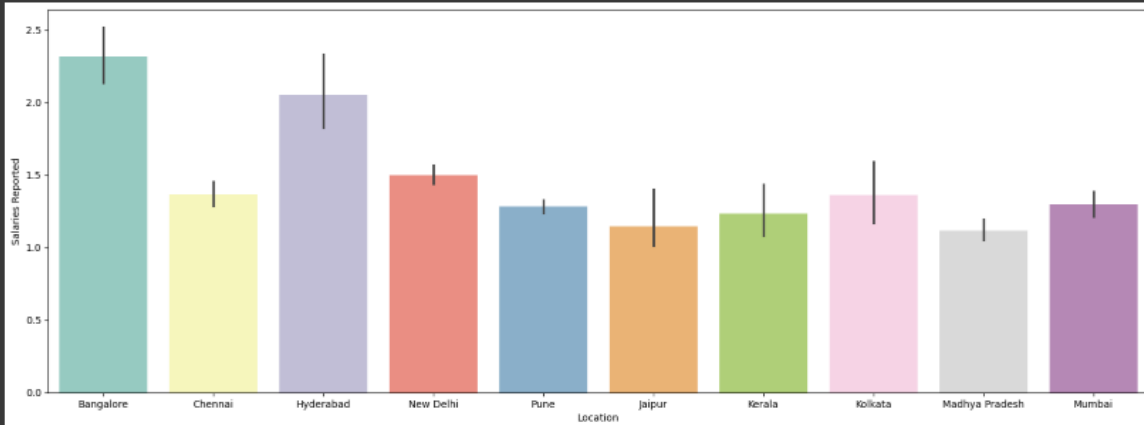
```

pyplot.figure(figsize = (20, 7))
seaborn.barplot(x = df["Location"], y = df["Salaries Reported"], palette = "Set3");

<ipython-input-82-b76926bdb09f>:2: FutureWarning:

```

Passing 'palette' without assigning 'hue' is deprecated and will be removed in v0.14.0. Assign the 'x' variable to 'hue' and set 'legend=False' for the same effect.

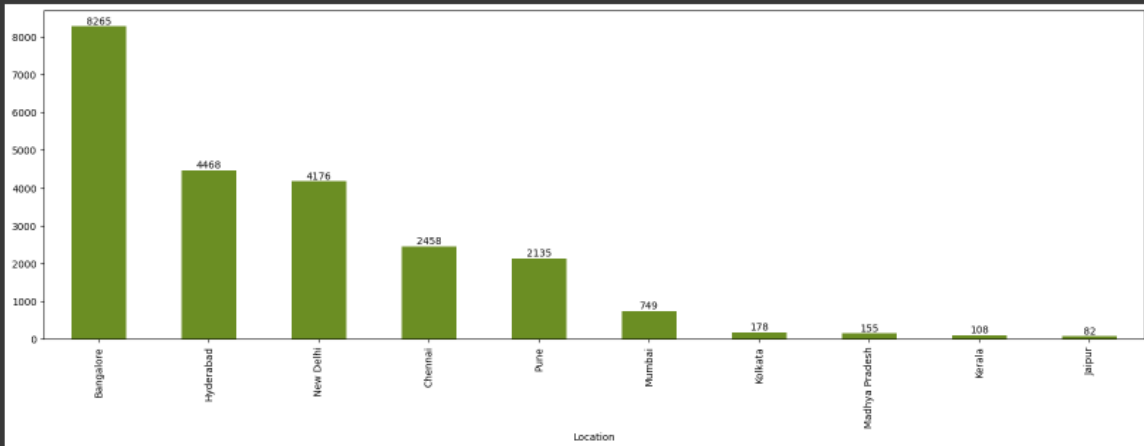


```

[ ] pyplot.figure(figsize = (20, 6))
ax = df.Location.value_counts().plot(kind = "bar", color = "olivedrab")

for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = "center", va = "bottom", color = "black")

```



```
[ ] df["Company Name"].value_counts()
```

Company Name	count
Tata Consultancy Services	273
Amazon	184
Infosys	169
Accenture	150
Cognizant Technology Solutions	144
...	...
IDEMIA	1
Alpha System (India)	1
Blasie	1
Aeolvis Technologies	1
Swisscontact Technologies	1

11262 rows x 1 columns

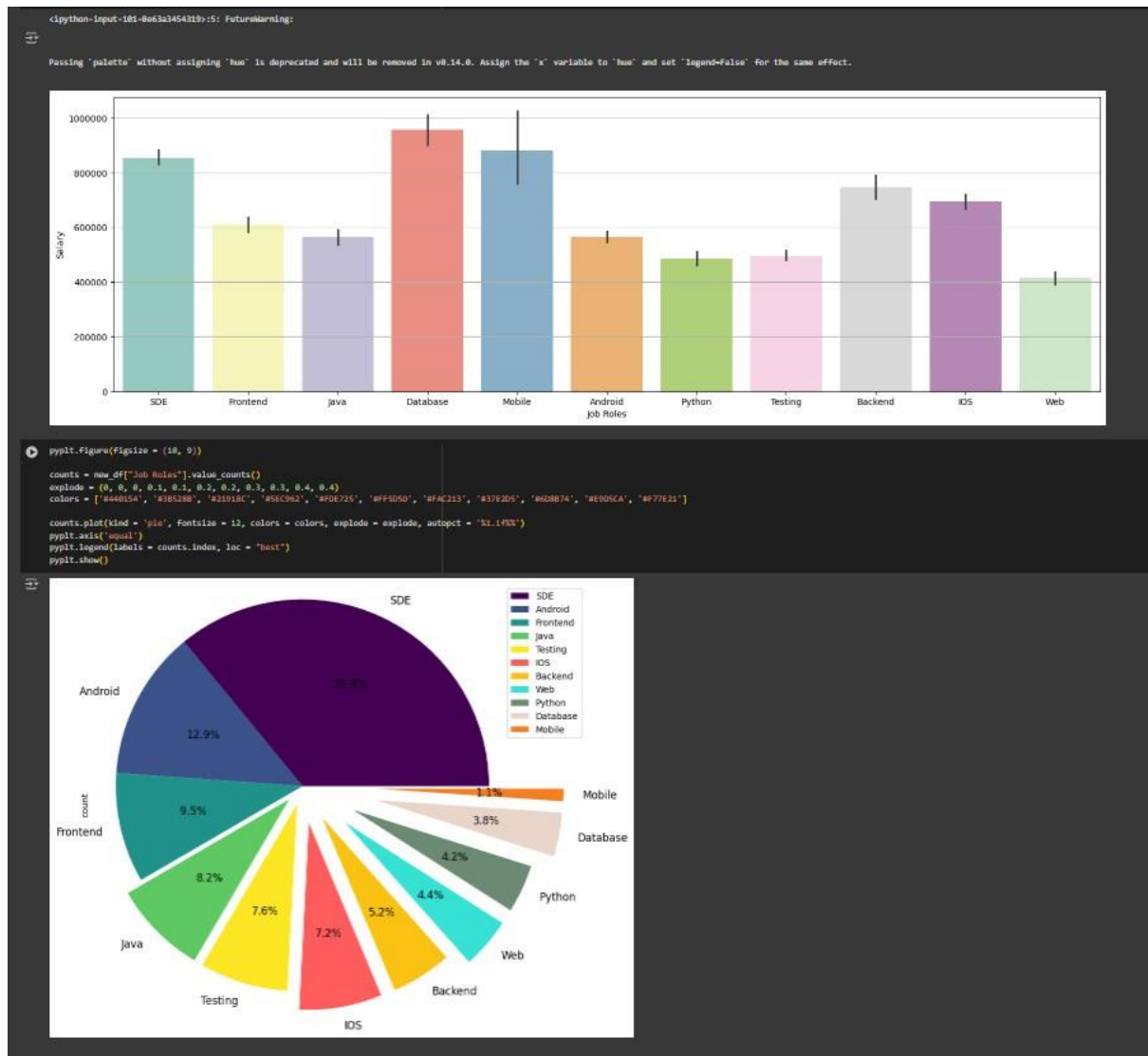
dtype: int64

```
[ ] text = "...join(Company for Company in df['Job Title'])"
font = "Times-Bold-10pt"
word_cloud = WordCloud(width=1600,
                        height=800,
                        colormap='jet',
                        background_color="white").generate(text)

pyplt.figure(figsize=(38, 6))
pyplt.imshow(word_cloud, interpolation="gaussian")
pyplt.axis("off")
pyplt.show()
```



```
[ ] pyplot.figure(figsize = (28, 6))
pyplot.grid()
pyplot.xticks(rotation = 90)
pyplot.ticklabel_format(style = 'plain')
df.sort_values("Salary", axis = 0,
ascending = False,
inplace = True)
```

9 References

anaconda.org, anaconda.org. (2019). :: *Anaconda Cloud*. Anaconda.org. <https://anaconda.org/>

jupyter. (2019). *Project Jupyter*. Jupyter.org. <https://jupyter.org/>

PYTHON. (n.d.). *Python*. Python.org; Python.org. <https://www.python.org/>

Python. (2020). *The Python Standard Library — Python 3.8.1 documentation*. Python.org.

<https://docs.python.org/3/library/index.html>