DATA VISUALIZATION
(PMC-434)

PROJECT-3
(India Travel Insights: Best Daytime Prediction and Destination Clustering)
MSC Mathematics and Computing, IV Sem

SUBMITTED BY: MUDITA SHARMA (302303008)
SUBMITTED TO: DR. KAVITA

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, PATIALA

INDEX

1. Introduction

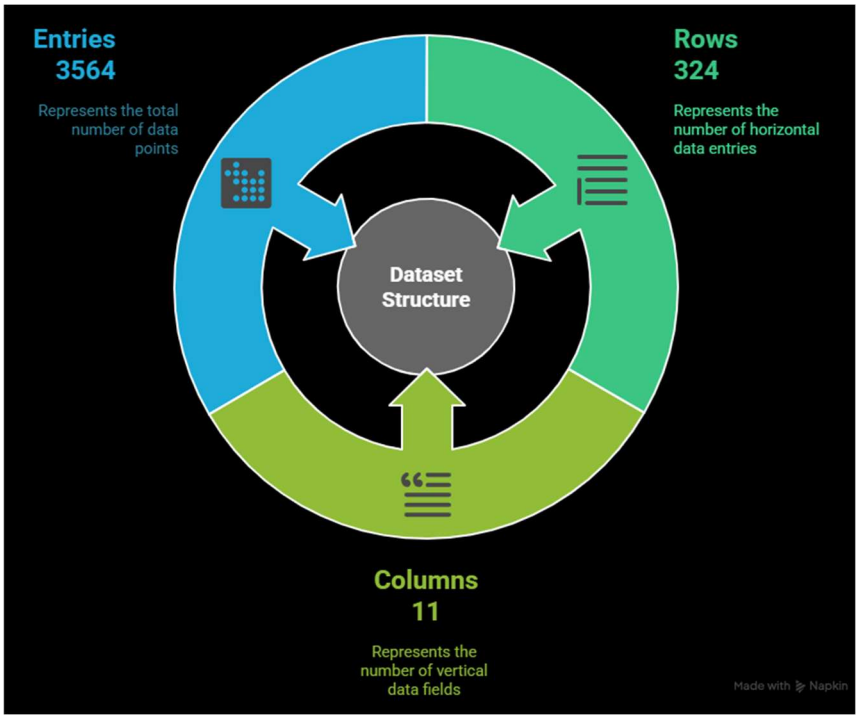
Every journey through India is not a vacation, it's an education- Shashi Tharoor

Welcome to "India Travel Insights: Best Daytime Prediction and Destination Clustering", a smart travel guide made using data. This project explores the beauty of India from the calm mountains in the north to the colourful palaces of Rajasthan, busy cities and peaceful beaches in the south.

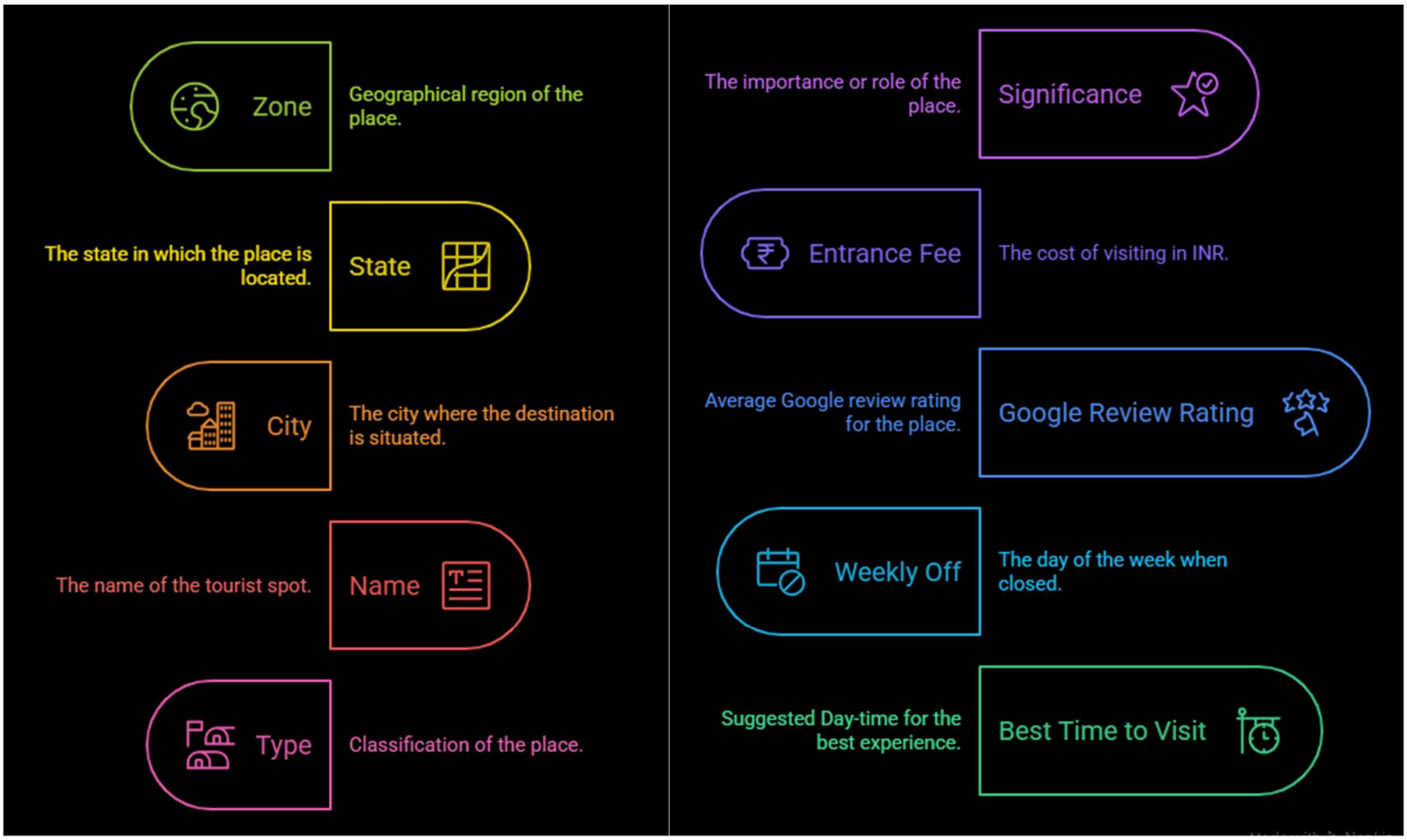
The project has two exciting goals:

1. Predicting best Daytime to visit destinations using models like Logistic Regression, Decision Tree, SVM and Random Forest.
2. Group similar destinations together through clustering techniques like K-means, DBSCAN, Hierarchical Clustering and KNN to reveal interesting travel patterns.

2. Dataset Overview



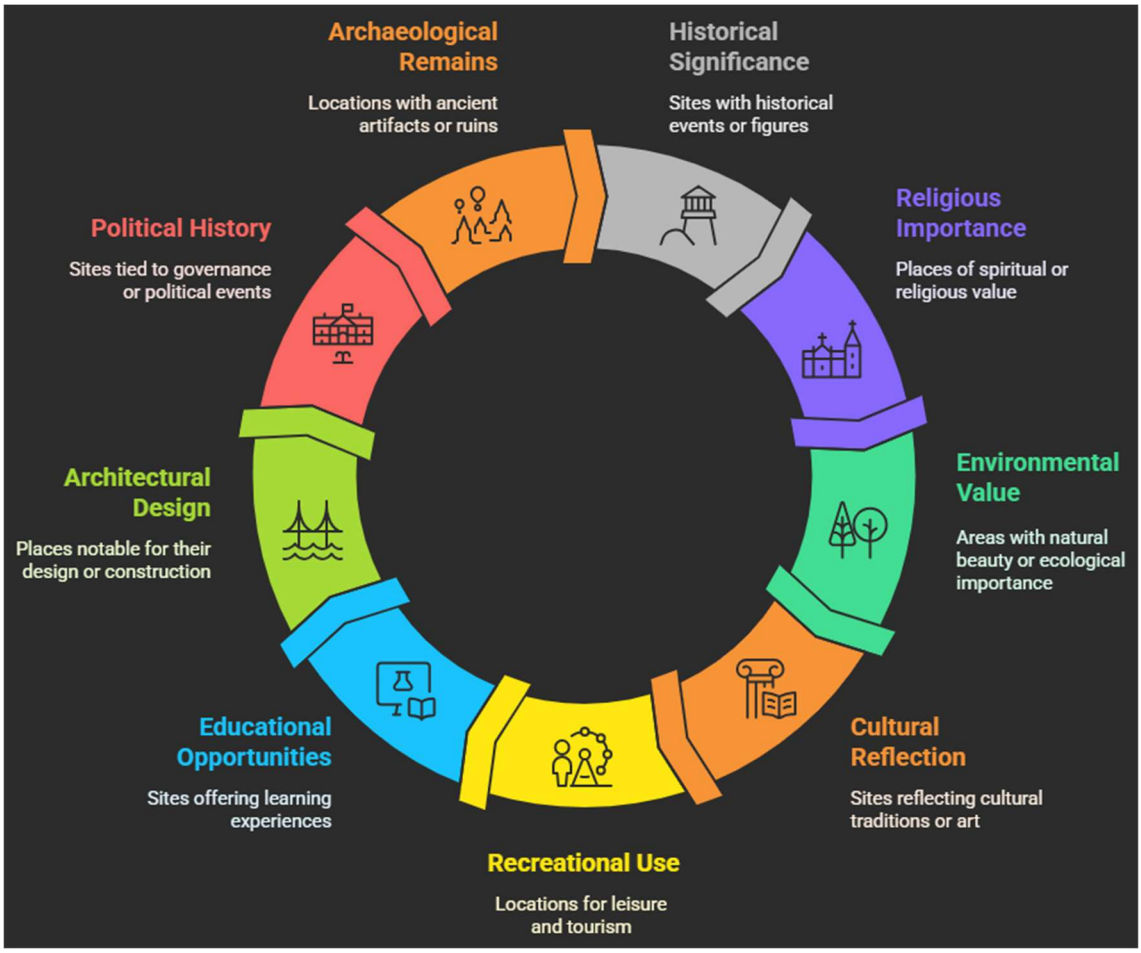
- **Attributes**



- **Type of Categories**

Category	Description
Temple	A place of worship
Church	A Christian place of worship
Mosque	An Islamic place of worship
War Memorial	Site commemorating military events
Natural Park	Protected natural area
Museum	Institution displaying artifacts, art
Monument	Structure commemorating an event
Beach	Coastal area known for sand
Fort	Fortified military structure
Palace	Grand residence of royalty
Garden	Landscaped area for recreation
Archaeological Site	Location with historical ruins
Historic Building	Significant building with historical value
Waterfall	Natural feature where water flows
Mountain	Significant natural elevation
Lake	Large body of water
Bridge	Notable structure with importance
Market	Cultural or historical marketplace
Zoo	Facility housing animals
Amusement Park	Recreational area with rides

- **Cycle of Significance**

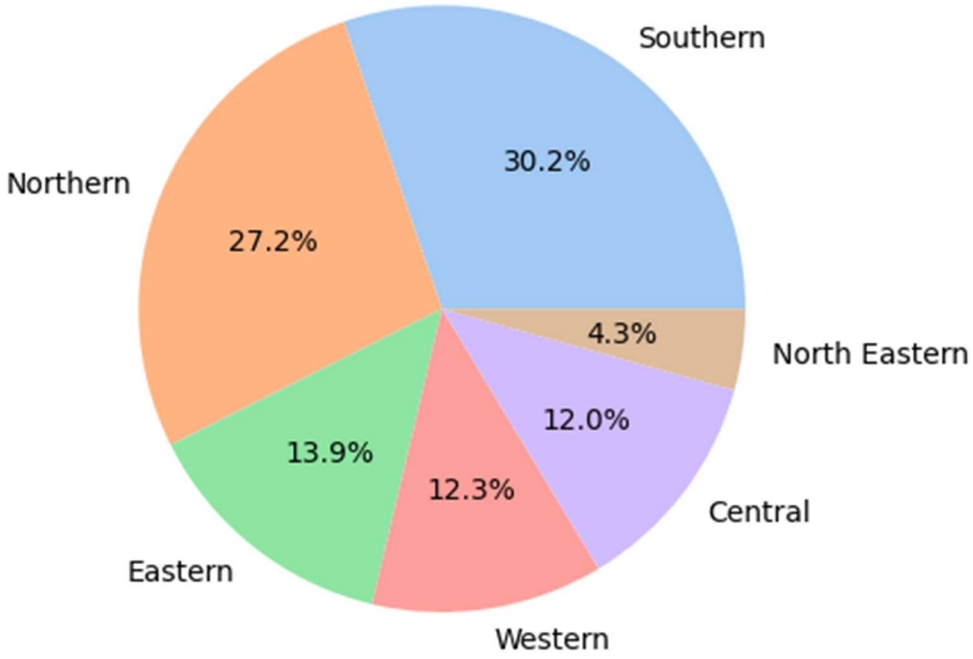


#	Column	Non-Null Count	Dtype
0	Zone	324 non-null	object
1	State	324 non-null	object
2	City	324 non-null	object
3	Name	324 non-null	object
4	Type	324 non-null	object
5	Significance	324 non-null	object
6	Entrance Fee in INR	324 non-null	int64
7	Google review rating	324 non-null	float64
8	Weekly Off	324 non-null	object
9	Best DayTime to visit	324 non-null	object

Zone	State	City	Name	Type	Significance	Entrance Fee in INR	Google review rating	Weekly Off	Best DayTime to visit
Northern	Delhi	Delhi	India Gate	War Memorial	Historical	0	4.6	All day open	Evening
Northern	Delhi	Delhi	Humayun's Tomb	Monument	Historical	30	4.5	All day open	Afternoon
Northern	Delhi	Delhi	Akshardham Temple	Temple	Religious	60	4.6	All day open	Afternoon
Northern	Delhi	Delhi	Waste to Wonder Park	Amusement Park	Recreational	50	4.1	Monday	Evening
Northern	Delhi	Delhi	Jantar Mantar	Archaeological Site	Archaeological	15	4.2	All day open	Morning

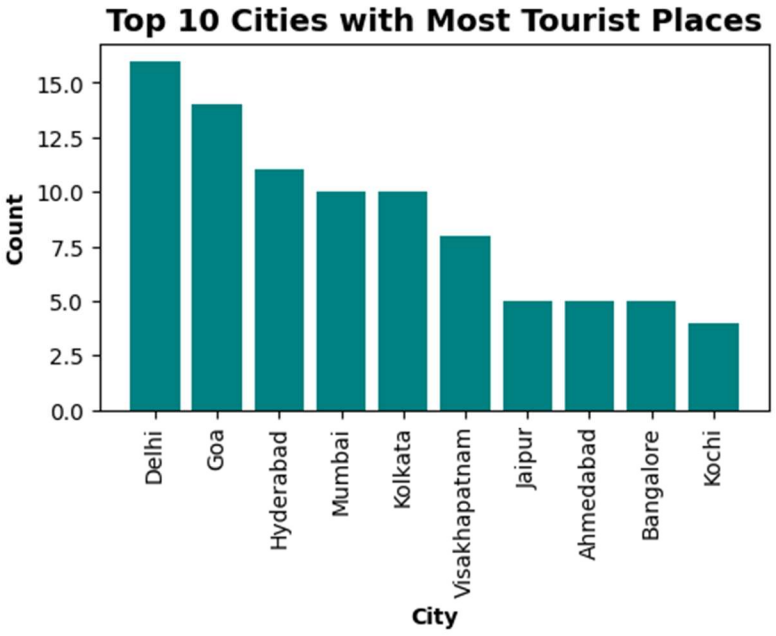
3.Data Visualization

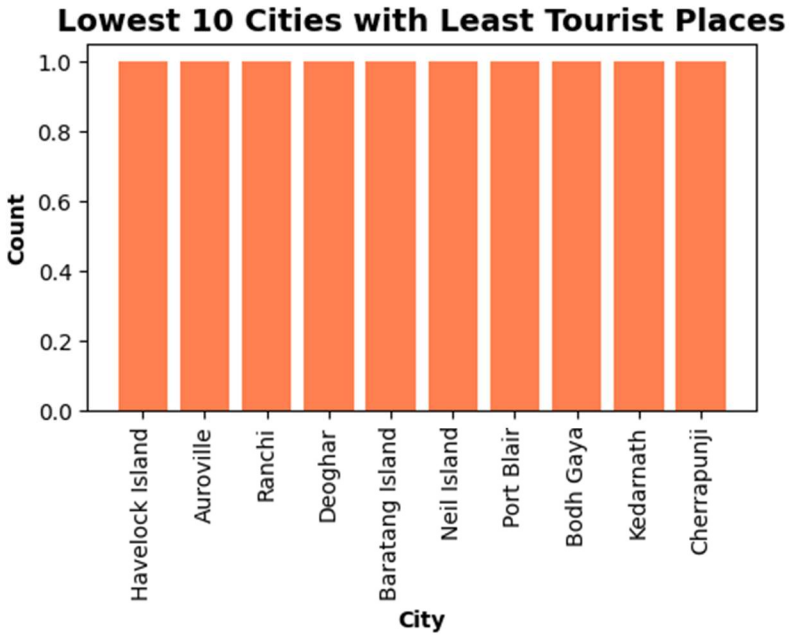
- **Pie- chart of tourist places by zone**



- First, I uses the value_counts() function to count how many places are in each zone of dataset (df). The plt.pie() function then draws the chart, with each zone labeled and its percentage.
- The pie chart shows that the Southern zone has the highest share of tourist places at 30.2%, followed by the Northern zone at 27.2%. The Eastern zone accounts for 13.9%, the Western zone 12.3%, the Central zone 12.0% and the North Eastern zone has the smallest share at 4.3%.
- This indicates that Southern and Northern zones are the most popular for tourism, while North Eastern has the least.

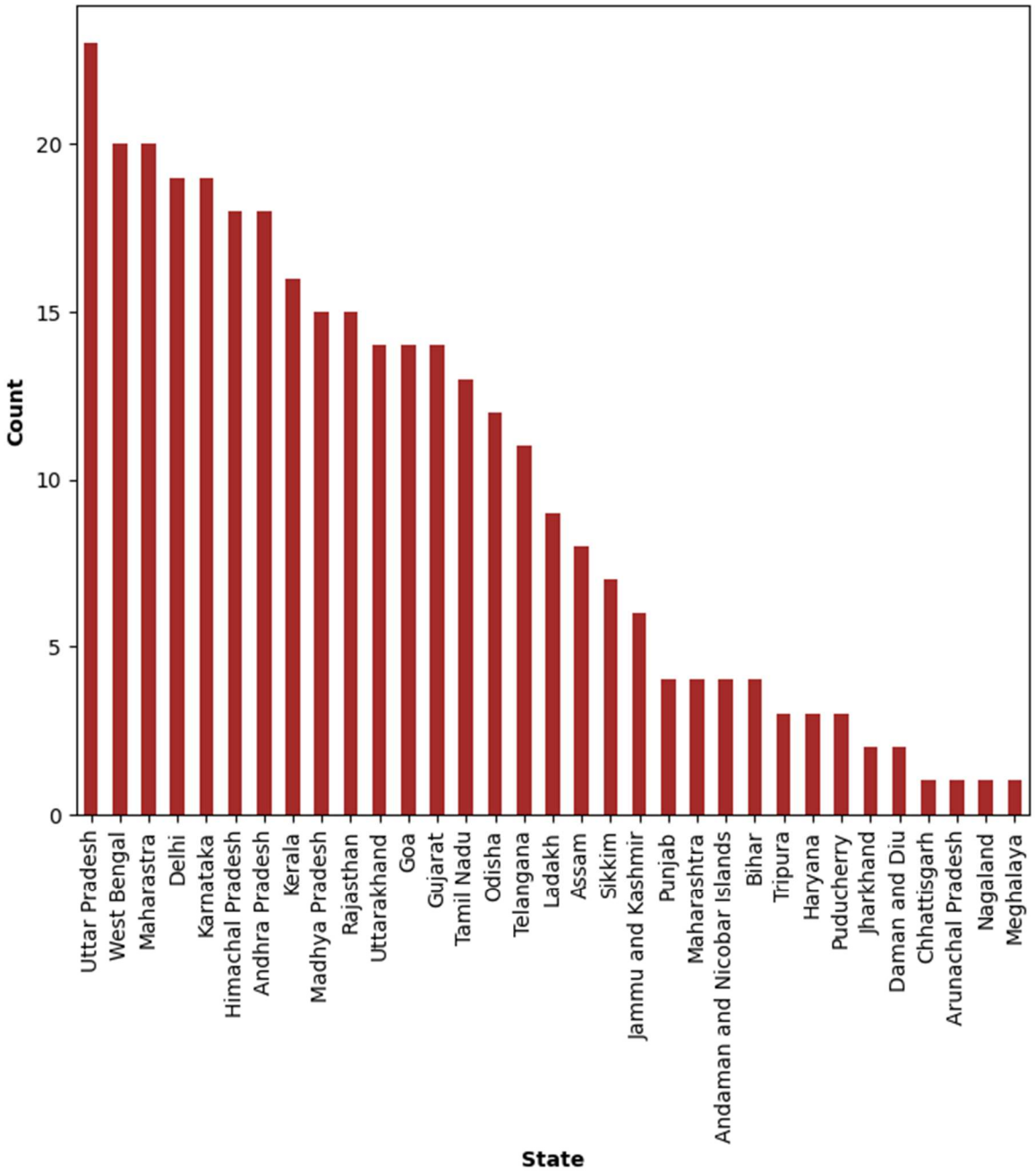
• Unique cities in the dataset





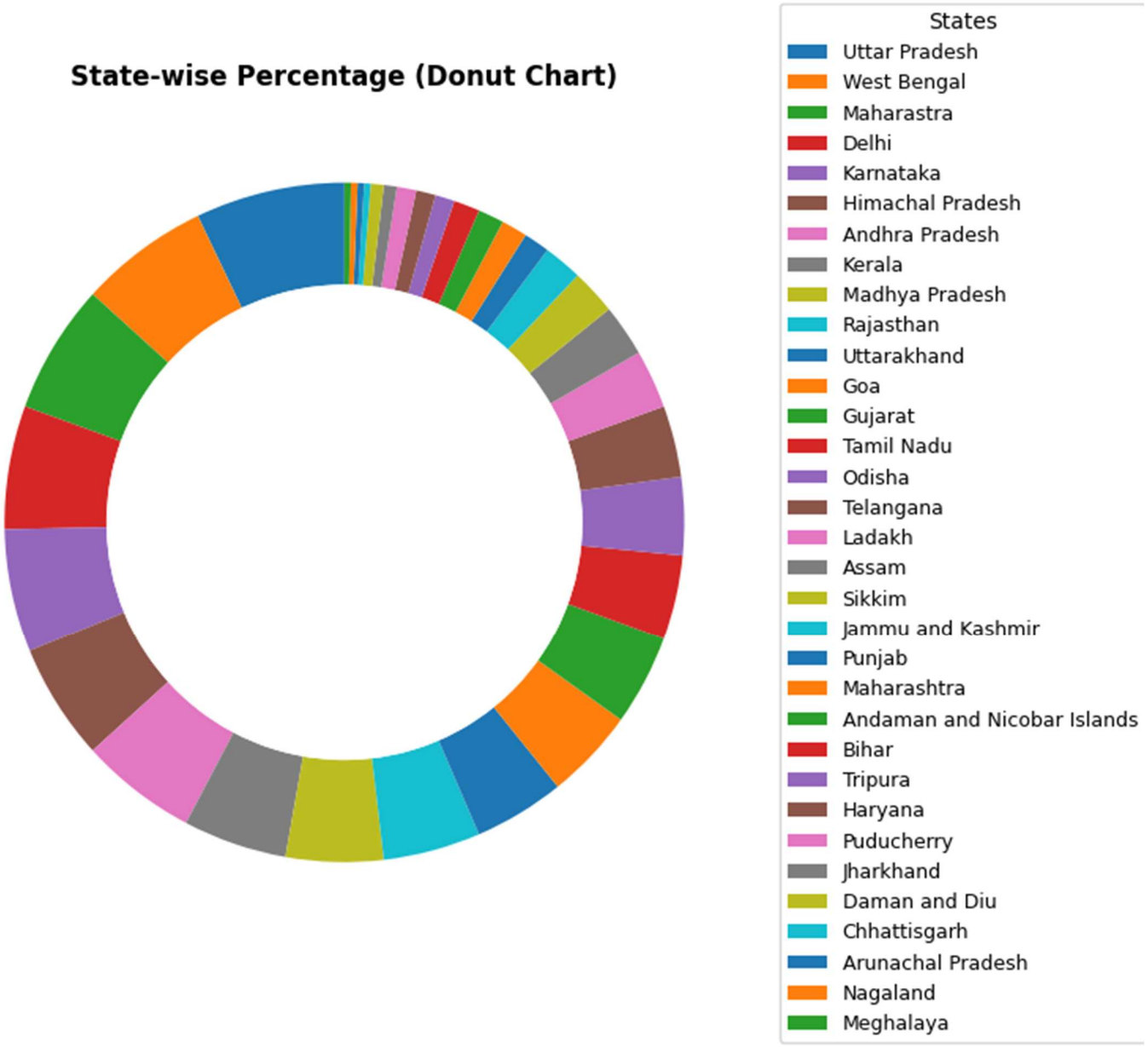
- Dataset contains 214 unique cities.
- Firstly, counts the number of tourist places per city in a dataset (df) using value_counts(). The head(10) picks the top 10 cities with the most places and tail(10) picks the bottom 10 with the least. Two charts are set up ,the left chart shows the top 10 cities with teal bars and the right chart shows the bottom 10 with coral bars.
- Delhi leads with the highest number of tourist places (around 15) followed by Goa and Hyderabad (around 12-13). Other top cities like Mumbai, Kolkata and Jaipur have 7-10 places. On the other hand, the lowest 10 cities (Havelock Island, Auroville, Ranchi) each have 0.2 to 1 place, showing they have very few tourist spots.

- **Distribution of States as per Tourist Places (bar chart)**



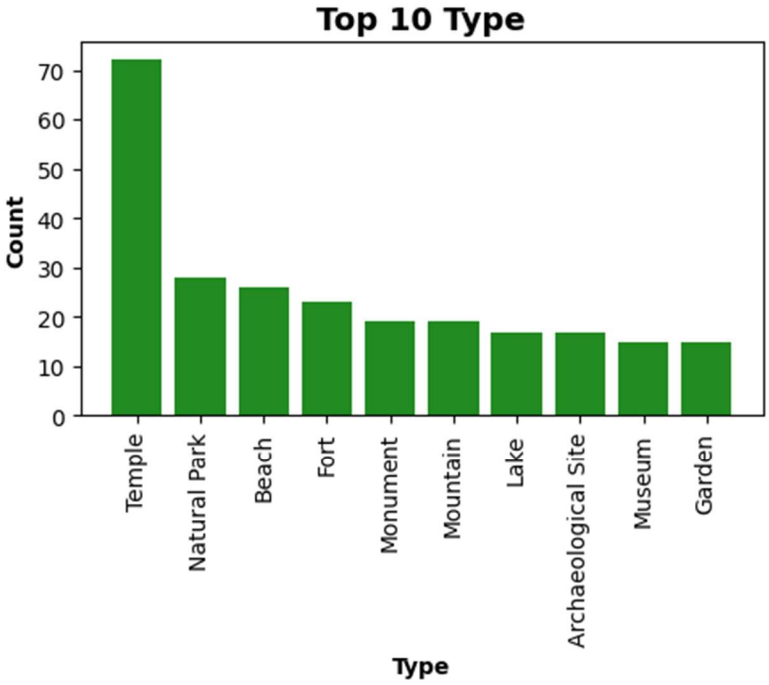
- Dataset contains 33 states (i.e, Lakshadweep-Mizoram-Manipur aren't present).
- The chart shown is a bar graph in brown, showing the count of places per state.
- It shows Uttar Pradesh has the most tourist places (around 25) followed by West Bengal and Maharashtra (around 20). Many states like Nagaland and Arunachal Pradesh have very few (less than 5).

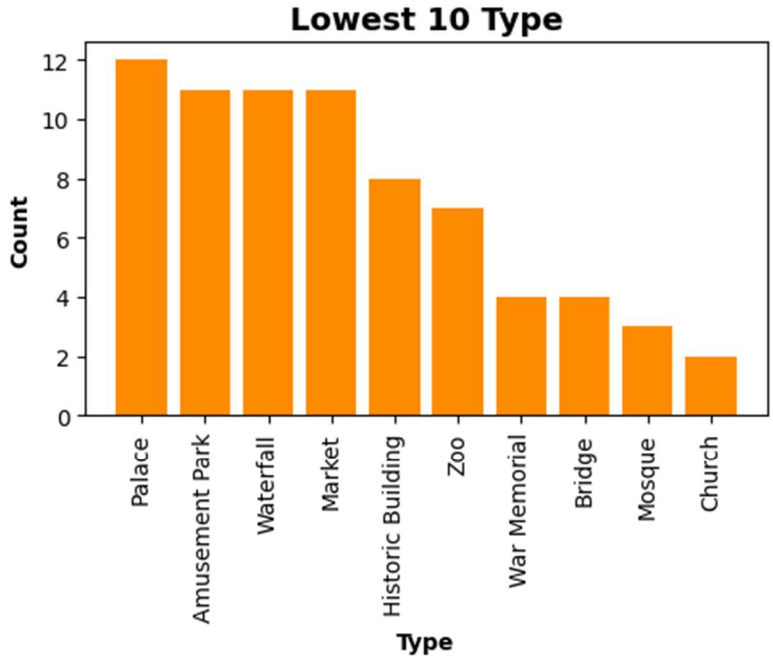
- **Distribution of States as per Tourist Places (donut chart)**



- The chart shown is a donut chart (pie() with a hollow center) showing the percentage of places per state with a label on the side listing all states.
- The donut chart reflects the same trend in percentages with Uttar Pradesh having the largest slice while smaller states have tiny slices ,indicating they contribute the least to tourist places.

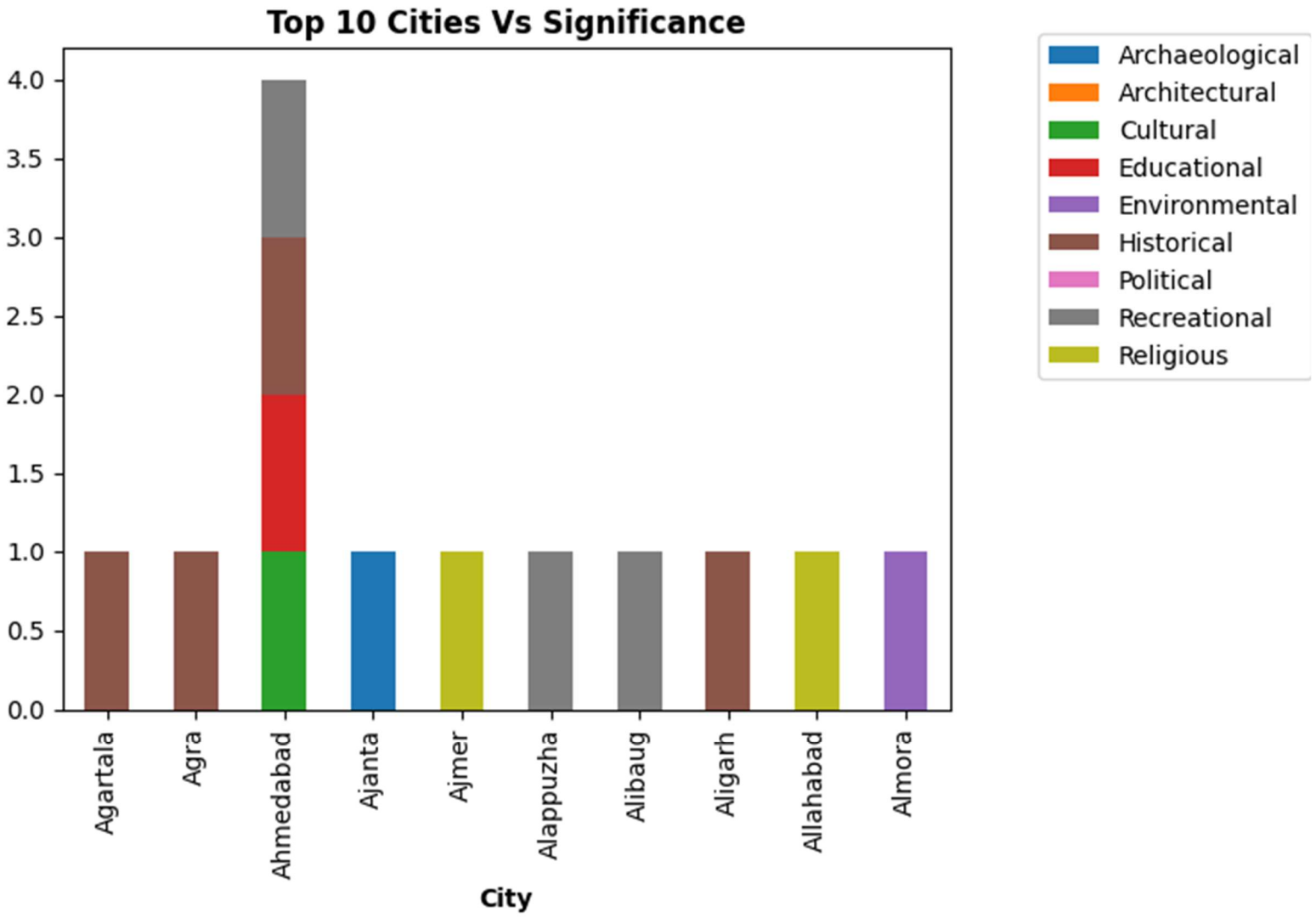
• Top 10 & Lowest 10 Type of Places





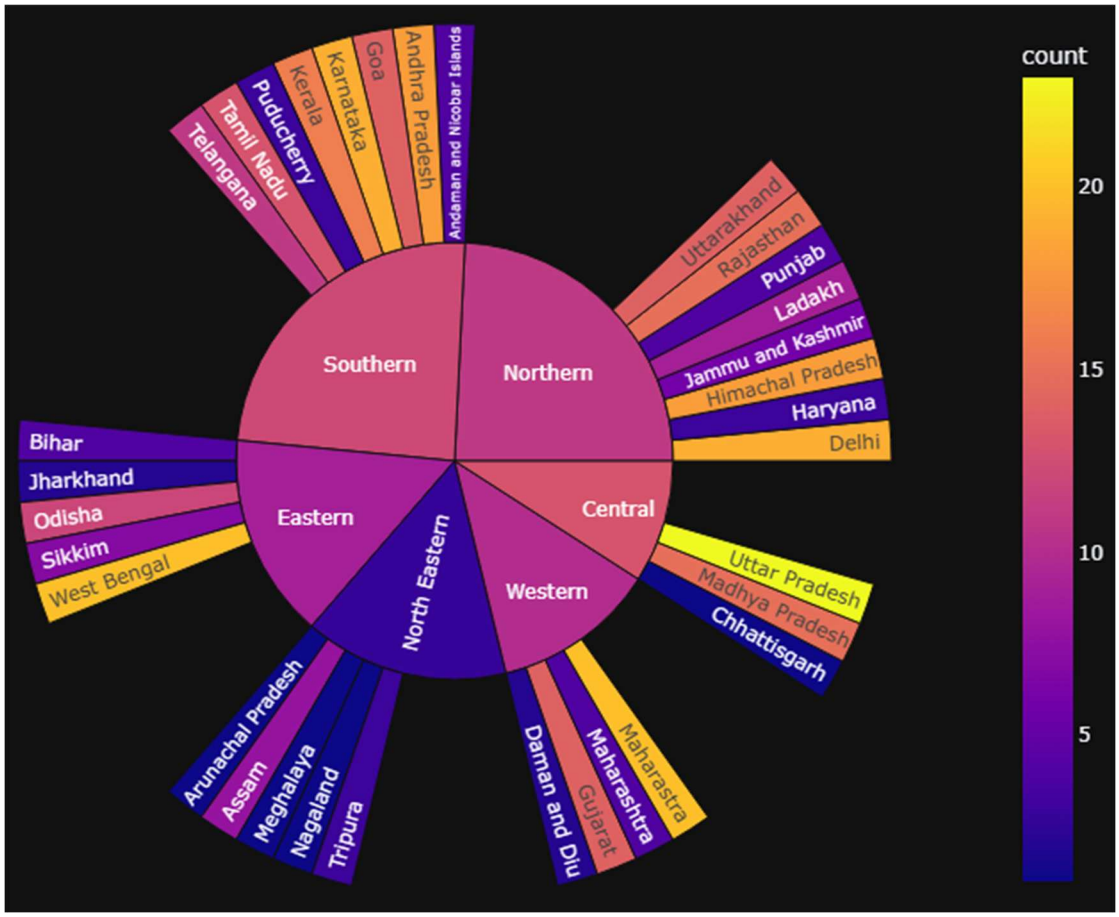
- The head(10) selects the top 10 types with the most places and tail(10) selects the bottom 10 with the least. The left chart shows the top 10 types with green bars and the right chart shows the bottom 10 with orange bars.
- The top 10 chart shows Temples have the highest count (around 70) followed by Natural Park and Beach (around 30-40). Other types like Fort, Monument and Lake have 20-30 each. The lowest 10 chart shows Market, Amusement Park and Waterfall each have around 8-10 while types like Bridge, Mosque and Church have 2-4, indicating they have the fewest tourist places.

- **Top 10 Cities Vs Significance**



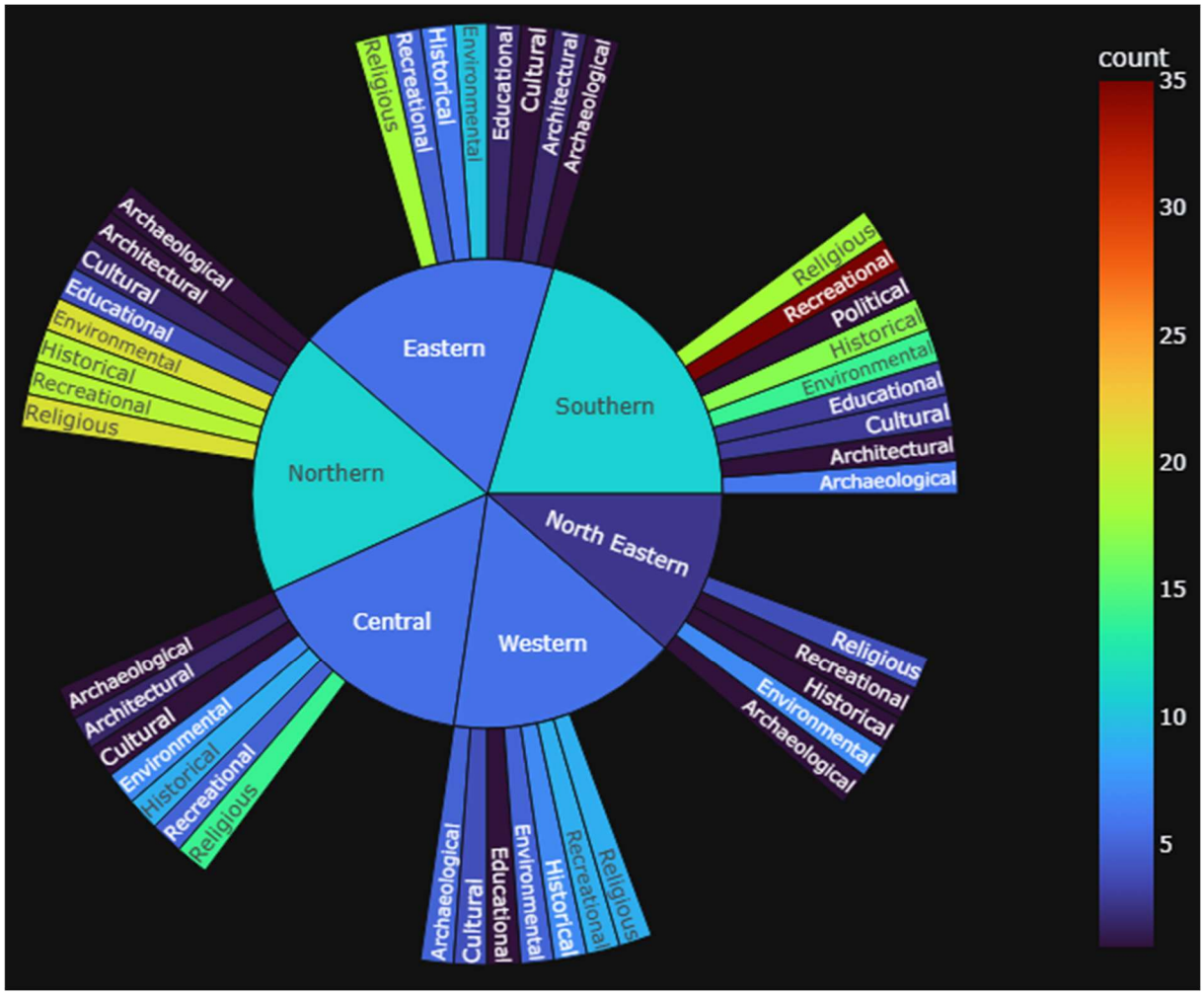
- This is a stacked bar chart, in which data is grouped by "City" and "Significance“, to count how many times each type appears.
- Ahmedabad has the highest total significance types (4), Cities like Agra, Agartala, Ajanta, Ajmer, Alappuzha, etc., have only one type of significance.

- **Which Zone & State has the Most Tourist Attractions (Sunburst chart)**



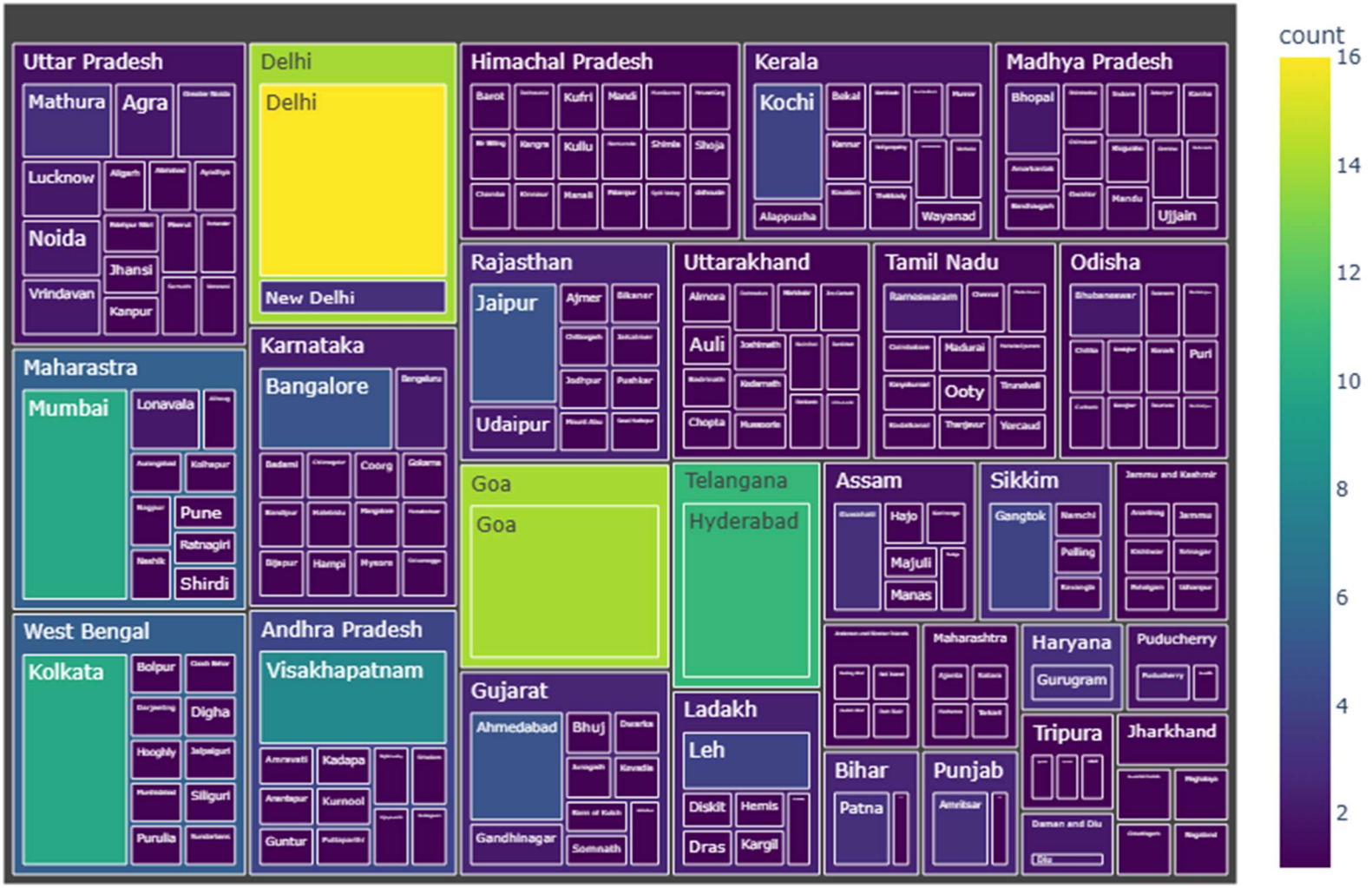
- Grouped the data by Zone and then State, and counting how many tourist attractions each has. Plotted a Sunburst chart where: Inner circle = Zone , Outer slices = States inside that zone. Color intensity = number of tourist attractions (yellow = more, purple = less).
- Southern Zone has the highest number of tourist attractions followed by northern Zone . In Central Zone, Uttar Pradesh is a major highlight

- **Which Zone & Significance has the Most Tourist Attractions (Sunburst chart)**



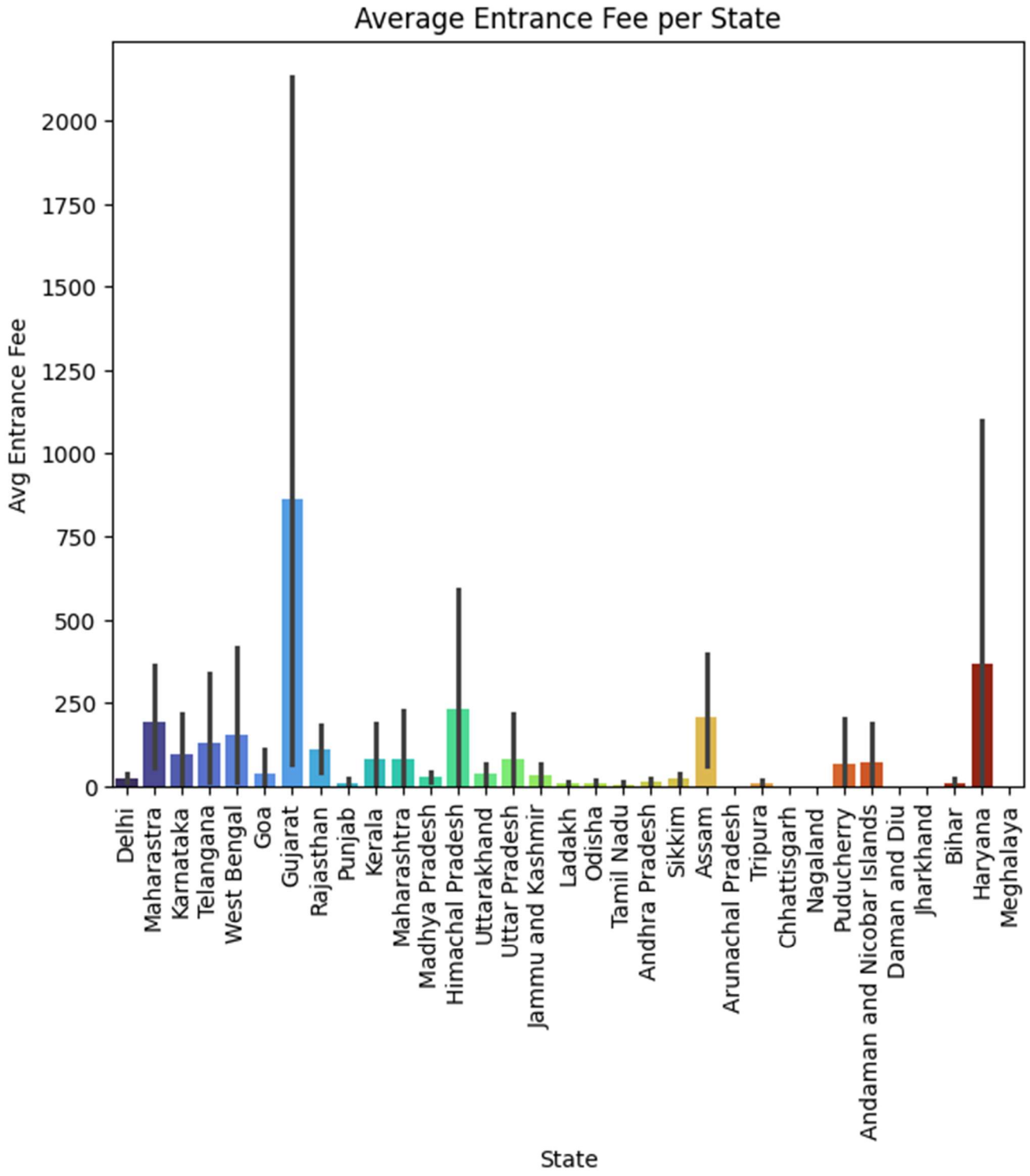
- Grouped the data by Zone and then Significance, and counting how many tourist attractions each has. Plotted a Sunburst chart where: Inner circle = Zone , Outer slices = Significance inside that zone. Color intensity = number of tourist attractions (yellow/red = more, purple/blue = less).
- Northern Zone and Southern Zone have the widest slices, meaning they have more tourist places.

- **TreeMap of Tourist Places by State and City**



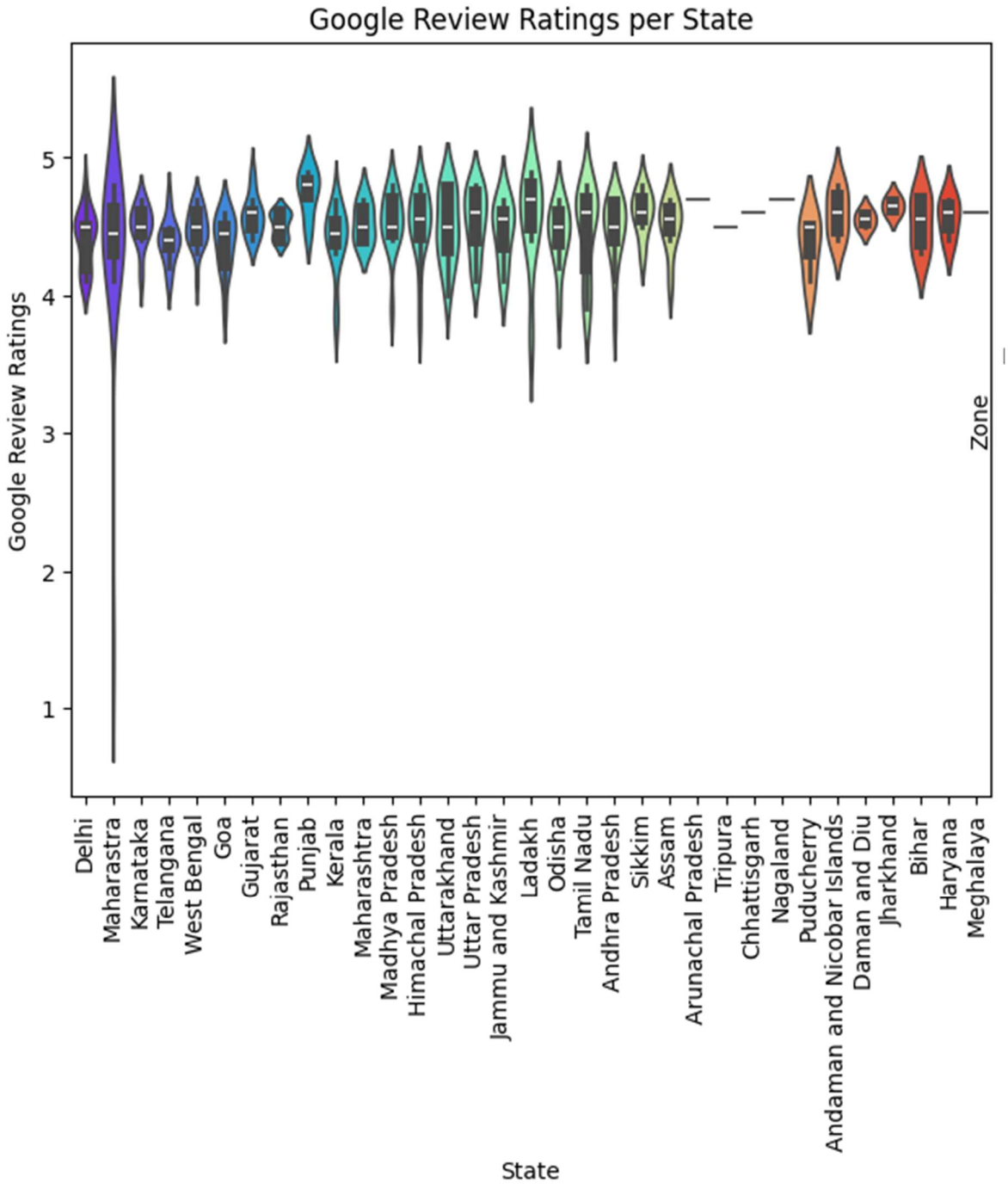
- Grouping by State and City, then counting tourist places in each.
- In TreeMap, Big rectangles = states, Nested rectangles = cities within states. Color & size = number of tourist places (brighter = more)
- Top Cities by Tourist Count: Delhi and New Delhi together form the largest bright yellow box(highest count = 16), Goa (state & city same) is next (count = 15), Mumbai (Maharashtra) and Hyderabad (Telangana) are also among the top
- States with Highest Spread: Uttar Pradesh has a dense grid of cities each with multiple

- **Entrance Fee analysis per State (Bar Plot)**



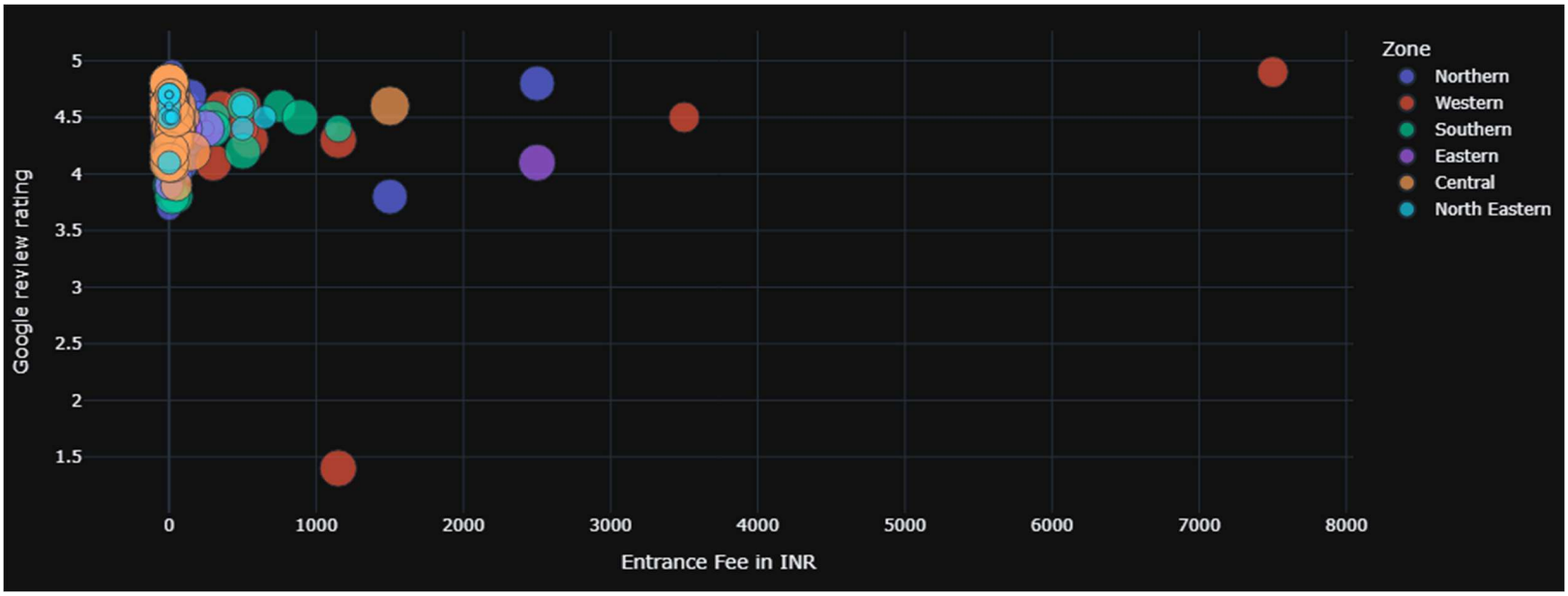
Gujarat has the highest average entrance fee. Followed by Haryana, Himachal Pradesh. Many states like Bihar, Tripura, Chhattisgarh have low average entrance fees

- **Google Review Rating Analysis (Violin-plot)**



- The plot resembles a violin, the width of the "violin" at any point represents the density of data points at that Google Review Rating value. Maharashtra and Karnataka have wide sections around 4-5.
- Violin plots include a small box plot inside the violin. In this plot, the thicker central part for states like West Bengal and Gujarat spans from about 3.5 to 4.5, showing the range where most ratings fall.

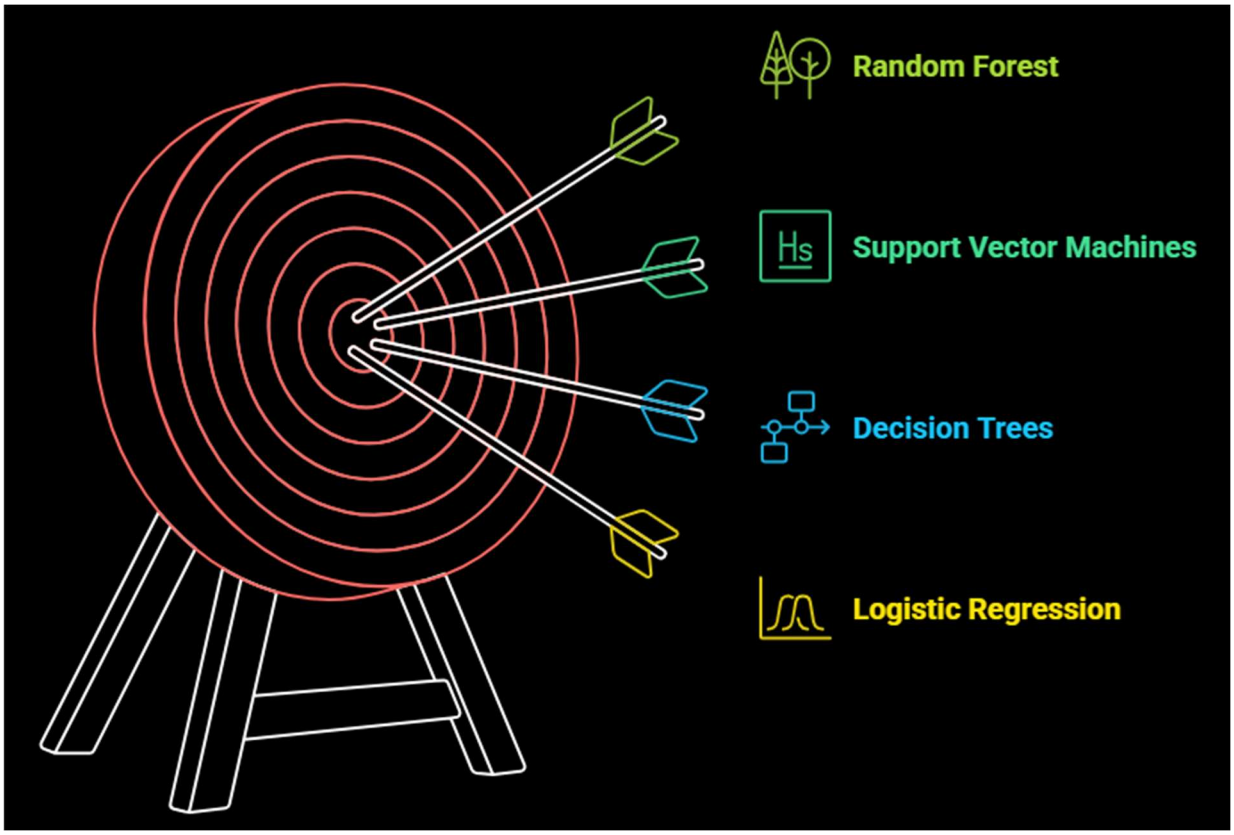
- **Bubble Chart: Entrance Fee vs Google Review (Bubble = No. of Places in State)**



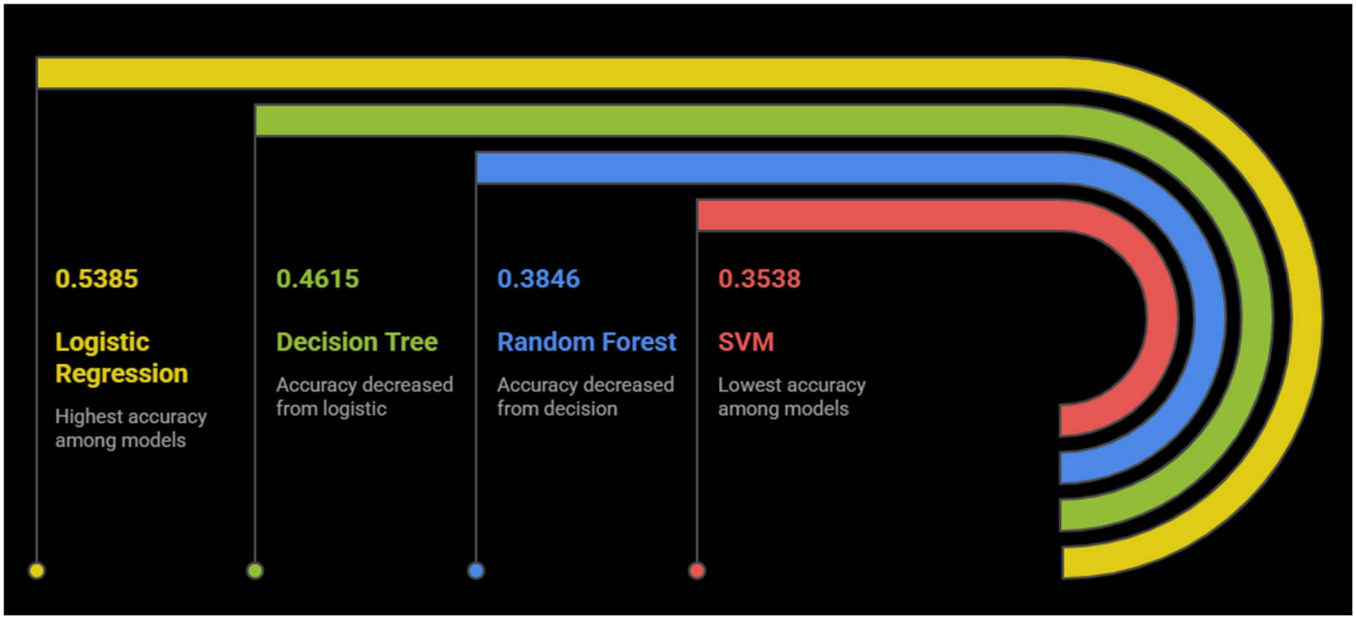
Majority of bubbles are grouped between INR 0 to INR 500, These places mostly have high ratings (around 4.2 to 4.8).This suggests low-cost tourist places are still delivering good visitor experiences. Only one noticeable bubble at 1.5 rating with entrance fee around ₹1200. Very large bubbles at low entrance fees and good ratings, These are states with a high number of tourist spots. Regionally, Northern and Western zones display a wider range of entrance fees, while the Southern and North Eastern zones maintain affordability and strong ratings.

- x-axis = Entrance Fee, y-axis = Google Rating, bubble size = Number of places in that State, bubble color = Which Zone the place belongs to

4.Classification: Predict "Best DayTime to Visit"



- **Model Comparison (Accuracy)**

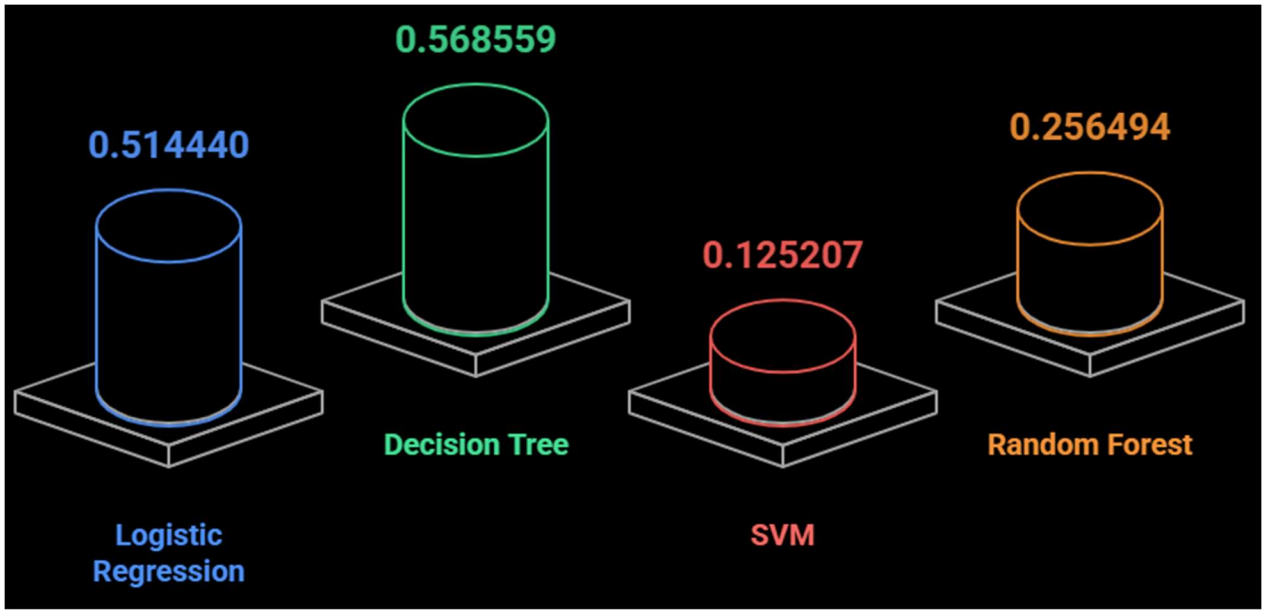


Logistic Regression is the best-performing model with 53.85% accuracy. The other models (Decision Tree, SVM and Random Forest) show accuracies below 50%, indicating they are less effective.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

TP (True Positives): Number of positive cases correctly predicted as positive.
TN (True Negatives): Number of negative cases correctly predicted as negative.
FP (False Positives): Number of negative cases incorrectly predicted as positive.
FN (False Negatives): Number of positive cases incorrectly predicted as negative.

- **Model Comparison (Precision)**

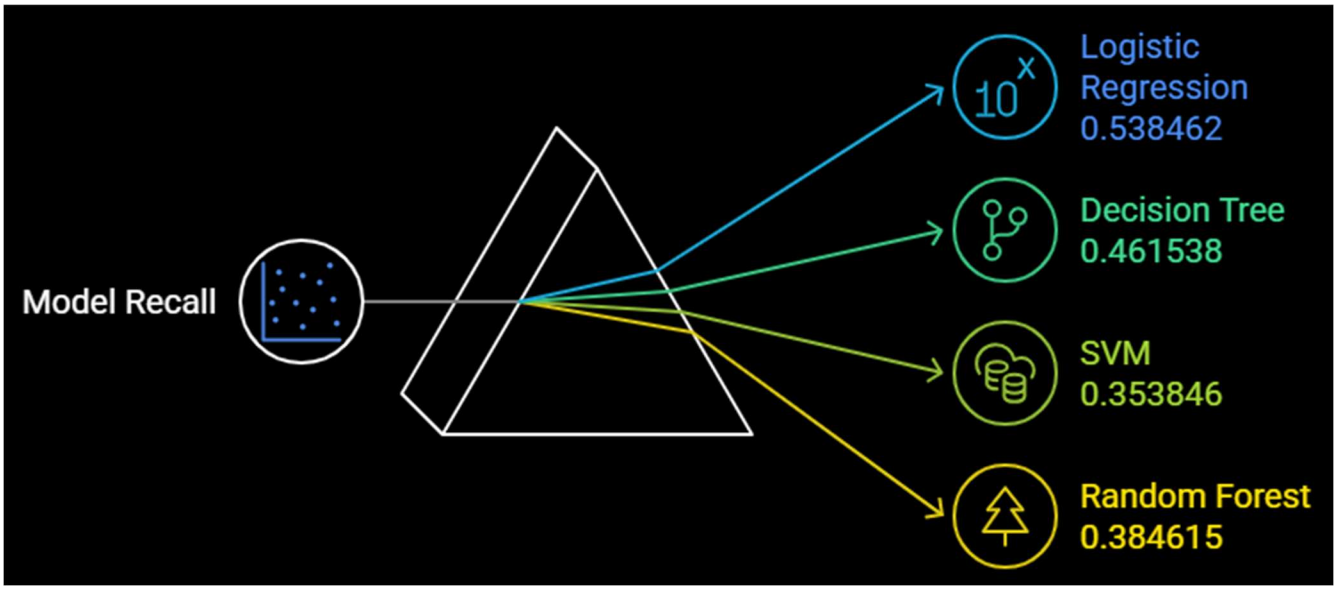


Precision measures the accuracy of positive predictions made by a classification model. It tells us of all the instances the model predicted as positive, how many were actually positive.
Decision Tree leads with 56.86% precision, it is the most reliable for correctly identifying positive cases among the models tested.

$$\text{Precision} = \frac{TP}{TP + FP}$$

TP (True Positives): Number of positive cases correctly predicted as positive.
FP (False Positives): Number of negative cases incorrectly predicted as positive.

- **Model Comparison (Recall)**



Recall also known as sensitivity or true positive rate measures the ability of a classification model to identify all relevant positive cases. It tells us of all the actual positive instances, how many did the model correctly predict as positive. Logistic Regression leads with 53.85% recall, it is the most effective at capturing actual positive cases among the models as it handles the positive class distribution better.

$$\text{Recall} = \frac{TP}{TP + FN}$$

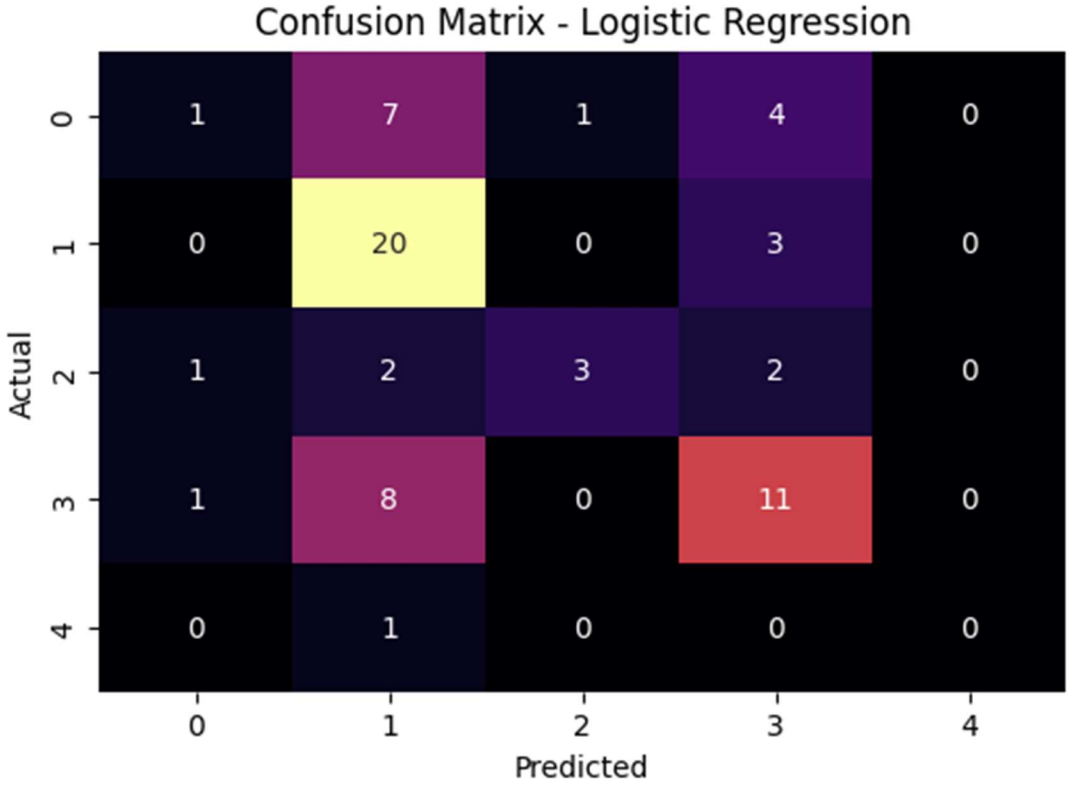
TP (True Positives): Number of positive cases correctly predicted as positive.

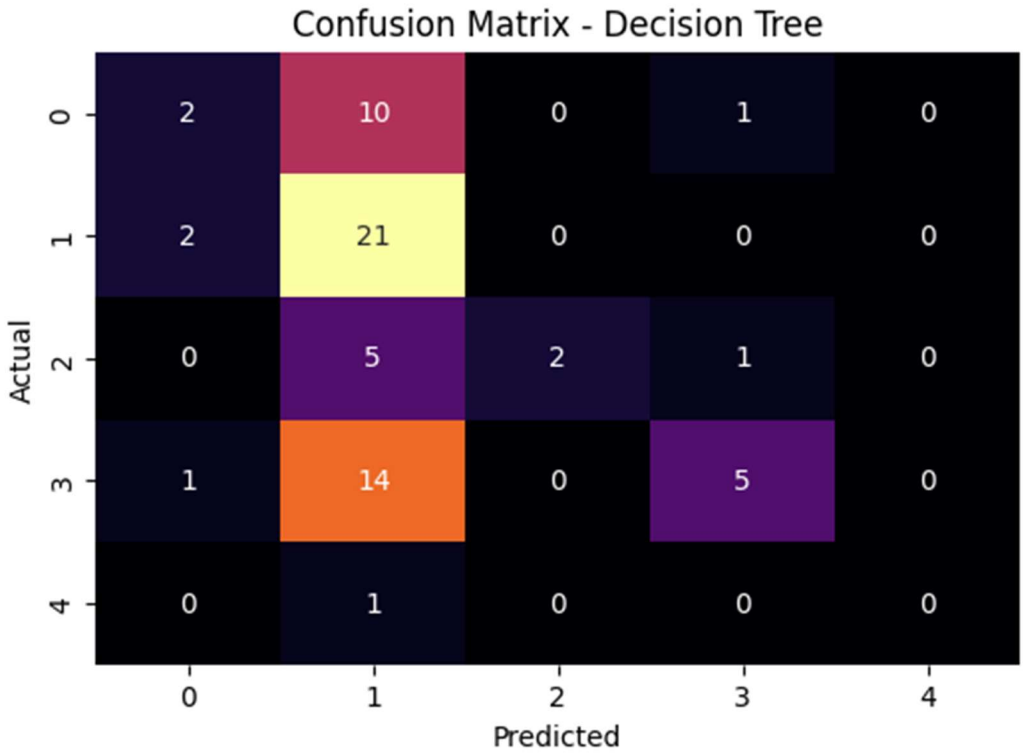
FN (False Negatives): Number of positive cases incorrectly predicted as negative.

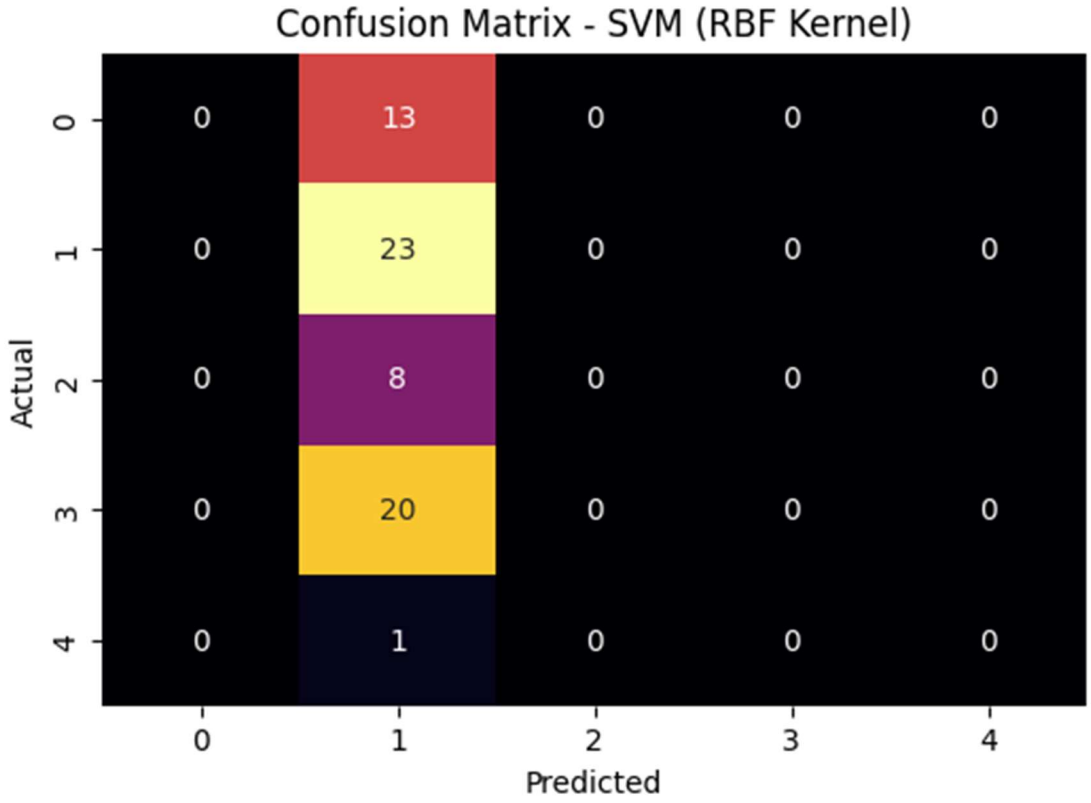
- **Model Comparison (F1-Score)**

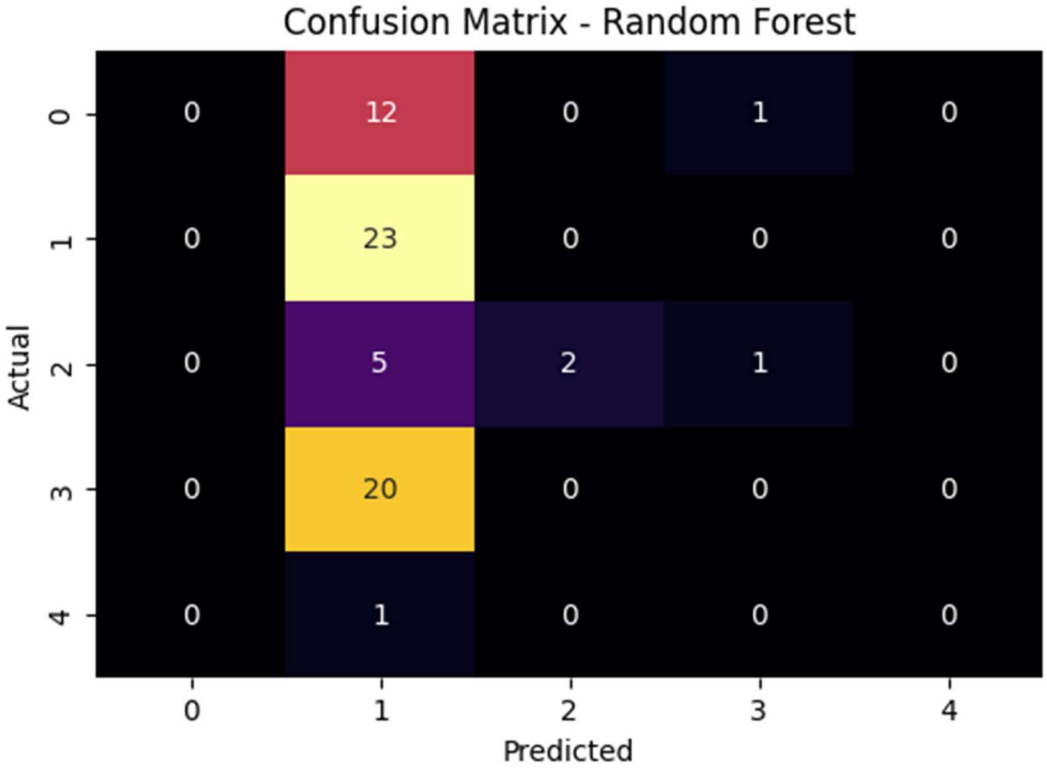
The F1-Score is the harmonic mean of precision and recall, providing a single metric that balances both the ability to correctly identify positive cases (recall) and the accuracy of positive predictions (precision). Logistic Regression leads with an F1-Score of 48.78%, making it the most effective model overall. This suggests it maintain the best balance between precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$


Confusion Matrix - Logistic Regression
Actual
Predicted
0
1
2
3
4
1 7 1 4 0
0 20 0 3 0
1 2 3 2 0
1 8 0 11 0
0 1 0 0 0


Confusion Matrix - Decision Tree
Actual
Predicted
0
1
2
3
4
2 10 0 1 0
2 21 0 0 0
0 5 2 1 0
1 14 0 5 0
0 1 0 0 0

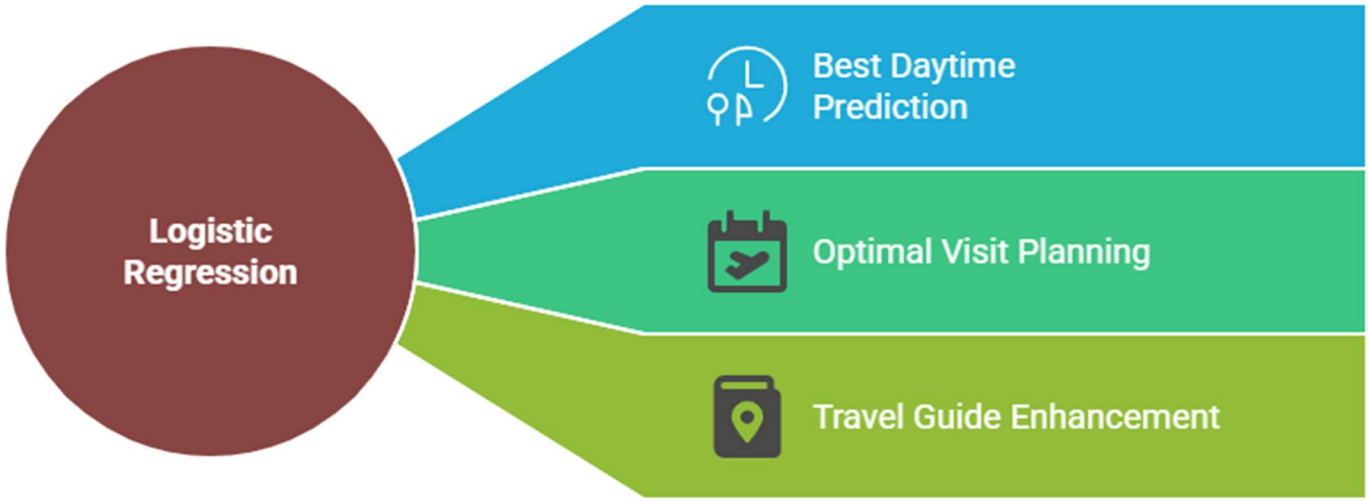

Confusion Matrix - SVM (RBF Kernel)
Actual
Predicted
0 13 0 0 0
0 23 0 0 0
0 8 0 0 0
0 20 0 0 0
0 1 0 0 0
0 1 2 3 4


Confusion Matrix - Random Forest
Actual
Predicted
0 12 0 1 0
0 23 0 0 0
0 5 2 1 0
0 20 0 0 0
0 1 0 0 0
0 1 2 3 4

• Choosing the best model

For a travel guide that helps pick the best time of day to visit places we need a model that:

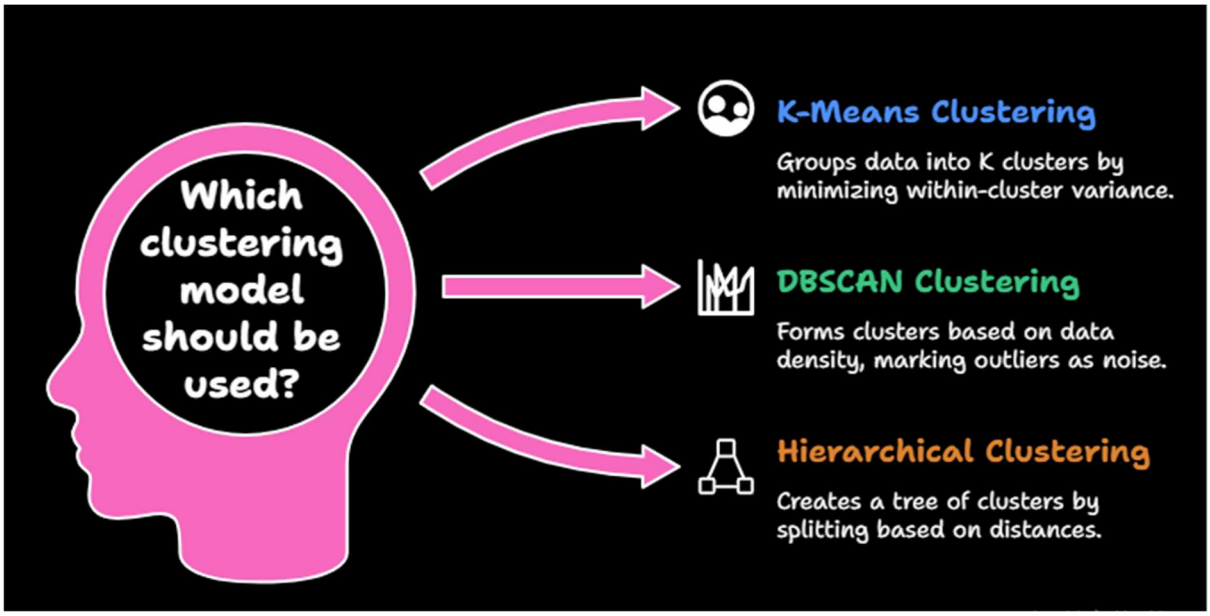
- Gets the most things right overall (high Accuracy).
- Correctly guesses the best times and doesn't miss too many good ones (a good mix of Precision and Recall as shown by the F1-Score).
- Works well for all time categories (like morning, afternoon) not just one.



- It has the highest Accuracy (0.538462), it's the most reliable overall.
- It has the highest F1-Score (0.487799), it balances guessing the right times and not missing good ones. This is super important for a travel guide where wrong suggestions can be a problem.
- Its confusion matrix shows it handles different times better than the other models, which is key for covering all daytimes.
- Even though its Precision (0.514440) isn't the top, it's still good and its Recall (0.538462) means it catches more of the real best times than the others.

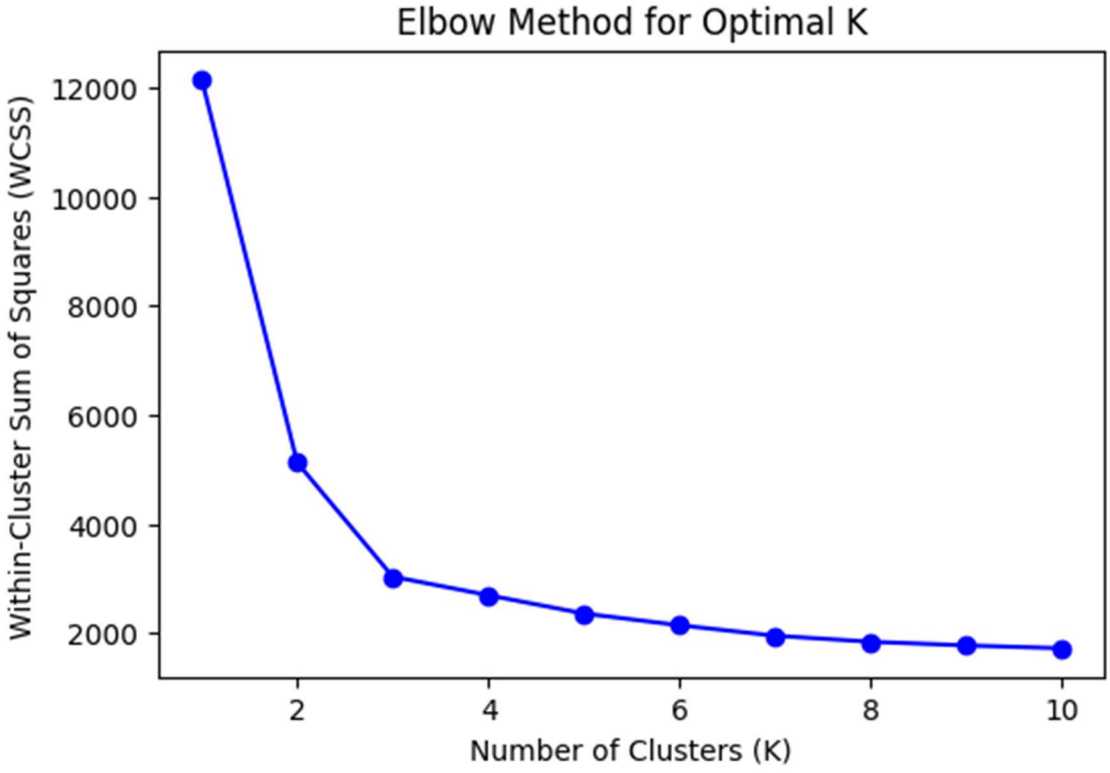
Thus, Logistic Regression is the best model for travel guide because it's the most accurate and balanced making it great for helping people plan their visits!

5.Clustering: Group Similar Destinations



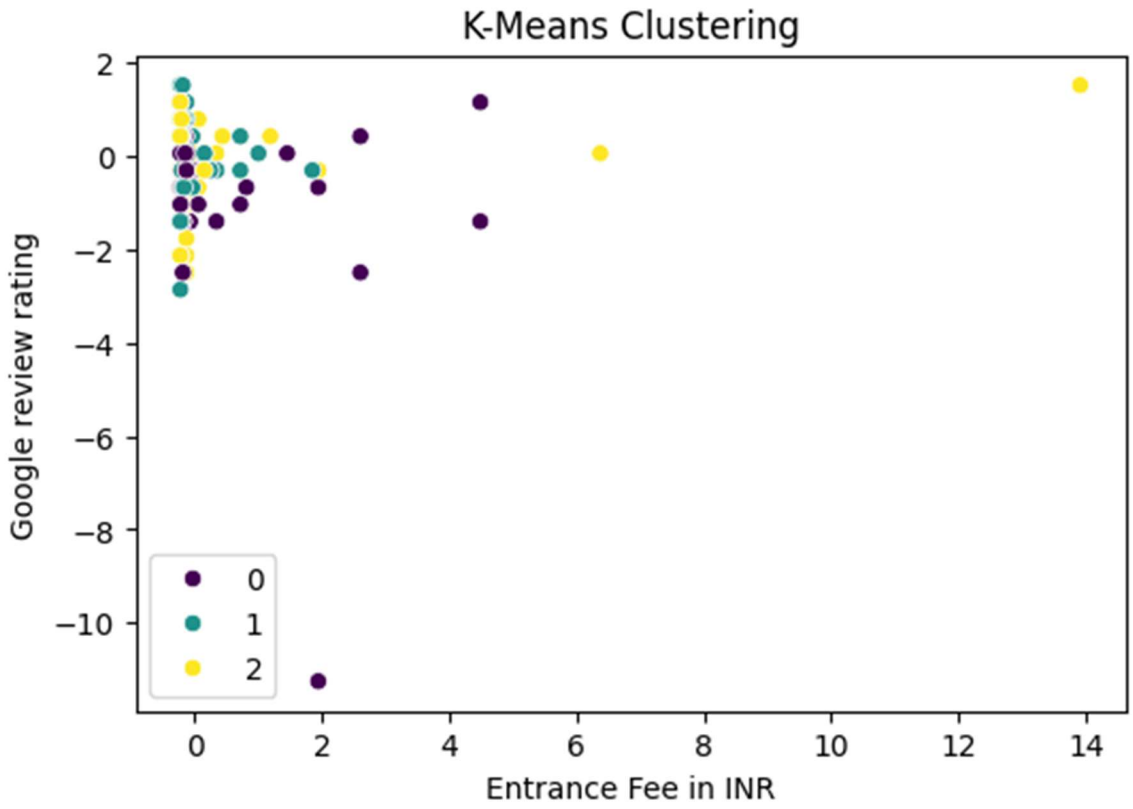
K-Means Clustering

- **Finding Optimal-k using Elbow Method**



- The X-axis represents the number of clusters (K), ranging from 1 to 10. The Y-axis represents the Within-Cluster Sum of Squares (WCSS) which measures the total variance within clusters. As K increases, WCSS decreases because more clusters fit the data more closely. The "elbow" is the point where the rate of decrease in WCSS slows down significantly forming a bend or an elbow shape.
- The curve shows a steep decline from K=1 to K=3 followed by a noticeable bend between K=3 and K=4. After K=3, the rate of WCSS reduction slows down suggesting that adding more clusters beyond 3 provides less improvement in clustering quality. Thus, the elbow point appears to be at **K=3**

• K-Means Clustering Plot



- The data points in the K-Means clustering plot are grouped based on their similarity in two features, "Google review rating" (Y-axis) and "Entrance Fee in INR" (X-axis). Using elbow method ,we get k=3. Distance between data-points is calculated using Euclidean distance, which ensures that points far apart in the feature space and are assigned to different clusters.
- Cluster 0 (purple): Points are grouped around Google ratings of -2 to 0 and entrance fees of 0-5 INR. These points are close to each other in this range indicating moderate ratings and low-to-moderate fees.
- Cluster 1 (teal): Points are concentrated between -3 to 0 ratings and 0-2 INR, lower ratings and very low fees.
- Cluster 2 (yellow): Points are scattered from -1 to 2 ratings and 0-14 INR, representing higher-rated places with higher fees.

DBSCAN Clustering

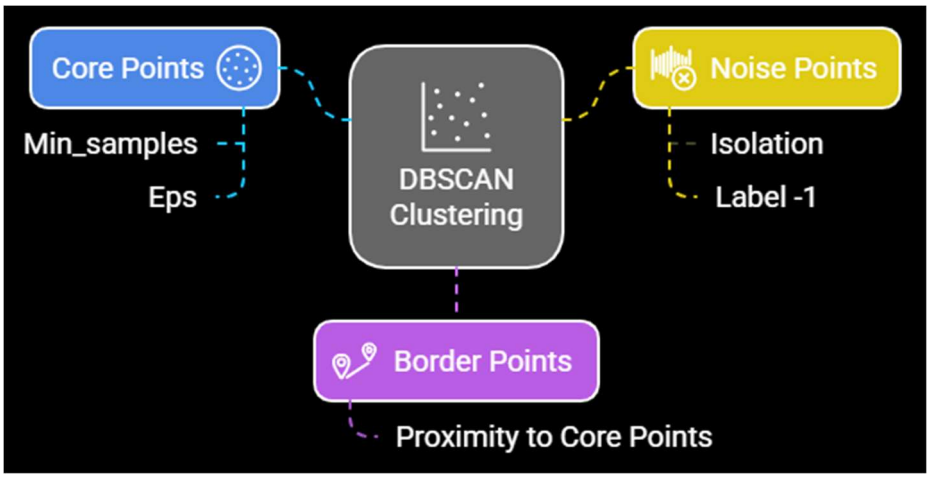
• DBSCAN Phenomenon

- DBSCAN stands for Density-Based Spatial Clustering of Applications with Noise.
- DBSCAN groups data points by looking at how close they are to each other using "density". It contains three major points:

Core Points: A point is a "core point" if it has at least 5 other points (min_samples) close to it within a distance of 3.0 (eps), it as a popular spot with lots of friends nearby.

Border Points: These are points that are close to a core point (within 3.0 distance) but don't have 5 friends of their own, They still join the core point's group but aren't the center of attention.

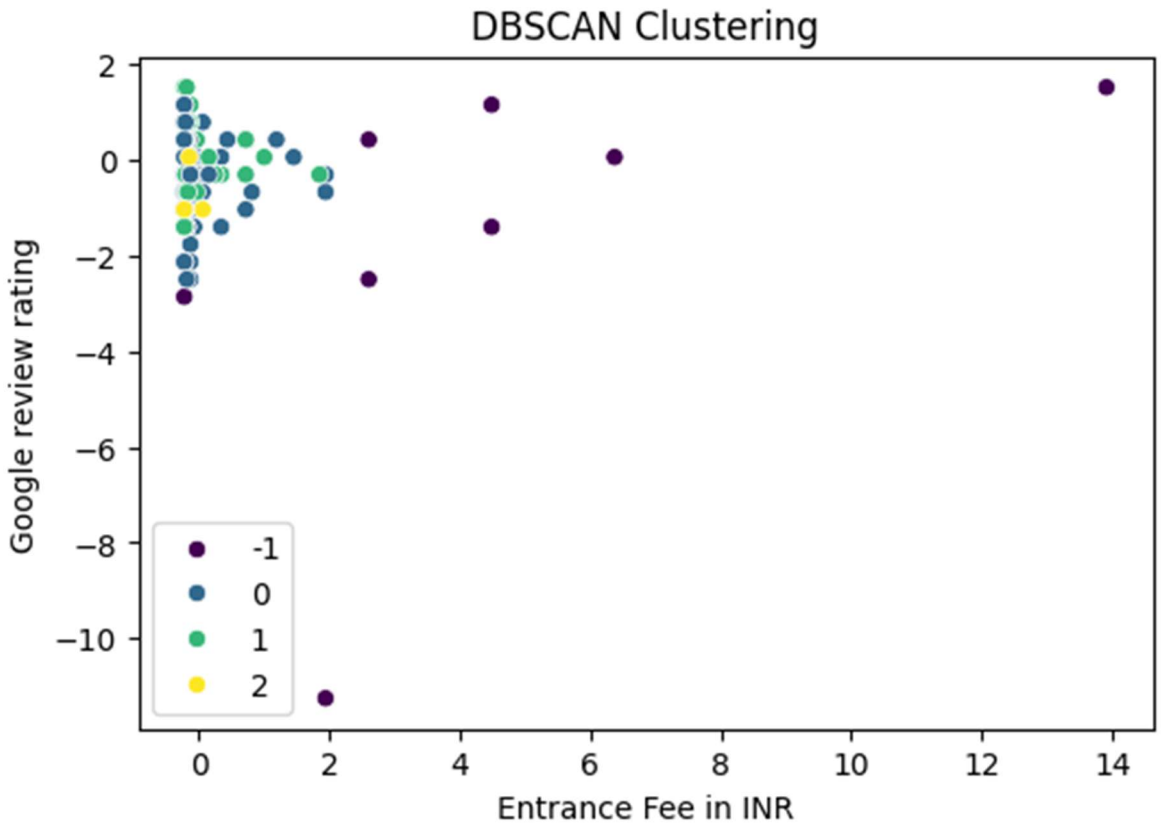
Noise Points: Points that are all alone not close to any core point and don't have enough nearby friends are called "noise" and labeled as -1, They don't fit into any group.



• How Clusters are formed?

- DBSCAN starts with a core point and pulls in all points within 3.0 of it including other core points and border points. If those points lead to more core points, it keeps growing the group. This led to the formation of clusters.
- Unlike K-Means, we don't need to tell DBSCAN how many groups to make, it figures that out by looking at the data. It also naturally spots outliers (noise) and doesn't force them into a group.

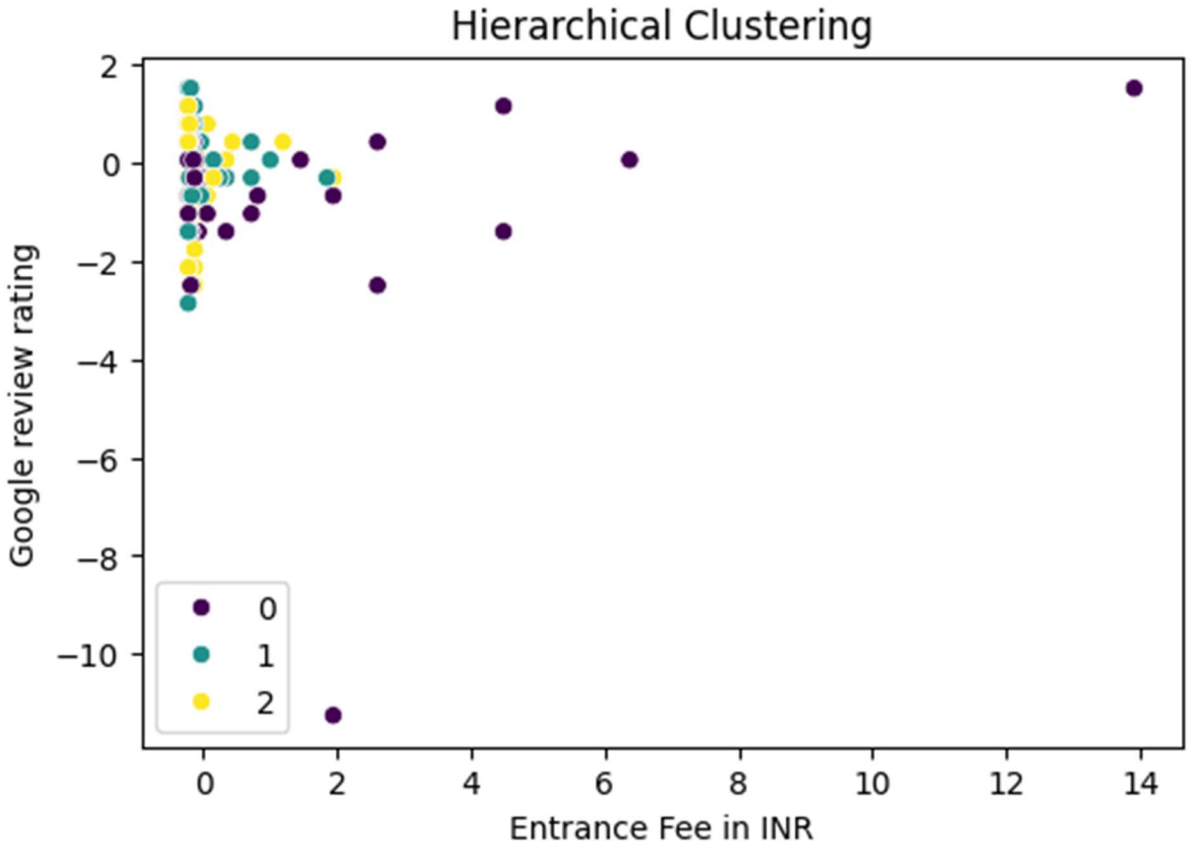
- **DBSCAN Clustering Plot**



- Cluster -1 (purple): Outliers scattered across the plot, these points don't belong to any dense region.
- Cluster 0 (teal): A dense group of points around 0-2 INR and -3 to 0 ratings representing low-fee, low-to-moderate-rated places that form a tight cluster due to high density.
- Cluster 1 (green): Points around 0-2 INR and 0-2 ratings slightly separated from Cluster 0 but still dense indicating a group of low-fee, higher-rated places.
- Cluster 2 (yellow): Points around 0-1 INR and -2 to 0 ratings, another dense group near Cluster 0 showing differences in ratings within low-fee places.
- Thus, DBSCAN identified three clusters within a dense region and marked scattered points as noise. The clusters reflect varying rating levels among low-fee places while high-fee or isolated points (14 INR) are outliers due to low density.

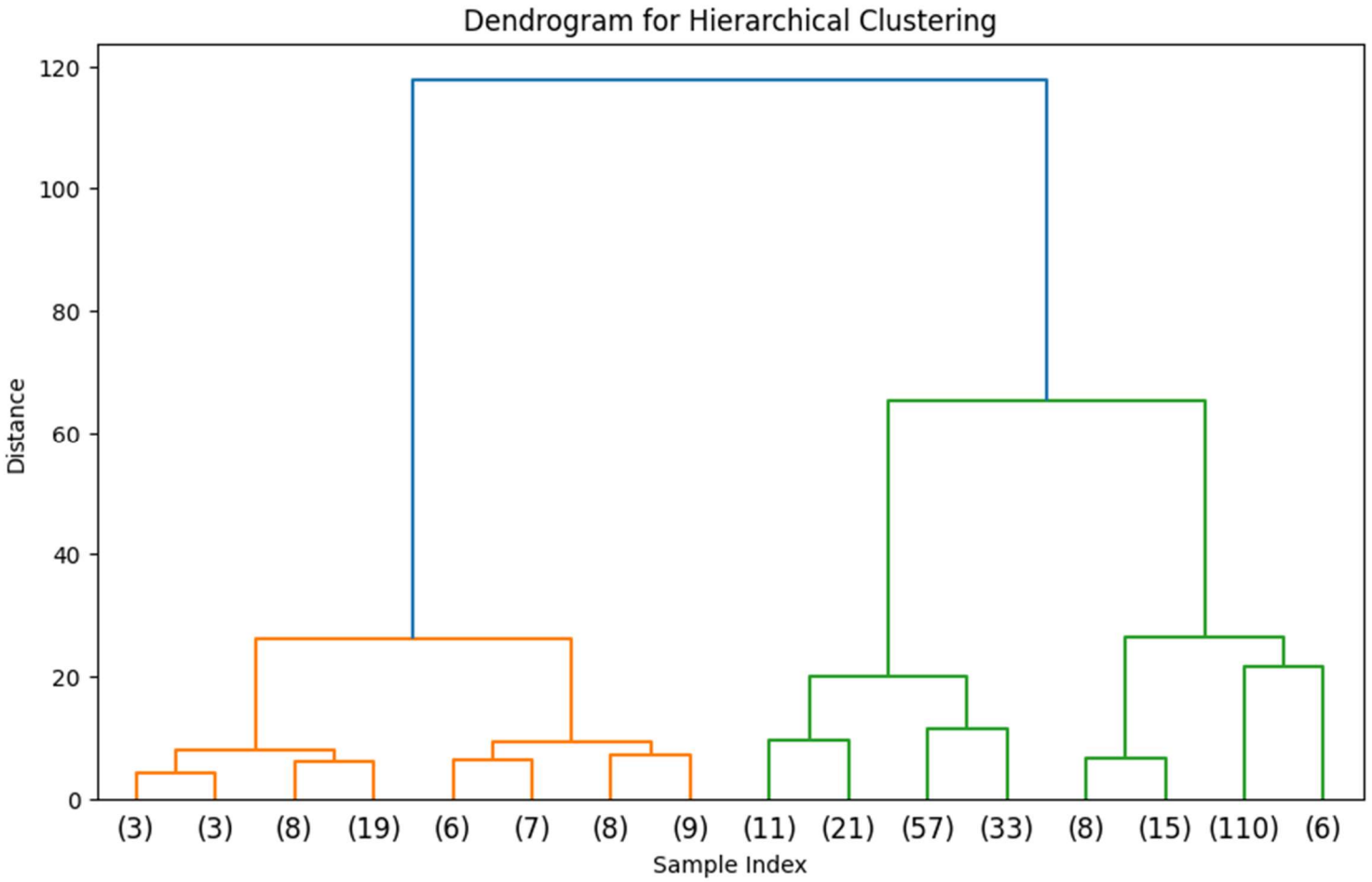
Hierarchical Clustering

Hierarchical Clustering using Agglomerative Clustering which starts with each data point as its own cluster and merges the closest pairs step-by-step until all points are in one big cluster.



- Cluster 0 (purple): Points spread from -3 to 2 ratings and 0-6 INR, with one at 14 INR, indicating moderate-rated, low-to-high-fee places.
- Cluster 1 (teal): Dense group around 0-2 INR and -3 to 2 ratings, indicating low-fee, low-to-moderate-rated places.
- Cluster 2 (yellow): Points around -2-2 INR and 0-2 ratings, a tight group of low-fee, higher-rated places.
- Thus, Hierarchical clustering splits the dense low-fee region (0-2 INR) into two rating-based groups (1 and 2) and assigns scattered high-fee points to Cluster 0.

• Dendrogram



- Early Merges (0-20 Distance, Orange line), At the bottom small clusters form with very low distances. This suggests these points are very similar.
- Mid-Level Merges (10-60 Distance, Green Line), Around a distance of 40-60, larger groups start forming. Connects several clusters small groups are merged into a bigger cluster.
- Final Merge (100-120 Distance, Blue Line), At the top at 120 distance connects all clusters into one big cluster. This represents the point where the entire dataset is considered a single cluster, meaning the maximum distance between any two points has been reached.

- **Model Comparison (Silhouette Score)**

The Silhouette Score measures how similar a point is to its own cluster compared to other clusters. It ranges from -1 to 1:

- A score close to 1 means the point is well-clustered (far from other clusters).
- A score near 0 means the point is on the boundary between clusters.
- A negative score means the point might be in the wrong cluster.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Where:

- a(i): The average distance between point i and all other points in the same cluster.
- b(i): The smallest average distance between point i and all points in any other cluster. It's the distance to the nearest cluster that i is not part of.
- max(a(i),b(i)): The larger of the two values ensures the score is normalized.

K-Means performs the best with a Silhouette Score of 0.417673 followed closely by Hierarchical (0.412287). Both are above 0.4, suggesting good clustering quality.