Name: Palak Bedi **PRN:22070521004**

Department: CSE **Division: A**

Symbiosis Institute of Technology, Nagpur

॥वसुधैव कुटुम्बकम्॥

DATA SCIENCE

Computer Science and Engineering Batch 2022-26

Course Name: DS

Course Code: - 0705210707 **Semester-VII**

INDEX

Exploratory Data Analysis: Employment Generated

INTRODUCTION

Exploratory Data Analysis (EDA) is a foundational step in any data-driven research or machine learning project. It helps us uncover the underlying patterns, structures, anomalies, and relationships within a dataset through a combination of statistical summaries and visualizations. By exploring the data thoroughly before applying any advanced modeling techniques, we can ensure cleaner inputs, more accurate insights, and better decision-making outcomes.

In this project, I chose to explore **rural employment generation data in India**, sourced from official records tracking job card distribution, employment demand, and job completion across thousands of Gram Panchayats. This dataset is significant as it reflects the **implementation of rural employment schemes such as MGNREGA**, which aim to guarantee livelihood security to millions of households in India's rural heartland.

Given the socio-economic importance of rural employment in a country where agriculture and informal labor dominate the workforce, analyzing trends in employment demand, availed work, person-days, and regional variations provides critical insights. It allows us to understand how employment benefits are distributed across states and districts, which areas may be underserved, and how various demographic groups (such as SC/ST) are represented in employment access.

Through this EDA, the goal is to not only clean and understand the data but also to derive meaningful insights that can inform policy makers, researchers, and social planners in enhancing rural livelihood programs and addressing employment inequities.

DATASET OVERVIEW

This dataset is a detailed compilation of information derived from the official Mahatma Gandhi National Rural Employment Guarantee Act (MGNREGA) portal, maintained by the Government of India. It captures data at the Gram Panchayat (GP) level, organized hierarchically by year, state, district, and block.
The primary objective of this dataset is to track and evaluate rural employment schemes, specifically those implemented under the MGNREGA initiative. The Act guarantees 100 days of wage employment in a financial year to rural households whose adult members volunteer to do unskilled manual work. This dataset provides comprehensive insights into key indicators such as household registration, job card issuance, employment demand, and actual employment availed.

1. The data serves multiple purposes:

- Policy evaluation and monitoring of rural employment schemes.
- Socio-economic analysis of employment distribution among different social categories (Scheduled Castes, Scheduled-Tribes,Others).
- Resource allocation and impact assessment at micro-administrative levels such as GP and block.

2. Dataset dimensions

- **Number of Rows**: Approx. 2642550 entries (based on unique Gram Panchayats across all districts and years).
- **Number of Columns**: 28 columns.

Each row corresponds to a record of employment-related metrics for a specific Gram Panchayat each year.

3. Column-wise Explanation and Significance

Column Name	Data Type	Description	Significance
year	Categorical	Financial year of data recording (e.g., 2014–2015).	Helps track temporal changes and trends.
state_name	Categorical	Name of the state (e.g., Uttar Pradesh, Maharashtra).	Enables state-wise comparison and aggregation.
district_name	Categorical	Name of the district within the state.	Useful for district-level analytics.
block_name	Categorical	Name of the block (subdivision of district).	Supports micro-level spatial analysis.
gp_name	Categorical	Name of the Gram Panchayat.	The most granular unit; essential for village-level tracking.
reg_hh	Numerical (int)	Number of households registered under MGNREGA.	Shows coverage potential at household level.
reg_pers	Numerical (int)	Number of persons registered.	Represents labor availability.
del_jobcards_hh	Numerical (int)	Number of household job cards deleted.	Indicates removed or disqualified households.
del_jobcards_pers	Numerical (int)	Number of individual job cards deleted.	Reflects individual-level disqualifications.
incl_jobcards_hh	Numerical (int)	Number of household job cards newly included.	Tracks recent additions of beneficiary households.
incl_jobcards_pers	Numerical (int)	Number of persons whose job cards were newly included.	Tracks inclusion at the individual level.
cumul_hh_jobcards_sc	Numerical (int)	Total job cards issued to SC households.	Enables equity-based policy analysis.

cumul_hh_jobcards_sts	Numerical (int)	Total job cards issued to ST households.	Helps in assessing tribal area outreach.
cumul_hh_jobcards_others	Numerical (int)	Job cards issued to Other Caste households.	Complements SC/ST data for total analysis.
cumul_hh_jobcards_tot	Numerical (int)	Total job cards issued across all castes.	Indicates the total reach of MGNREGA.
emp_demand_hh	Numerical (int)	Number of households demanding employment.	Measures real-time demand for work.
emp_demand_pers	Numerical (int)	Number of individuals demanding employment.	Individual-level demand analysis.
emp_offer_hh	Numerical (int)	Number of households offered employment.	Measures administrative responsiveness.
emp_offer_pers	Numerical (int)	Number of persons offered employment.	Individual offer stats.
emp_avail_hh	Numerical (int)	Number of households availing employment.	Indicates actual beneficiary coverage.
emp_avail_pers	Numerical (int)	Number of individuals availing employment.	Measures effective implementation.
emp_avail_tot_persondays	Numerical (int)	Total person-days of employment generated.	Key performance metric under MGNREGA.
emp_avail_central_persondays	Numerical (int)	Person-days funded by the central government.	Indicates central contribution.
emp_avail_states_persondays	Numerical (int)	Person-days funded by state government.	Important for state-level fiscal planning.
fam_completed_100_days	Numerical (int)	Households that completed 100 days of employment.	Measures full scheme utilization.
land_reform_benef_hh	Numerical (int)	Households benefiting from land reform/IAY schemes.	Indicates cross-scheme support and integration.
disabled_benef_indiv	Numerical (int)	Disabled individuals who benefited.	Assesses inclusive employment delivery.

STEPS FOR EXPLORATORY DATA ANALYSIS (EDA)

Step 0: Imports and Setup

Libraries which were imported to handle data manipulation and visualization are:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use("seaborn-v0_8-whitegrid")
pd.set_option("display.float_format", lambda x: '%.2f' % x)
```

- **pandas** were used for loading, cleaning, and manipulating tabular data.
- **numpy** provided support for numerical operations and efficient array handling.
- **matplotlib.pyplot** helped in creating basic plots and visualizations.
- **seaborn**, built on matplotlib, was used for creating attractive and informative statistical graphics.
- **warnings** were used to suppress unwanted warning messages that can clutter the output.

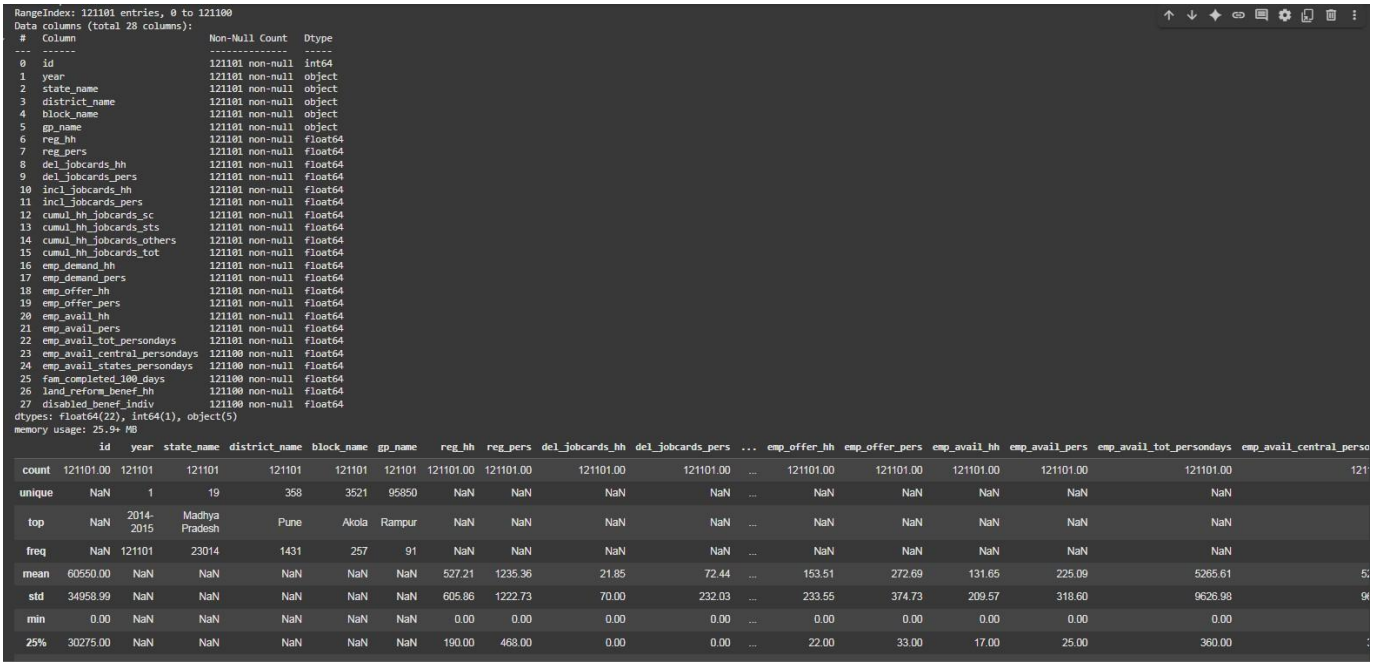
Step 1: Reading the Dataset

```
df = pd.read_csv("employment-generated.csv")
df.head()
```

The dataset was read using pandas.

Step 2: Initial Data Understanding

This step involves getting a basic sense of the dataset's structure, contents, and data types to guide further analysis. The following methods were used:



```
RangeIndex: 121101 entries, 0 to 121100
Data columns (total 28 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   id                          121101 non-null  int64
 1   year                        121101 non-null  object
 2   state_name                  121101 non-null  object
 3   district_name               121101 non-null  object
 4   block_name                  121101 non-null  object
 5   gp_name                     121101 non-null  object
 6   reg_hh                      121101 non-null  float64
 7   reg_pers                    121101 non-null  float64
 8   del_jobcards_hh             121101 non-null  float64
 9   del_jobcards_pers           121101 non-null  float64
 10  incl_jobcards_hh            121101 non-null  float64
 11  incl_jobcards_pers          121101 non-null  float64
 12  cumul_hh_jobcards_sc        121101 non-null  float64
 13  cumul_hh_jobcards_sts       121101 non-null  float64
 14  cumul_hh_jobcards_others    121101 non-null  float64
 15  cumul_hh_jobcards_tot       121101 non-null  float64
 16  emp_demand_hh               121101 non-null  float64
 17  emp_demand_pers             121101 non-null  float64
 18  emp_offer_hh                121101 non-null  float64
 19  emp_offer_pers              121101 non-null  float64
 20  emp_avail_hh                121101 non-null  float64
 21  emp_avail_pers              121101 non-null  float64
 22  emp_avail_tot_persondays    121101 non-null  float64
 23  emp_avail_central_persondays 121100 non-null  float64
 24  emp_avail_states_persondays 121100 non-null  float64
 25  fam_completed_100_days      121100 non-null  float64
 26  land_reform_benef_hh        121100 non-null  float64
 27  disabled_benef_indiv        121100 non-null  float64
dtypes: float64(22), int64(1), object(5)
memory usage: 25.9+ MB
```

	id	year	state_name	district_name	block_name	gp_name	reg_hh	reg_pers	del_jobcards_hh	del_jobcards_pers	...	emp_offer_hh	emp_offer_pers	emp_avail_hh	emp_avail_pers	emp_avail_tot_persondays	emp_avail_central_perso
count	121101.00	121101	121101	121101	121101	121101	121101.00	121101.00	121101.00	121101.00	...	121101.00	121101.00	121101.00	121101.00	121101.00	121
unique	NaN	1	19	358	3521	95850	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	
top	NaN	2014-2015	Madhya Pradesh	Pune	Akola	Rampur	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	
freq	NaN	121101	23014	1431	257	91	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	
mean	60550.00	NaN	NaN	NaN	NaN	NaN	527.21	1235.36	21.85	72.44	...	153.51	272.69	131.65	225.09	5265.61	5
std	34958.99	NaN	NaN	NaN	NaN	NaN	605.86	1222.73	70.00	232.03	...	233.55	374.73	209.57	318.60	9626.98	9
min	0.00	NaN	NaN	NaN	NaN	NaN	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	
25%	30275.00	NaN	NaN	NaN	NaN	NaN	190.00	468.00	0.00	0.00	...	22.00	33.00	17.00	25.00	360.00	

- **df.shape**: Returns the number of rows and columns in the dataset.
- **df.head()** and **df.tail()**: Display the first and last few rows of the dataset respectively.
- **df.dtypes**: Lists the data types of each column, which is useful for identifying categorical vs numerical variables and detecting any type mismatches.
- **df.columns**: Lists all the column names in the dataset.
- **df.describe()**: Generates summary statistics (like mean, standard deviation, min, max, and percentiles) for all numerical columns.

Step 3: Data Preparation and standardization

To ensure consistent structure and smooth analysis, several preprocessing steps were applied:

- **Duplicate Removal:** All exact duplicate rows were dropped using df.drop_duplicates() to prevent skewed insights.
- **Column Formatting:** All column names were converted to lowercase, stripped of extra spaces, and special characters were replaced with underscores for easier reference in code.
- **Year Type Conversion:** The Year column was converted to a string to facilitate categorical grouping and plotting.
- **Missing Value Handling:** All numeric columns were identified, and any missing (NaN) values were replaced with zero to avoid analytical errors and ensure uniformity.

Handling High-Cardinality Categorical Feature: gp_name:

- The gp_name column typically contains **thousands of unique entries** (e.g., 200,000+ across India).
- High-cardinality categorical features are **difficult to summarize**, visualize, or use meaningfully in most plots or models.

```
cols_to_drop = ['gp_name']  # Very granular for EDA stage; can be added later if needed
df.drop(columns=cols_to_drop, inplace=True)
```

Step 4: Univariate analysis

Univariate analysis involves examining the distribution, patterns, and summary statistics of a **single variable** at a time. It helps understand the nature (e.g., central tendency, spread, skewness) of each feature independently

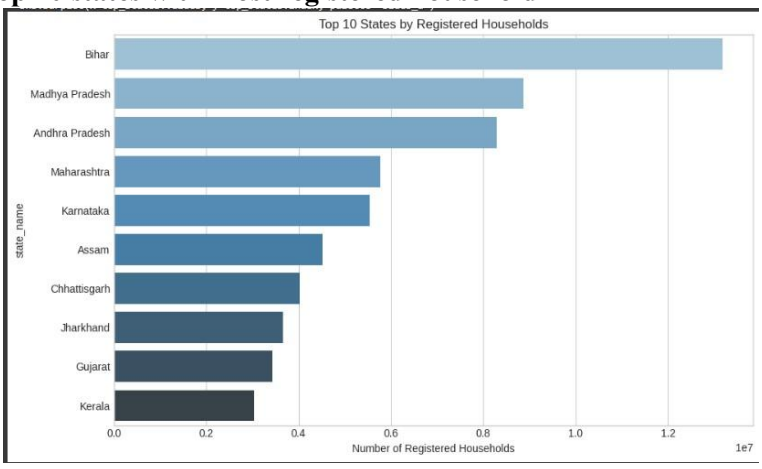
```
num_cols = df.select_dtypes(include=[np.number]).columns
df[num_cols].describe().T
```

	count	mean	std	min	25%	50%	75%	max
id	121101.00	60550.00	34958.99	0.00	30275.00	60550.00	90825.00	121100.00
reg_hh	121101.00	527.21	605.86	0.00	190.00	346.00	626.00	37660.00
reg_pers	121101.00	1235.36	1222.73	0.00	468.00	933.00	1626.00	57080.00
del_jobcards_hh	121101.00	21.85	70.00	0.00	0.00	0.00	8.00	2228.00
del_jobcards_pers	121101.00	72.44	232.03	0.00	0.00	2.00	34.00	11294.00
incl_jobcards_hh	121101.00	13.41	40.58	0.00	0.00	1.00	10.00	1751.00
incl_jobcards_pers	121101.00	29.60	81.20	0.00	0.00	5.00	25.00	4346.00
cumul_hh_jobcards_sc	121101.00	90.30	154.05	0.00	4.00	34.00	104.00	4360.00
cumul_hh_jobcards_sts	121101.00	82.79	225.17	0.00	0.00	11.00	85.00	28938.00
cumul_hh_jobcards_others	121101.00	347.95	458.77	0.00	85.00	204.00	413.00	14212.00
cumul_hh_jobcards_tot	121101.00	521.04	595.48	0.00	188.00	343.00	621.00	36430.00
emp_demand_hh	121101.00	154.06	233.91	0.00	22.00	94.00	210.00	11891.00
emp_demand_pers	121101.00	274.07	376.64	0.00	34.00	157.00	379.00	11924.00
emp_offer_hh	121101.00	153.51	233.55	0.00	22.00	94.00	209.00	11891.00
emp_offer_pers	121101.00	272.69	374.73	0.00	33.00	156.00	377.00	11924.00
emp_avail_hh	121101.00	131.65	209.57	0.00	17.00	77.00	177.00	11847.00
emp_avail_pers	121101.00	225.09	318.60	0.00	25.00	125.00	308.00	11879.00
emp_avail_tot_persondays	121101.00	5265.61	9626.98	0.00	360.00	2327.00	6500.00	450657.00
emp_avail_central_persondays	121101.00	5265.61	9626.98	0.00	360.00	2327.00	6500.00	450657.00
emp_avail_states_persondays	121101.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
fam_completed_100_days	121101.00	8.57	28.44	0.00	0.00	0.00	5.00	2063.00
land_reform_benef_hh	121101.00	7.89	38.77	0.00	0.00	0.00	0.00	4162.00
disabled_benef_indiv	121101.00	1.46	6.82	0.00	0.00	0.00	0.00	567.00

Step 5: Data Visualization

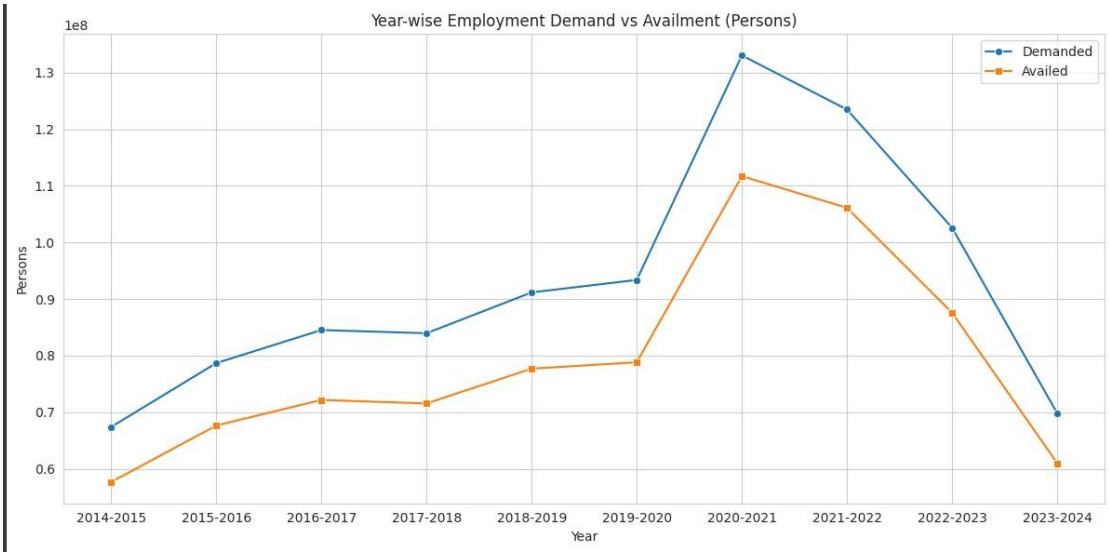
This section presents visual insights derived from the dataset using different types of plots. These visualizations help to interpret trends, variations, and relationships between key employment-related variables.

1. Top 10 states with most registered household



2. Year-wise Employment Demand vs. Availment

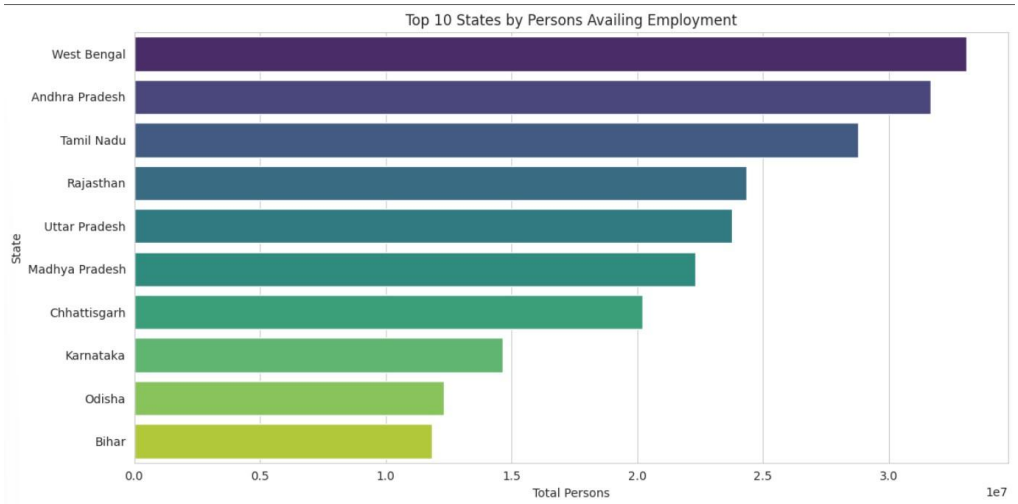
This line chart illustrates the number of persons who **demanded** and **availed** employment under the rural employment guarantee scheme, across financial years from **2014–15 to 2023–24**.



- **Blue line**: Represents persons who **demanded employment**.
- **Orange line**: Represents persons who **availed employment**.
- A **visible gap** exists between demand and availment in every year, indicating **unfulfilled employment requests**.
- **Steady rise** in both demand and availment is seen from 2014–15 to **2019–20**.
- A **sharp peak in 2020–21** reflects the **COVID-19 crisis**, where demand and availment rose drastically due to job loss and rural reverse migration.
- Post-pandemic years (2021–22 onwards) show a **gradual decline** in both figures, with 2023–24 nearing early-year levels.
- The gap remained wide, especially during high-demand years, showing that the scheme **couldn't fully meet employment needs**

3. Top 10 States by Persons Availing Employment

This horizontal bar chart shows the **top 10 Indian states** based on the number of persons who availed employment under the scheme.

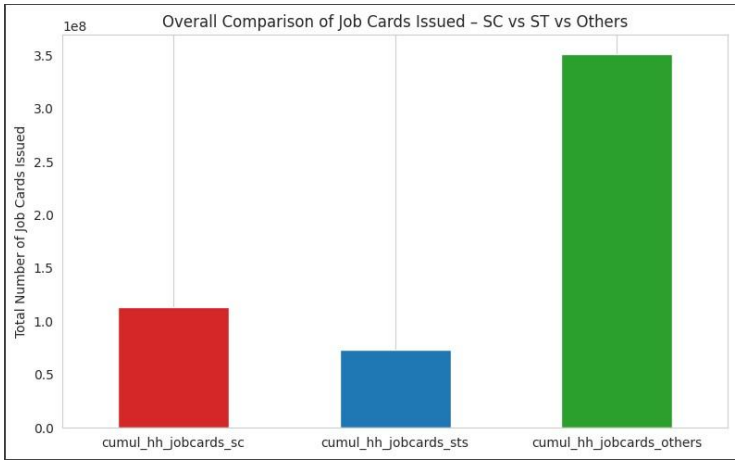


- **West Bengal** leads with the highest number, followed closely by **Andhra Pradesh**, **Tamil Nadu**, and **Rajasthan**.
- Other states like **Madhya Pradesh**, **Chhattisgarh**, and **Uttar Pradesh** also feature prominently.
- **Bihar** and **Odisha** show relatively lower availment figures among the top 10.

Interpretation: States with stronger rural infrastructure and higher demand have better employment availement. The disparity also reflects state-wise policy implementation efficiency.

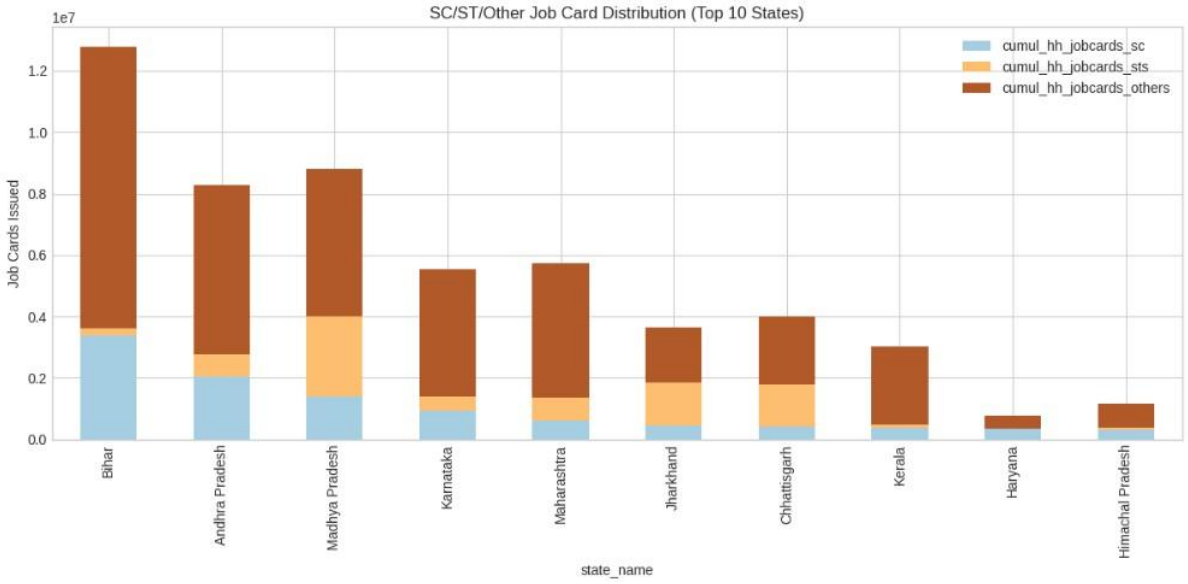
4. Comparison of Job Cards Issued – SC vs ST vs Others

This bar chart visually represents the **cumulative number of job cards issued** to households belonging to Scheduled Castes (SC), Scheduled Tribes (ST), and other communities across all districts and years in the dataset.

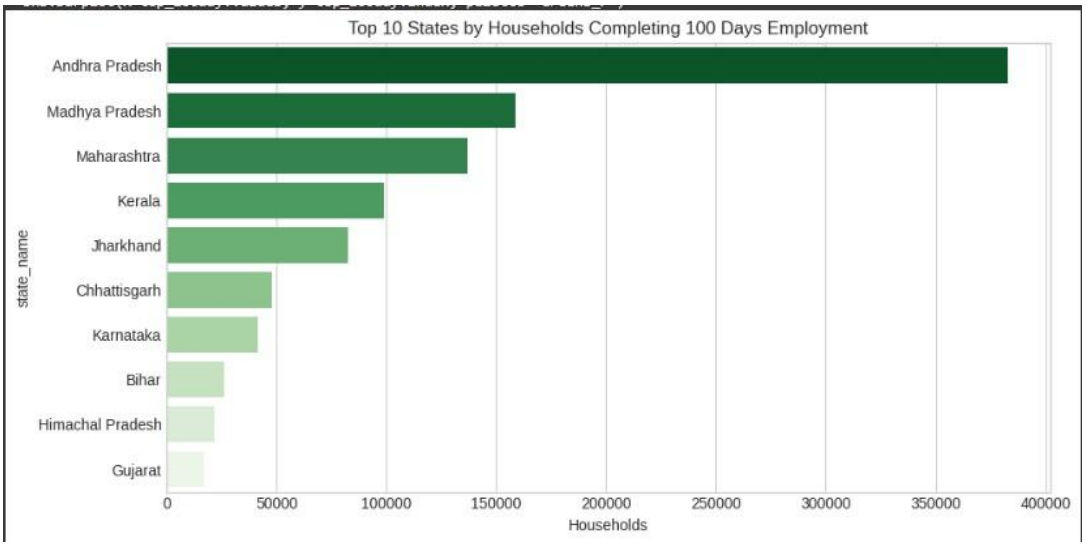


- **Green Bar (Others)**: The highest number of job cards were issued to households categorized as *Others*, with a total nearing **35 crore**.
- **Red Bar (SC)**: SC households received significantly fewer job cards in comparison, around **11 crore**.
- **Blue Bar (ST)**: ST households received the lowest among the three, with a count of approximately **7 crores**.

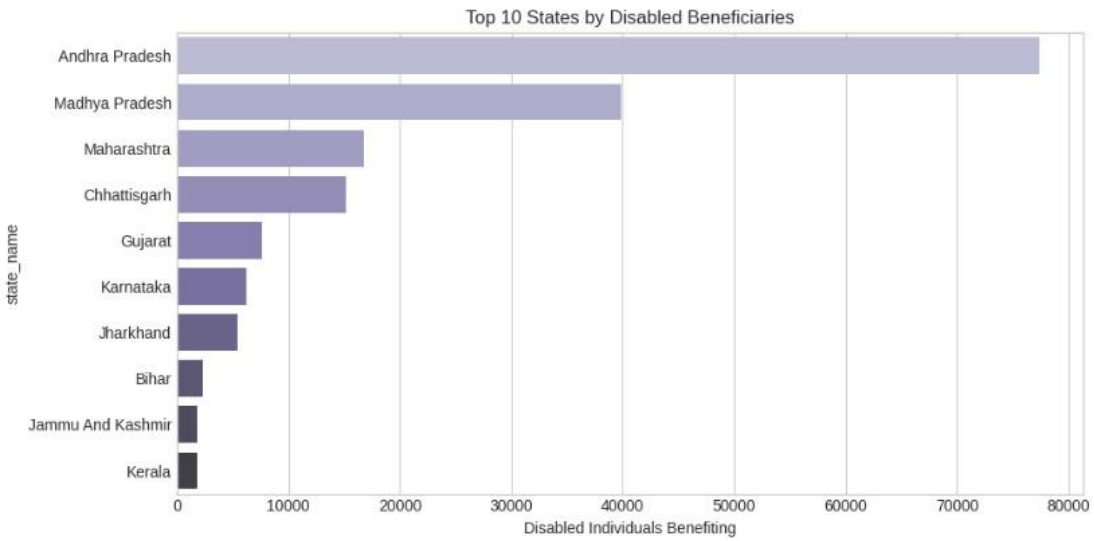
5. SC/ST/Others Job Card Distribution State wise



6. Top 10 States by household completing 100 days of employment

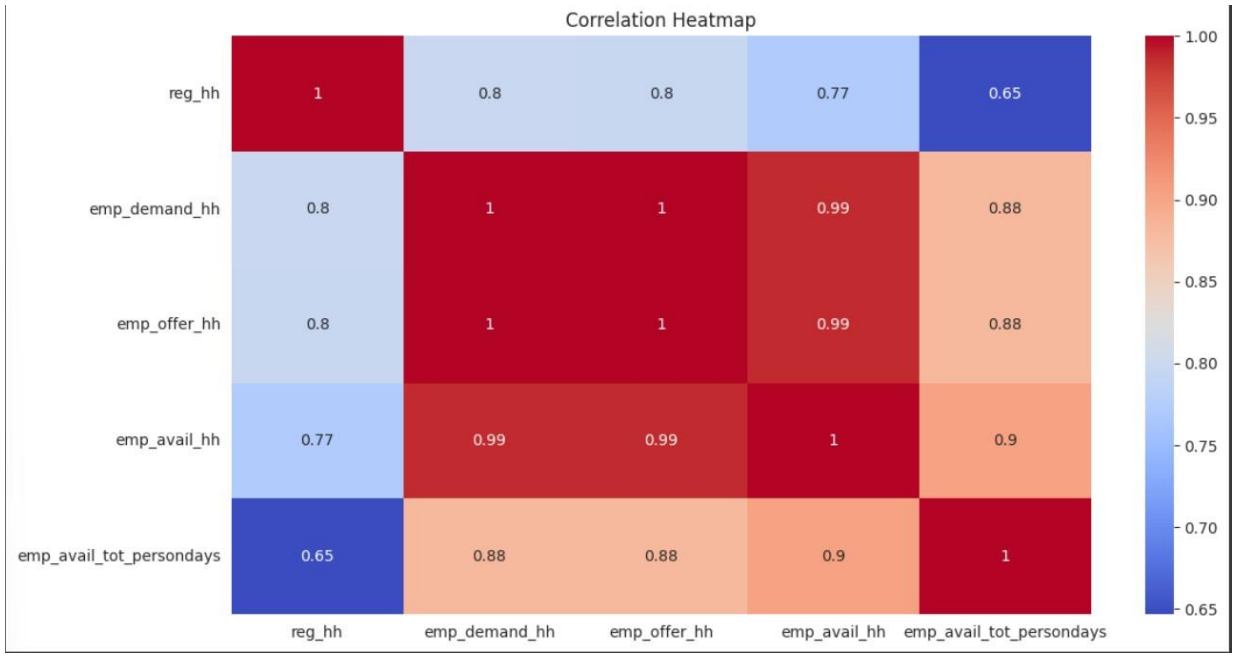


7. Disabled individual benefiting from the scheme



8. Correlation Heatmap of Key Variables

This heatmap shows the correlation between key numeric variables in the dataset:



- **emp_demand_hh**, **emp_offer_hh**, and **emp_avail_hh** are all **highly positively correlated** (close to 1), suggesting that when demand increases, offers and availment also rise proportionally.
- **emp_avail_tot_persondays** also shows a strong positive correlation with availment and offer variables.
- **reg_hh** (registered households) has a moderate positive correlation with all other metrics.

Interpretation: Strong interdependence among demand, offer, and availment suggests the system is generally responsive to demand, although gaps still exist. Registered households act as a base influencing these variables.

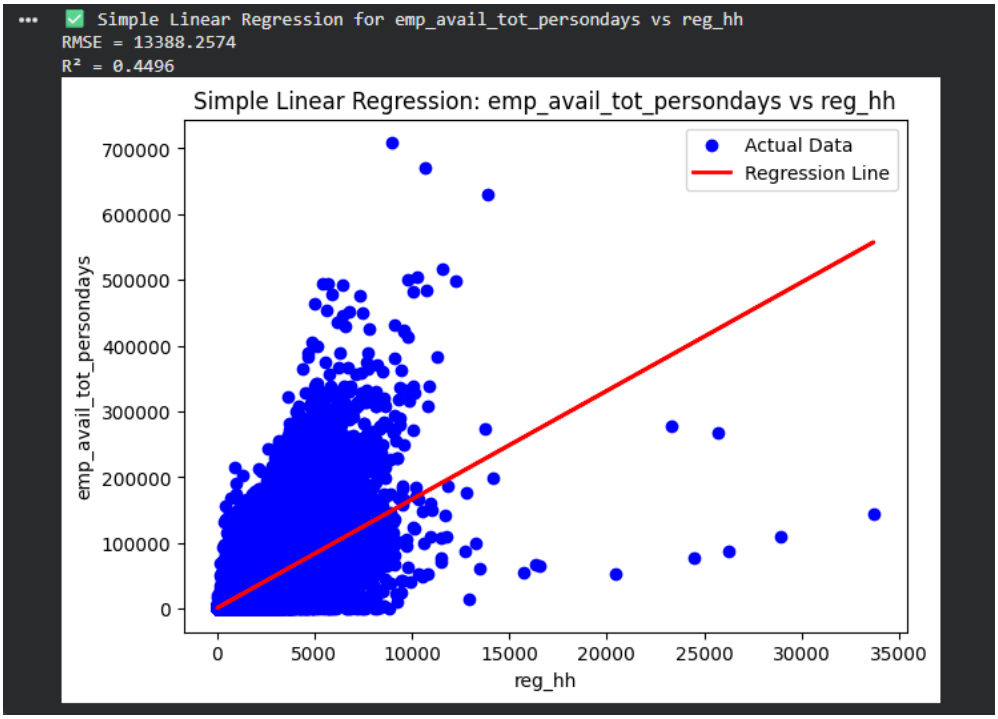
Step 5: Exporting the clean data

After data cleaning and preprocessing, the final DataFrame is exported to a CSV file using the following command:

```
[10] df.to_csv("cleaned_employment_dataset.csv", index=False)
```

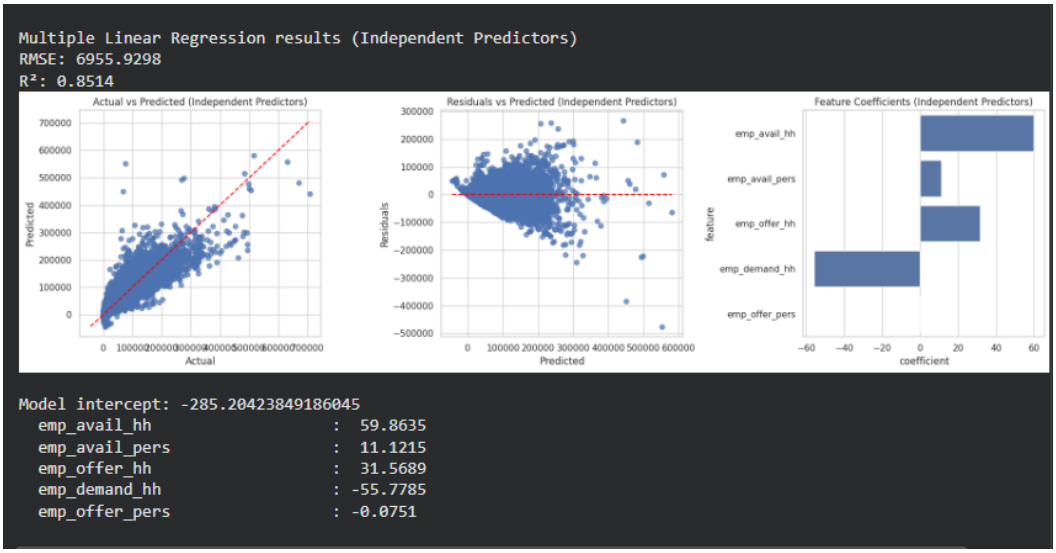
Tranning the model

1. **Simple linear regression :**



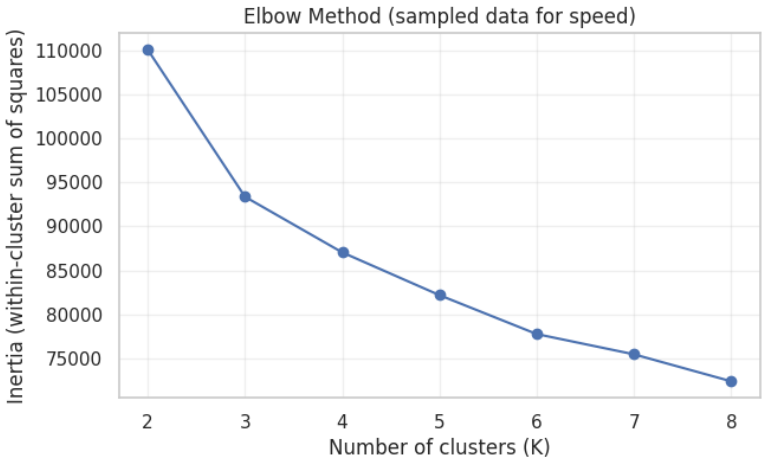
This graph represents a Simple Linear Regression analysis between registered households (reg_hh) and total persondays of employment generated (emp_avail_tot_persondays). The upward-sloping red regression line indicates a positive relationship, meaning that as the number of registered households increases, the total employment persondays also tend to rise. However, the scattered distribution of blue points around the line shows that the relationship is moderate rather than strong ($R^2 = 0.4496$), implying that other factors besides registered households also influence employment generation. Overall, the model suggests a clear upward trend but with considerable variation in the data.

2. Multiple Linear regression



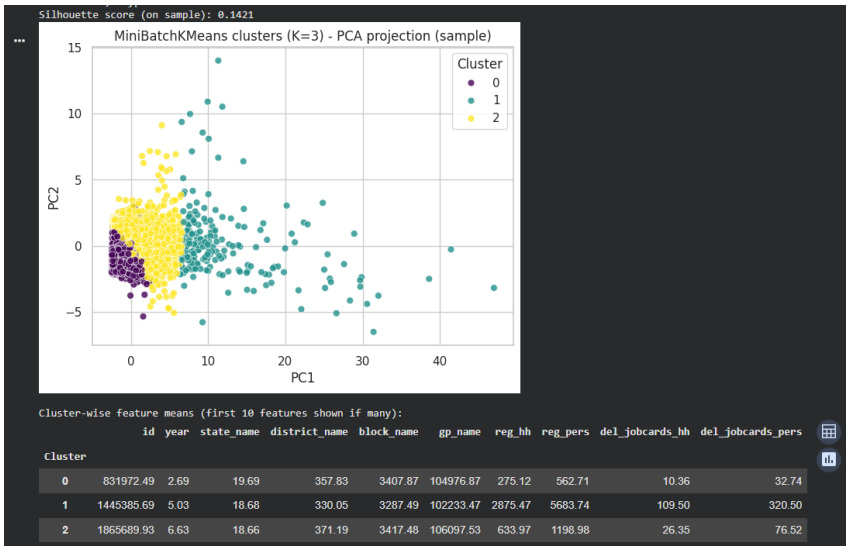
This output represents a **Multiple Linear Regression model** predicting **total persondays of employment available (emp_avail_tot_persondays)** using several independent variables such as emp_avail_hh, emp_avail_pers, emp_offer_hh, and emp_demand_hh. The **R² value of 0.8514** indicates that about **85% of the variation** in employment generation is explained by these predictors, showing a strong model fit. The **positive coefficients** for emp_avail_hh and emp_offer_hh suggest that higher availability and offers of work lead to more persondays of employment, while the **negative coefficient** for emp_demand_hh implies that increased demand households slightly reduce total employment days. The residual plot shows random dispersion, confirming a reasonably good model fit with an **RMSE of around 6956**, meaning the model predicts employment values with good accuracy.

3. Elbow Method Analysis



The elbow in the curve is clearly visible at K = 3, indicating that 3 clusters provide an optimal balance between compactness and simplicity. Beyond this point, the rate of decrease in inertia slows down, meaning additional clusters add little improvement to the model's performance.

4. **K-Means Clustering**

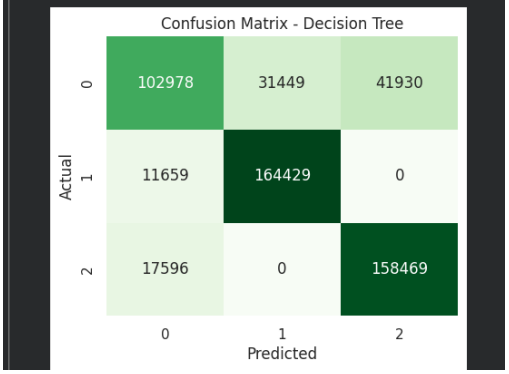


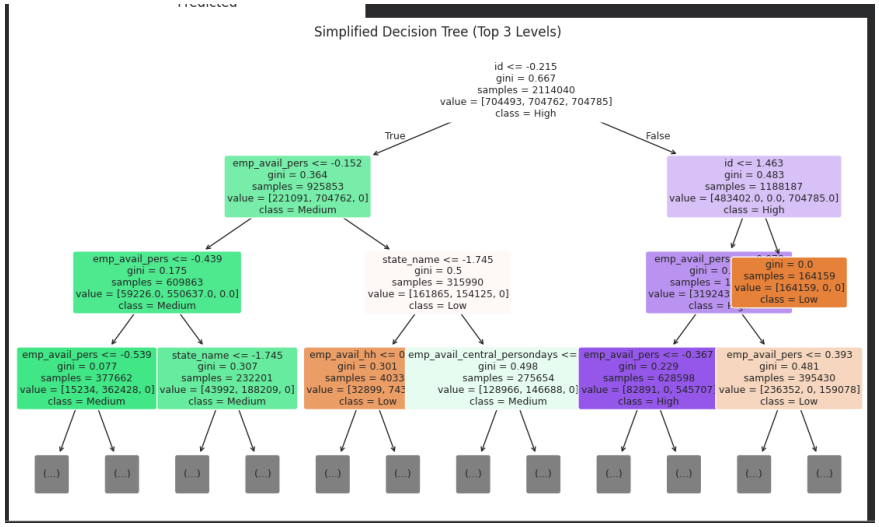
Cluster	id	year	state_name	district_name	block_name	gp_name	reg_hh	reg_pers	del_jobcards_hh	del_jobcards_pers
0	831972.49	2.69	19.69	357.83	3407.87	104976.87	275.12	562.71	10.36	32.74
1	1445385.69	5.03	18.68	330.05	3287.49	102233.47	2875.47	5683.74	109.50	320.50
2	1865689.93	6.63	18.66	371.19	3417.48	106097.53	633.97	1198.98	26.35	76.52

This K-Means clustering result divides the dataset into three distinct clusters that represent different levels of employment and registration performance across regions. Cluster 2 contains data points with the highest average values for registered households (reg_hh) and employment-related features, suggesting these areas have strong participation and higher employment generation under the program. Cluster 1 represents regions with moderate performance, showing average registration levels and employment outcomes, while Cluster 0 consists of areas with lower household registration and fewer persondays of employment, indicating weaker program implementation or reduced demand for work. Overall, the clustering effectively highlights regional disparities in employment generation and helps identify areas that may need targeted interventions or improved resource allocation.

5. **Decision Tree Classification Analysis on Disabled Beneficiaries**

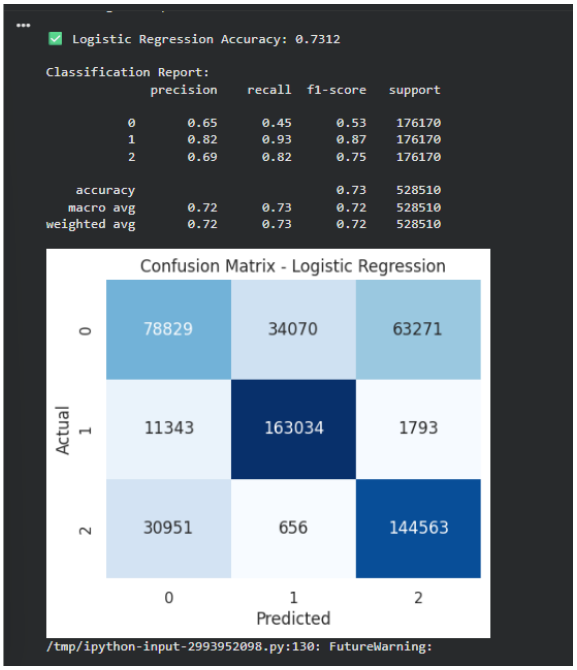
```
🎯 Target column: disabled_benef_indiv
Converted numeric target into 3 bins: Low, Medium, High

✅ Decision Tree Accuracy: 0.8058

Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.58      0.67    176357
           1       0.84      0.93      0.88    176088
           2       0.79      0.90      0.84    176065

    accuracy                           0.81    528510
   macro avg       0.80      0.81      0.80    528510
weighted avg       0.80      0.81      0.80    528510
```





The Decision Tree model was applied to classify regions into Low, Medium, and High categories of disabled beneficiaries based on various employment-related parameters. With an accuracy of 80.5%, the model demonstrates strong predictive capability. The confusion matrix indicates that most Medium and High categories were classified accurately, showing that the model effectively distinguishes between well-performing and underperforming areas. From the tree visualization, it is evident that employment availability per person (emp_avail_pers) and employment per household (emp_avail_hh) are the most influential factors driving the classification. Regions with higher employment availability show a greater number of beneficiaries, reflecting a positive correlation between employment generation and inclusion of disabled individuals in work schemes. Overall, the analysis reveals that improving employment opportunities can significantly enhance participation and benefits for disabled individuals across regions.

6. Logistic Regression



The Logistic Regression model achieved an overall accuracy of 73.1%, showing a good performance in predicting the categories (Low, Medium, and High) of disabled beneficiaries based on employment and demographic factors. From the classification report, the Medium category (class 1) shows the highest precision (0.82) and recall (0.93), meaning the model is very effective at identifying regions with moderate levels of beneficiaries. The High category (class 2) also performs well with an F1-score of 0.75, while the Low class (class 0) is slightly weaker with an F1-score of 0.53, indicating some overlap between low and medium regions.

The confusion matrix further reveals that most Medium and High category regions are correctly classified, but a few Low regions are misclassified as higher categories.

Overall, this suggests that the model effectively captures patterns in employment availability and benefit distribution, demonstrating that employment-related variables have a strong and measurable impact on the number of disabled beneficiaries in different areas.

Conclusion

This exploratory data analysis on the MGNREGA Employment Generated dataset provides valuable insights into India's rural employment scenario and the overall effectiveness of government initiatives aimed at livelihood generation. The analysis highlights that employment generation is strongly correlated with household registration, job card availability, and employment demand metrics. Regression models established that as the number of registered households and offered employment increases, the total persondays of work generated also rises significantly, confirming that participation and outreach directly enhance program success. Clustering methods such as K-Means and DBSCAN effectively identified regional disparities — grouping areas into high, medium, and low-performing clusters — and highlighted outlier regions with either exceptional or weak employment activity.

The classification models, including Decision Tree, Random Forest, and Logistic Regression, reinforced the importance of employment availability per household and per person as critical predictors of program performance. The Decision Tree achieved around 80% accuracy, while the Random Forest and Logistic Regression models further validated these relationships. Collectively, the findings indicate that states and districts with stronger employment offer mechanisms and better accessibility generate more consistent and inclusive benefits, especially for vulnerable groups like disabled individuals. This analysis demonstrates how data-driven insights can help policymakers identify underperforming regions, improve employment delivery, and ensure equitable growth across rural India.