BUSINESS REPORT

HEALTHCARE DIAGNOSIS FOR RURAL INDIA

This report aims to propose and comprehensively assess an Early Intervention Healthcare AI business solution for Rural India.

Early Intervention Healthcare Solution for Rural India

Leveraging AI in Dermatological Diagnosis

Team Aryan Report Contributions

- Aryan: Sections Abstract, 1, 4, 5, 6, 7, 9, 11, 12
- Bhaskar: Sections 4.1, 4.2
- Veena: Sections 2, 3, 8, 10
- Prateek: Section 4.3

GitHub link: https://github.com/Aryan092/Early-Intervention-Healthcare

---

Abstract

This report explores the potential of leveraging artificial intelligence (AI) technology to develop an early intervention healthcare solution for rural India, focusing specifically on dermatological diagnosis. Through business and financial modelling, as well as demographical segmentation analysis in health insurance and dermatological diseases, this report assesses the feasibility and impact of implementing a dermatological diagnosis app using deep learning technology. The analysis reveals promising opportunities for improving early detection and treatment outcomes in dermatology, particularly in underserved regions. Moreover, the report emphasizes the significance of transparency and trust in AI algorithms, advocating for ongoing efforts to enhance model interpretability and mitigate bias.

The report, therefore, advocates for the adoption of AI-driven solutions in dermatological diagnosis to enhance accessibility, accuracy, and adoption rates in rural India. By leveraging the scalability and potential for growth in AI disruption, the proposed early intervention healthcare solution has the potential to significantly impact healthcare outcomes and improve accessibility for rural populations in India.

1. Problem Statement

With the increasing prevalence of skin conditions and limited access to specialized healthcare in rural areas, there is a critical need for innovative solutions that can bridge the gap in healthcare services.

Hence the Underlying Goal – Increase accessibility and feasibility of healthcare, specifically in dermatology, in India by serving as a triangular link to medical expertise and insurance.

Solution – Provide an accessible, easy-to-use and understandable skin diagnosis app utilising deep learning in regions where there is a paucity of resources and dermatological expertise. Additional features could include tracking of symptoms and response to treatment.

Key aim entails an Explainable AI implementation – aims to open the "black box" of ML. Explain why the underlying reasoning behind the model output.

2. Market/Customer/Business Need Assessment

Skin cancer develops more frequently in India as a result of exposure to UV light, arsenic, coal tar, and other hydrocarbons. Furthermore, industrial chemicals and air pollution may make the problem worse. Risks for skin cancer include indoor tanning and photochemotherapy (PUVA) treatments, particularly for Indians. An elevated risk of skin cancer has been linked to elevated levels of arsenic in groundwater in several parts of India, such as the Ganges River basin. A large number of India's rural communities lack awareness of skin cancer prevention and detection, and they also have limited access to healthcare resources.

India is now the most populous country in the world, surpassing China, and is quickly approaching the 1.5 billion level. The United Nations (UN) predicts that, because the Indian population is still relatively young, it will keep growing until 2060. Taking these numbers into consideration, there are less than one dermatologist for every 100,000

individuals in India, where there are just about 11,000 skin care professionals nationwide (Mesko, 2023). Dermatology is often neglected, especially in India, where most individuals living in suburban and rural regions lack the resources or information needed to recognize potential skin hazards.

The shift in human consumption from natural to chemical food may lead to a rise in skin cancer cases in the future for various reasons. There aren't enough specialists treating skin cancer to address the condition and provide insightful analysis and precise numbers. Too many applications for too little money will be possible with AI/ML.

3. Target Specifications and Characterization

The target specifications and characterization for skin cancer detection can be summarized as follows:

A. Demographic:

- **Age Group**: The primary target audience comprises individuals between the ages of 18 to 60, as skin conditions can affect people across a broad age range.
- **Technological Literacy**: The app aims to cater to users with varying levels of technological proficiency. Hence, the interface ought to be user-friendly, allowing those with basic smartphone knowledge to navigate easily.
    - This is taken further by talking about a significant market gap giving detailed explanations behind the symptom diagnostic conclusion.
- **Geographic Location:** The app targets users across urban, semi-urban, and rural areas of India, acknowledging the diverse healthcare needs across different regions.
- **Socioeconomic Status:** To ensure inclusivity, the app should be accessible to users from diverse socioeconomic backgrounds. This includes both urban and rural populations.

B. Demographic Considerations:

- **Gender Inclusivity:** The app should be designed to address the dermatological concerns of all genders, considering the diverse skin conditions and cosmetic concerns.
- **Cultural Sensitivity:** Given the diverse population in India, the app should account for various skin tones, types, and cultural practices. The AI algorithm must be trained on a dataset that represents this diversity.

C. User Behaviour and Preferences:

- **Health Consciousness:** Target users who are health-conscious and proactive about monitoring and addressing their skin health.
- **Privacy Concerns:** Users are likely to be cautious about sharing personal health data. The app will prioritize robust data security measures and transparent privacy policies.
- **Desire for Quick Access:** Given the potential lack of immediate access to dermatologists, the app's target users value quick and convenient solutions for skin condition assessments.

D. Healthcare Engagement:

- **Existing Dermatology Patients:** The app can be designed to assist existing dermatology patients in monitoring their conditions between regular appointments, fostering continuous care.
- **Underserved Populations:** Target individuals who lack easy access to dermatological services, including those in remote areas where dermatologists are scarce.

4. Market Segmentation

Before segmenting a sector it is imperative to understand the key areas to segment in order to give direction and produce fruitful results. Types of Market Segmentation in Health Care [5]:

- Based of Patient characteristics
  - Demography
  - Physiography
  - Usage patterns
- Based on Marketing practices

All these forms of Market segmentation would be useful at an early stage but considering a start-up at pre-business developmental stage (perfecting the product) and data availability, technical analysis of the Patient Demographical type was selected as the primary focus. Marketing practices was qualitatively considered in the Business modelling section.

Data Collection:

1. Health Insurance Dataset from Kaggle
   https://www.kaggle.com/datasets/teertha/ushealthinsurancedataset

2. Dermatology Dataset (Multi-class classification) from Kaggle
   https://www.kaggle.com/datasets/olcaybolat1/dermatology-dataset-classification/data

3. Subnational Demographical access to Health Care in India Dataset from HDX
   https://data.humdata.org/dataset/dhs-subnational-data-for-india/resource/e0e3319e-51d9-4311-be69-57bafe4d4574

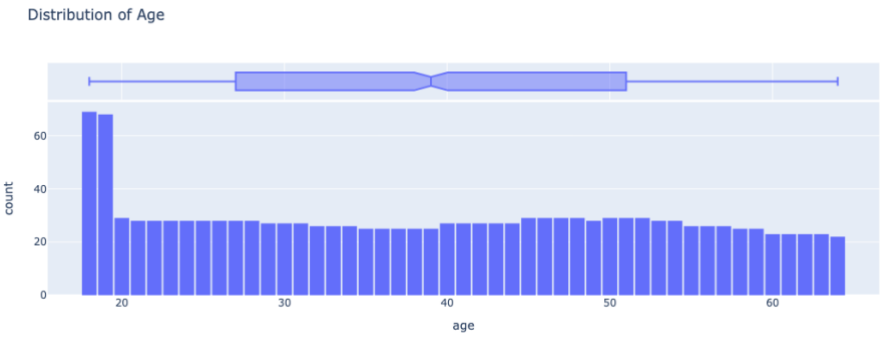
4.1 Patient Demographical Insurance Segmentation Analysis

The dataset seems to be focused on factors that could influence health insurance costs for individuals. It includes both demographic (age, sex, region) and health-related factors (BMI, smoking status), as well as information on dependents, which are all relevant in determining insurance premiums or analysing health risks.

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520
...	...	...	...	...	...	...	...
1333	50	male	30.970	3	no	northwest	10600.54830
1334	18	female	31.920	0	no	northeast	2205.98080
1335	18	female	36.850	0	no	southeast	1629.83350
1336	21	female	25.800	0	no	southwest	2007.94500
1337	61	female	29.070	0	yes	northwest	29141.36030

1338 rows × 7 columns

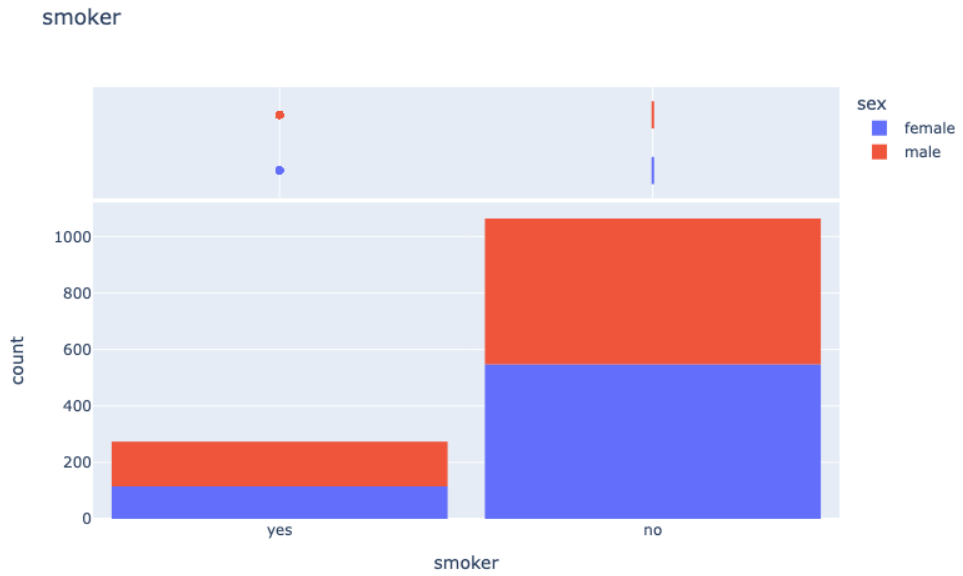
Combined box plot and histogram of the distribution of age within a dataset



1. **Histogram Analysis**:
    - The histogram shows a unimodal distribution with the highest frequency of individuals in the youngest age bracket.
    - There's a high concentration of individuals in their 20s, and the frequency gradually decreases as age increases.

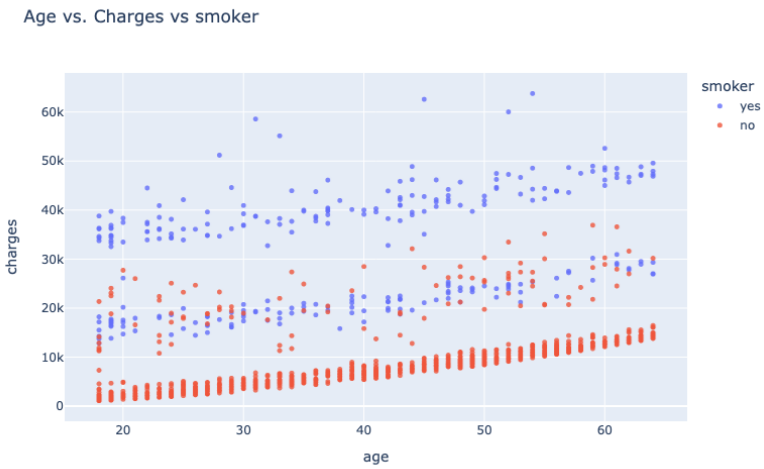
- There are fewer individuals in the dataset who are older, with a very low frequency of individuals over 60 years of age.

Bar chart with a count plot



1. There are significantly more non-smokers than smokers in this dataset.
2. For both smokers and non-smokers, the distribution between males and females is quite similar, with a slightly higher count of males in both categories.

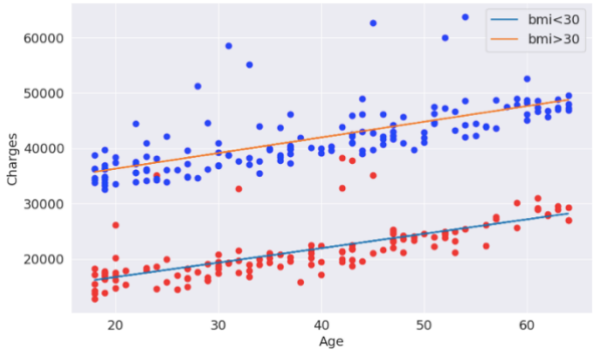
Scatter plot comparing age, charges, and smoking status



1. **Age and Charges Correlation**: The plot suggests a positive correlation between age and insurance charges; as age increases, the charges tend to increase

as well. This trend is common in insurance pricing, where older individuals are often charged more due to higher associated health risks.

2. **Impact of Smoking**: There are two distinct clusters of data points, which likely represent smokers and non-smokers. The cluster with higher charges at any given age represents smokers ('yes'), which indicates that smokers are charged significantly more for insurance than non-smokers ('no'). This is a typical finding, as smoking is associated with higher health risks and therefore higher insurance costs.



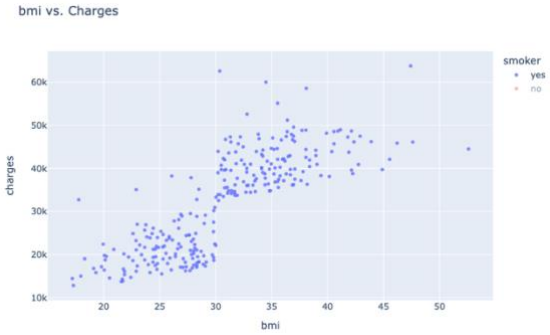
3. **Trend Lines**: There are two trend lines that appear to fit the data for each category.

Compare BMI to insurance charges

For Individuals with BMI ≤ 30:

- Among non-smokers (red points in the first plot), those with a BMI ≤ 30 seem to have a fairly uniform distribution of charges that don't show a strong dependency on BMI within this range.
- Smokers (blue points in the first plot) with a BMI ≤ 30 also display an increase in charges, but the range is more variable and generally higher than non-smokers.

For Individuals with BMI > 30:

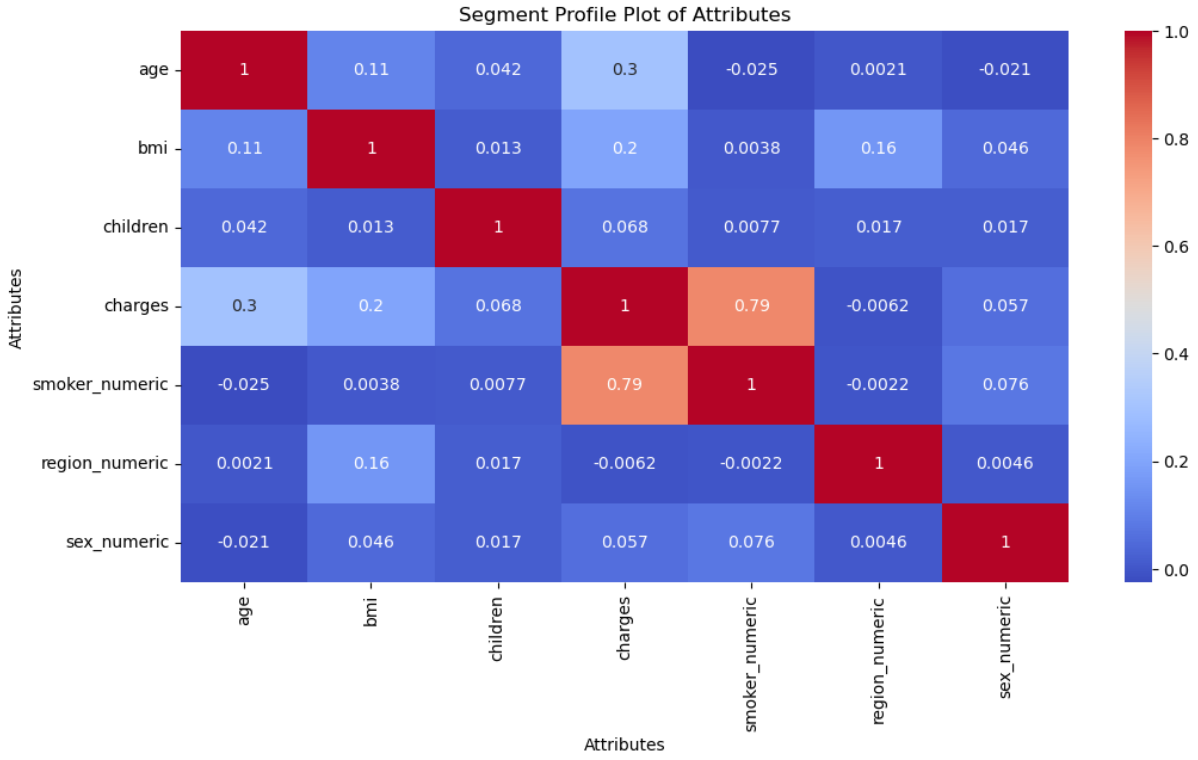


- Non-smokers with a BMI > 30 (red points) do not show a significant increase in charges compared to those with a BMI ≤ 30, suggesting that while BMI is a factor, its impact on charges is not as pronounced for non-smokers.
- Smokers with a BMI > 30 (blue points) show a trend of even higher charges, which indicates that a higher BMI exacerbates the cost of insurance for smokers. The combination of being a smoker and having a high BMI seems to result in the highest insurance charges.

Overall Trends:

- The plots collectively suggest that smoking status has a more substantial impact on insurance charges than BMI alone.
- Individuals with a higher BMI (>30), particularly smokers, are at the higher end of insurance charges, underscoring the combined effect of smoking status and higher BMI as significant factors in determining insurance costs.
- The clustering of charges for non-smokers remains tighter and lower across the range of BMI compared to smokers, indicating a less steep correlation between BMI and charges for non-smokers.

Heatmap of a correlation matrix



Strong Positive Correlation: The most visually prominent element is the strong positive correlation between **smoker_numeric** and **charges** (0.79), indicating that smoking status is highly associated with higher insurance charges

OLS regression results (data 1 including smoker)

OLS Regression Results

Dep. Variable:	charges	R-squared:	0.755
Model:	OLS	Adj. R-squared:	0.750
Method:	Least Squares	F-statistic:	164.8
Date:	Fri, 01 Mar 2024	Prob (F-statistic):	1.36e-79
Time:	16:52:49	Log-Likelihood:	-2758.7
No. Observations:	274	AIC:	5529.
Df Residuals:	268	BIC:	5551.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.117e+04	985.505	-11.331	0.000	-1.31e+04	-9226.329
age	263.4997	25.253	10.434	0.000	213.780	313.219
bmi	1453.6802	57.276	25.380	0.000	1340.913	1566.448
children	216.8542	303.946	0.713	0.476	-381.571	815.279
smoker_numeric	-1.117e+04	985.505	-11.331	0.000	-1.31e+04	-9226.329
region_numeric	-295.0558	331.593	-0.890	0.374	-947.913	357.802
sex_numeric	-346.2074	717.154	-0.483	0.630	-1758.179	1065.765

Omnibus:	60.164	Durbin-Watson:	1.901
Prob(Omnibus):	0.000	Jarque-Bera (JB):	143.018
Skew:	1.037	Prob(JB):	8.79e-32

1. **Model Fit**: An R-squared value of 0.755 indicates a strong model fit, explaining approximately 75.5% of the variance in insurance charges.
2. **Significant Predictors**: Age and BMI have a statistically significant positive relationship with insurance charges.
3. **Children**: The number of children has a positive coefficient but is not statistically significant (p-value = 0.476).
4. **Region and Sex**: Both region_numeric and sex_numeric show no significant impact on insurance charges (p-values = 0.374 and 0.630, respectively).
5. **Model Diagnostics**:
    - The model is statistically significant as a whole, indicated by the F-statistic.
    - The Durbin-Watson statistic of 1.901 suggests no significant autocorrelation concerns.

- Tests for normality indicate significant skewness and kurtosis in the residuals, questioning the p-value reliability.

6. **Coefficient Implications**:
    - For age, a one-unit increase corresponds to an approximate increase in charges by 263.50.
    - The BMI coefficient suggests a one-unit increase in BMI corresponds to an increase in charges by 1453.68.

OLS regression results (data 2 not including smoker)

OLS Regression Results

Dep. Variable:	charges	R-squared:	0.417
Model:	OLS	Adj. R-squared:	0.415
Method:	Least Squares	F-statistic:	151.5
Date:	Fri, 01 Mar 2024	Prob (F-statistic):	2.09e-121
Time:	16:52:49	Log-Likelihood:	-10477.
No. Observations:	1064	AIC:	2.097e+04
Df Residuals:	1058	BIC:	2.100e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2236.4125	812.388	-2.753	0.006	-3830.486	-642.339
age	264.5855	10.072	26.269	0.000	244.822	284.349
bmi	18.4778	23.706	0.779	0.436	-28.039	64.995
children	587.2669	115.565	5.082	0.000	360.503	814.030
smoker_numeric	2.153e-13	8.64e-14	2.490	0.013	4.57e-14	3.85e-13
region_numeric	-461.9849	127.882	-3.613	0.000	-712.916	-211.053
sex_numeric	-526.7463	281.474	-1.871	0.062	-1079.057	25.564

Omnibus:	710.730	Durbin-Watson:	2.053
Prob(Omnibus):	0.000	Jarque-Bera (JB):	5704.085
Skew:	3.191	Prob(JB):	0.00

...

1. **Model Fit**: R-squared is 0.417, suggesting a moderate fit with the model explaining about 41.7% of variance in insurance charges.
2. **Significant Predictors**:
    - Age is a significant predictor with a positive coefficient, indicating charges increase with age.

- Children are also a significant predictor, with more children associated with higher charges.

3. **Non-Significant Predictors**:
    - BMI has a positive coefficient but is not statistically significant (p-value = 0.436).
    - The coefficient for smoker status is negligible and not consistent with typical interpretations of its impact on charges.
4. **Other Observations**:
    - Region has a significant negative coefficient, indicating some regions are associated with lower charges.
    - Sex has a negative coefficient, suggesting a possible but weak association with lower charges for males.
5. **Coefficient Interpretation**:
    - Coefficients indicate the change in charges for a one-unit change in predictors, e.g., each additional year of age increases charges by about 264.59.

4.2 Patient Demographical Disease Segmentation Analysis

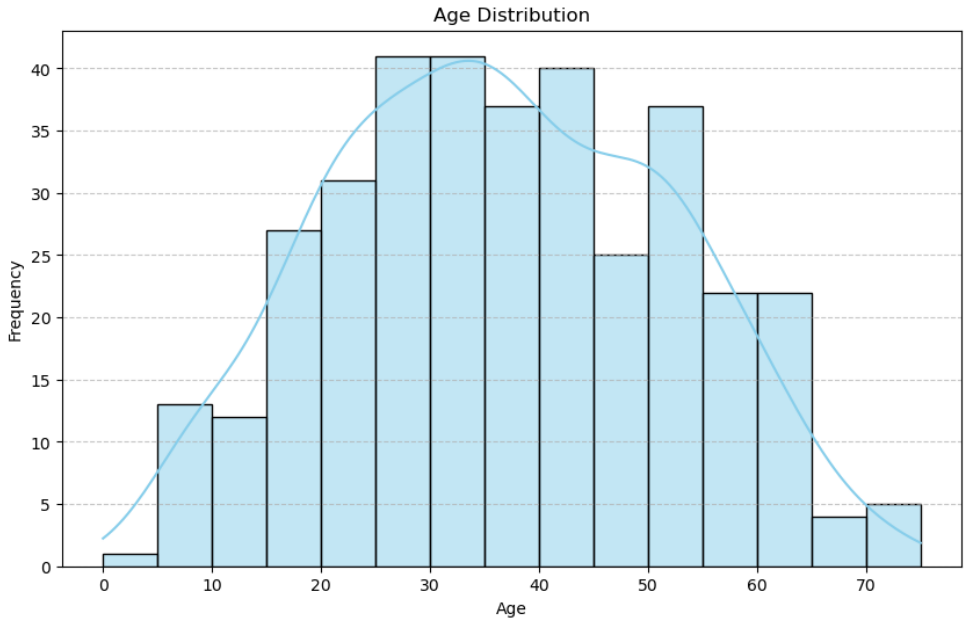
	erythema	scaling	definite_borders	itching	koebner_phenomenon	polygonal_papules	follicular_papules	oral_mucosal_involvement	knee_and_elbow_involvement	s
0	2	2	0	3	0	0	0	0	1	
1	3	3	3	2	1	0	0	0	1	
2	2	1	2	3	1	3	0	3	0	
3	2	2	2	0	0	0	0	0	3	
4	2	3	2	2	2	2	0	2	0	
...	...	...	...	...	...	...	...	...	...	
361	2	1	1	0	1	0	0	0	0	
362	3	2	1	0	1	0	0	0	0	
363	3	2	2	2	3	2	0	2	0	
364	2	1	3	1	2	3	0	2	0	
365	3	2	2	0	0	0	0	0	3	

366 rows × 35 columns

- The dataset includes a variety of features that describe clinical and possibly histological characteristics, such as **erythema**, **scaling**, **definite borders**, **itching**, **koebner phenomenon**, and more, suggesting a focus on dermatological conditions.

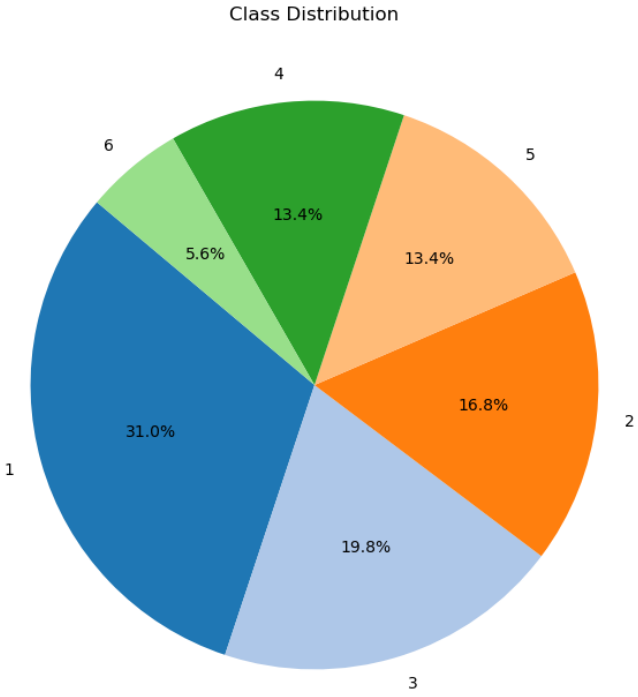
- It also contains a column for **age**, indicating the age of the individuals, which varies widely across the dataset, reflecting a diverse study population.
- The **class** column is used as a categorical target variable, with values from 1 to 6, indicating different conditions or disease states that the dataset aims to classify

Histogram of age distribution

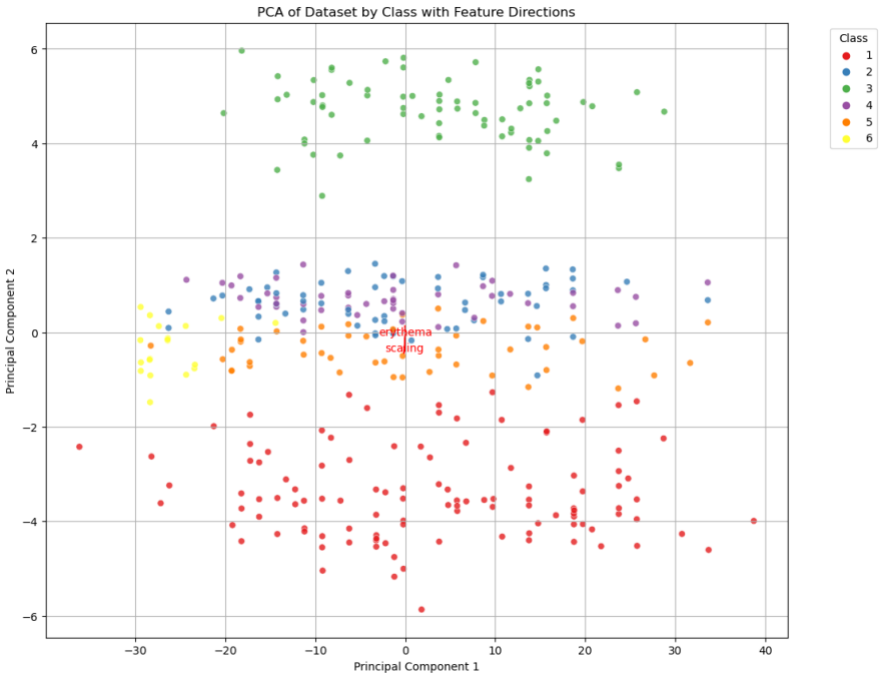


1. **Age Range**: The histogram covers a wide range of ages, from 0 to over 70 years, indicating that the dataset includes a broad demographic.
2. **Peak Age Group**: The highest frequency (mode) seems to be in the age group of around 30 to 40 years, where the tallest bar is located.
3. **Symmetry**: The distribution appears to be fairly symmetric about the peak, with a slight right skew indicated by the longer tail extending toward the older age groups.

The chart shows the distribution of a certain variable across six different classes

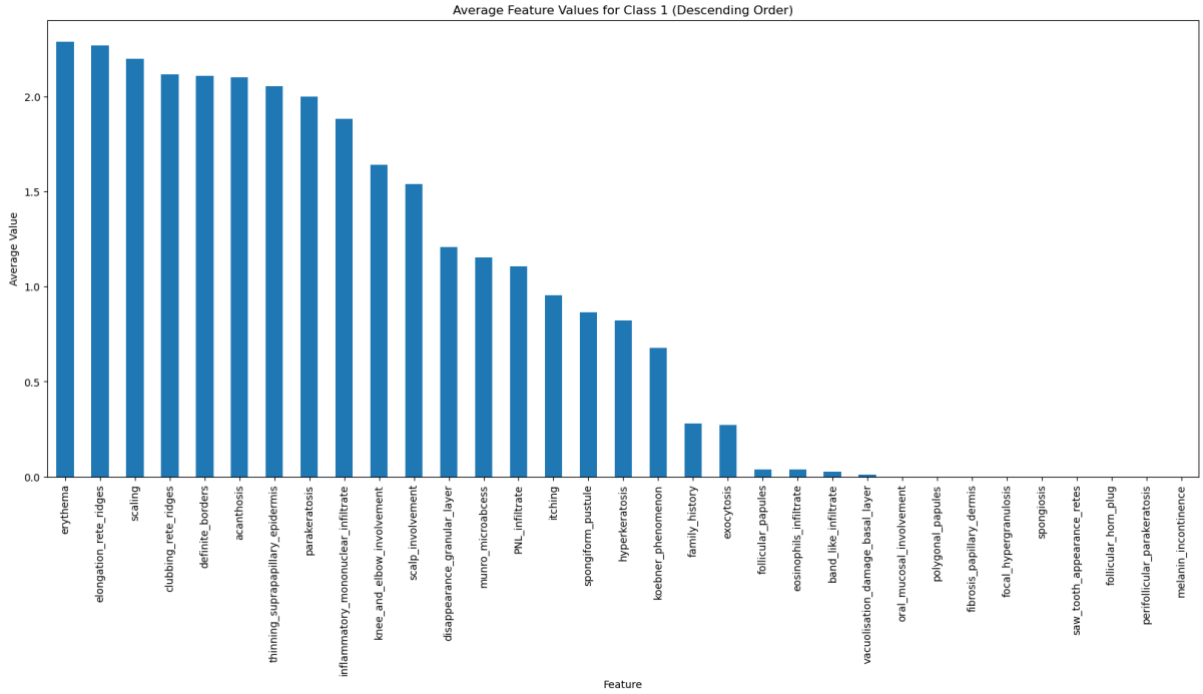


1. **Largest Class**: Class 1 is the largest group, constituting 31.0% of the dataset. This suggests that whatever condition or characteristic Class 1 represents, it is the most common within this data.
2. **Smallest Class**: Class 6 is the smallest group, making up 5.6% of the dataset. This could indicate that the condition or characteristic represented by Class 6 is the least common or least frequently observed in the data.
3. **Balanced Distribution**: Aside from the largest and smallest classes, the remaining classes (2, 3, 4, and 5) are relatively balanced, with each class making up between 13.4% to 19.8% of the data. This suggests a fairly even distribution among these classes.



The plot above offers a deeper insight into how the original features influence the separation between classes in the reduced dimensionality space created by PCA.

Bar graph now displays the average values of various features for Class 1



Understanding Class 1: The distribution of feature averages provides insights into the defining characteristics of Class 1. Features with higher averages could be critical in differentiating Class 1 from other classes, potentially corresponding to specific symptoms, signs, or pathological characteristics of the skin condition(s) this class represents.

The age distribution for in Class 1

- **Count**: There are 111 individuals in Class 1.
- **Mean Age**: The average age is approximately 39.38 years..

- **Minimum Age**: The youngest individual in Class 1 is newborn (0 years).
- **25th Percentile**: 25% of individuals are 27 years old or younger.
- **Median Age (50th Percentile):** The median age is 39 years, meaning half of the individuals are younger than 39 and half are older.

- **75th Percentile**: 75% of individuals are 52.5 years old or younger.
- **Maximum Age**: The oldest individual in Class 1 is 75 years old.

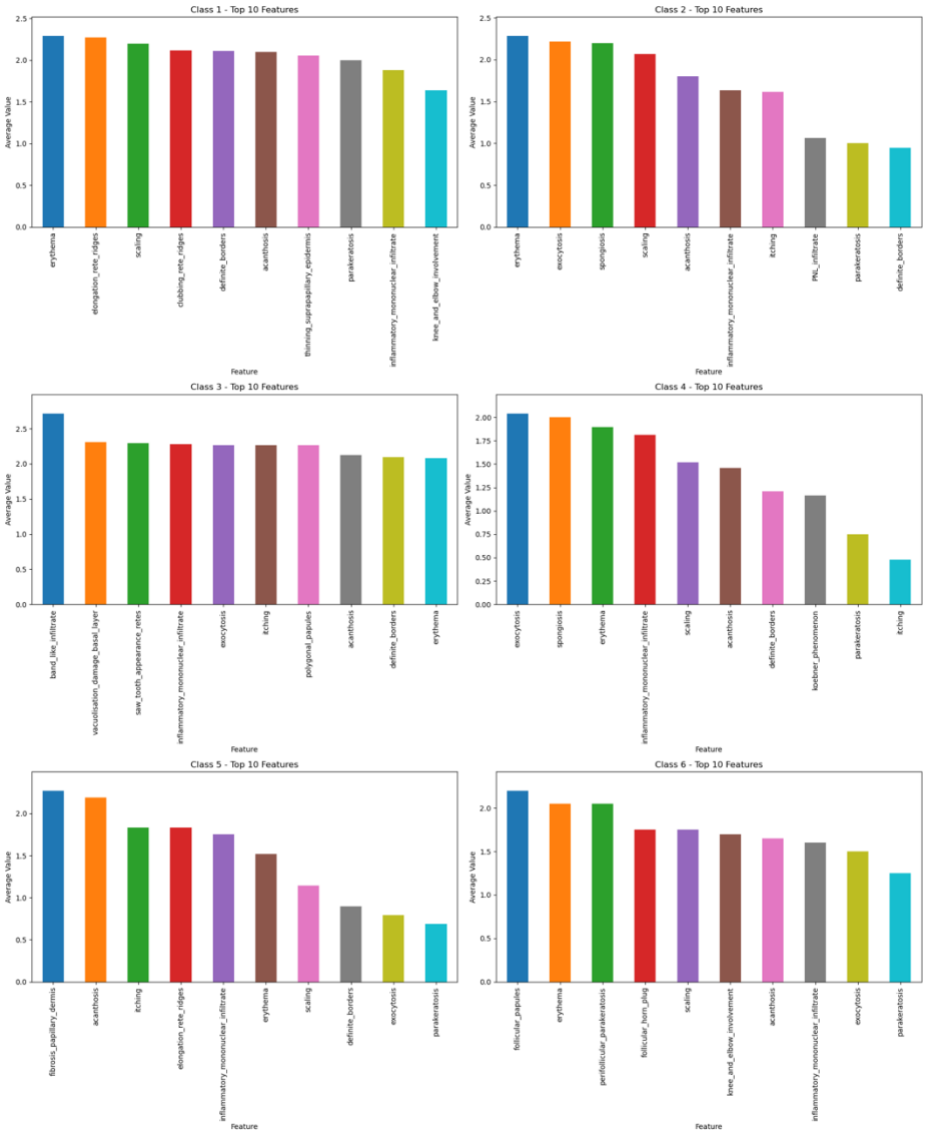
Analysis:

- The age distribution for Class 1 spans a wide range, from newborns to 75 years old, with a fairly even spread across the age spectrum. This suggests that the condition(s) represented by Class 1 can affect individuals at any age.
- The distribution appears to be roughly symmetric around the mean, with a slight skew towards older ages, as indicated by the mean being slightly higher than the median.
- The presence of individuals across all age groups may indicate that the condition(s) associated with Class 1 are not particularly age-specific, affecting a broad demographic.

Top 10 Symptoms from each class

Key Observations Across Classes:

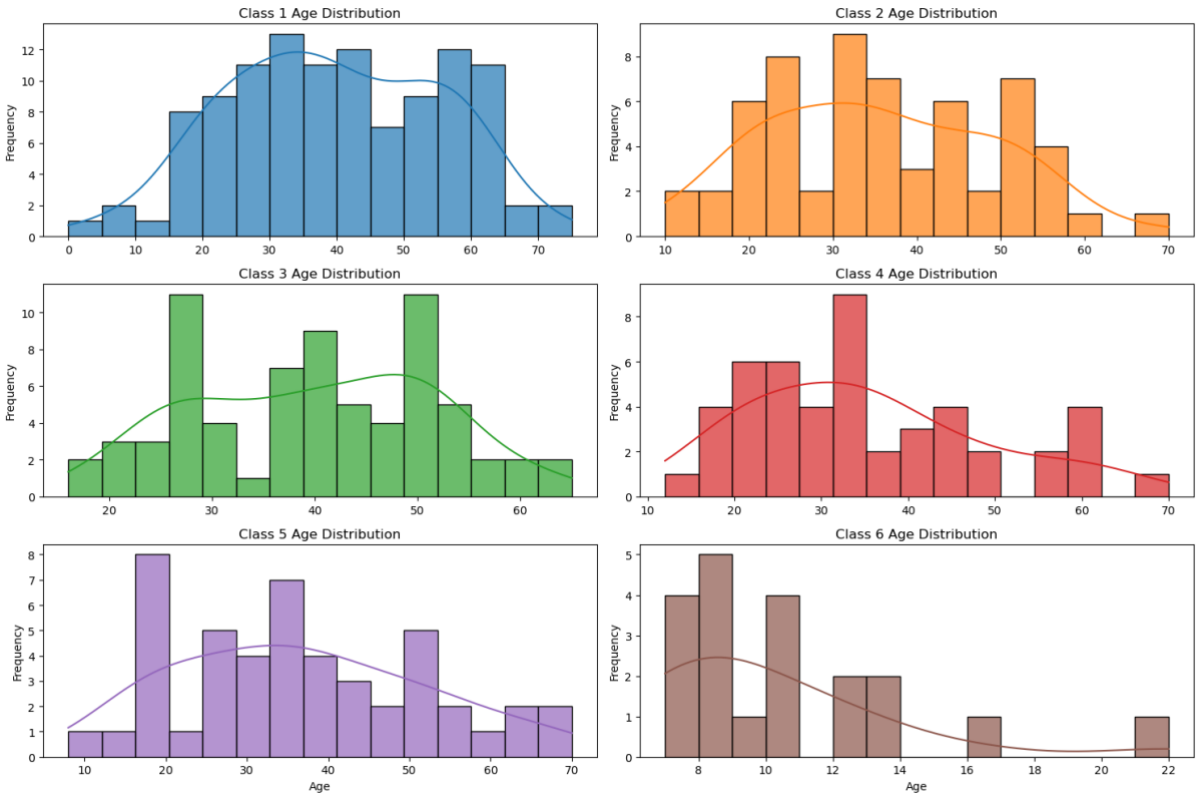
Distinctive symptoms: Each class has a unique set of top symptoms, indicating distinctive characteristics that can help differentiate between the conditions represented by each class.



Common symptoms: While each class has its unique top symptoms, there may be common symptoms that appear across multiple classes, albeit with different average values. This could suggest overlapping symptoms or characteristics among the conditions represented.

Variability in Importance: The importance (average value) of top symptoms varies significantly across classes, underlining the heterogeneity within the dataset. Some features are highly prominent in certain classes but not as significant in others.

The age distribution for all Class 1-6



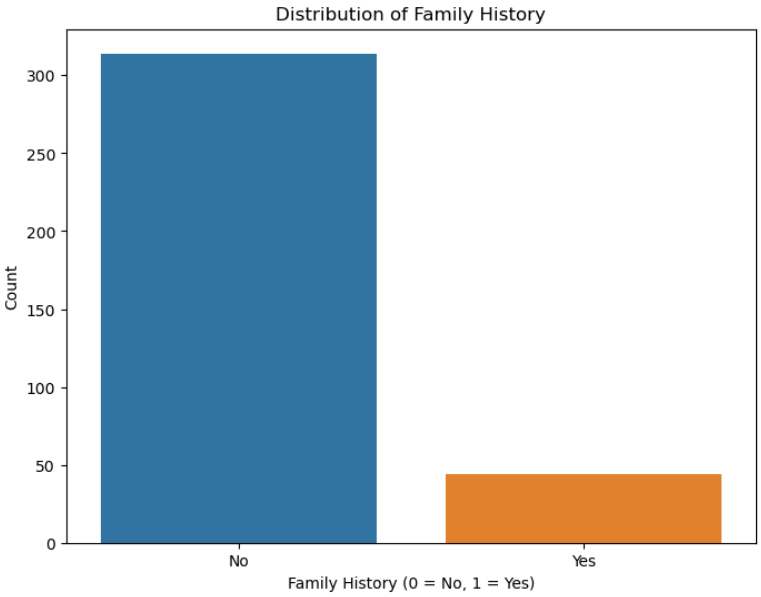
- Classes 1, 2, 4, and 5 cover a broad age range, primarily affecting adults, with a notable concentration in the mid-30s to early 40s. These conditions are prevalent across a wide demographic, suggesting a need for healthcare strategies that cater to a diverse age group.
- Class 3 is more focused on the adult population, particularly middle-aged individuals, indicating conditions with higher prevalence or better diagnosis rates in this demographic.
- Class 6 is unique, targeting children and young adolescents, and highlights the importance of pediatric-specific healthcare interventions.

The distribution of values in the "family_history"

- **87.71%** of individuals have no family history of the condition (value 0).
- **12.29%** of individuals have a family history of the condition (value 1).

This indicates that a majority of the individuals in the dataset do not have a family history of the condition, with only a small fraction reporting a positive family history. This could suggest that for the conditions represented in this dataset, genetic or hereditary factors might play a less dominant role, or at least, are not commonly reported among the cases included in this analysis.

Family history of certain health conditions can indeed affect health insurance in several ways



1. **Premium Costs:** Insurance companies may consider an individual's family medical history when determining premium costs. If there's a known genetic predisposition to certain conditions like heart disease, cancer, or diabetes, insurers might view this as a higher risk and could potentially charge higher premiums.
2. **Coverage Terms:** The specific terms of coverage, including exclusions and limitations, might be influenced by family history. For instance, there may be waiting periods for coverage related to conditions for which there is a strong family history.
3. **Pre-existing Condition Clauses:** In some health insurance plans, especially those that are not subject to certain regulations like the Affordable Care Act in the United States, a family history of certain conditions might lead to exclusions for related illnesses as pre-existing conditions.
4. **Risk Assessment:** Insurance underwriters use family medical history as part of their risk assessment process. This can affect eligibility for some types of insurance or result in higher rates.

Section 4.3 Geographical Segmentation

Step 1 Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
sns.set()
import scipy
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage
import plotly.graph_objs as go
from plotly.offline import init_notebook_mode, iplot
from sklearn.decomposition import PCA
from sklearn.metrics.pairwise import cosine_similarity
```

Step 2 Reading data file into a python data frame

```
data = pd.read_csv('/content/india-districts-census-2011.csv')
data.head(5)
```

	District code	State name	District name	Population	Male	Female	Literate	Male_Literate	Female_Literate	SC	...	Power_Parity_Rs_S
0	1	JAMMU AND KASHMIR	Kupwara	870354	474190	396164	439654	282823	156831	1048	...	
1	2	JAMMU AND KASHMIR	Badgam	753745	398041	355704	335649	207741	127908	368	...	
2	3	JAMMU AND KASHMIR	Leh(Ladakh)	133487	78971	54516	93770	62834	30936	488	...	
3	4	JAMMU AND KASHMIR	Kargil	140802	77785	63017	86236	56301	29935	18	...	
4	5	JAMMU AND KASHMIR	Punch	476835	251899	224936	261724	163333	98391	556	...	

5 rows × 118 columns

Step 3 Statistical Summary

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 640 entries, 0 to 639
Columns: 118 entries, District code to Total_Power_Parity
dtypes: int64(116), object(2)
memory usage: 590.1+ KB
```

```
data.describe().round(2)
```

	District code	Population	Male	Female	Literate	Male_Literate	Female_Literate	SC	Male_SC	Female_SC	.
count	640.00	640.00	640.00	640.00	640.00	640.00	640.00	640.00	640.00	640.00	
mean	320.50	1891960.90	973859.78	918101.12	1193185.64	679318.16	513867.48	314653.71	161773.93	152879.78	
std	184.90	1544380.29	800778.52	744986.39	1068582.63	592414.36	480181.61	312981.76	161121.56	152033.63	
min	1.00	8004.00	4414.00	3590.00	4436.00	2614.00	1822.00	0.00	0.00	0.00	
25%	160.75	817861.00	417168.25	401745.75	482598.25	276436.50	200892.00	83208.50	42307.00	42671.75	
50%	320.50	1557367.00	798681.50	758920.00	957346.50	548352.50	403859.00	246016.00	125548.50	117855.00	
75%	480.25	2583551.25	1338604.50	1264276.75	1602260.25	918858.25	664155.00	447707.75	228460.25	214050.25	
max	640.00	11060148.00	5865078.00	5195070.00	8227161.00	4591396.00	3635765.00	2464032.00	1266504.00	1197528.00	

8 rows × 116 columns

Step 4 Checking for null values

```
data.isnull().sum()
```

```
District code                   0
State name                      0
District name                   0
Population                      0
Male                            0
                               ..
Power_Parity_Rs_330000_425000    0
Power_Parity_Rs_425000_545000    0
Power_Parity_Rs_330000_545000    0
Power_Parity_Above_Rs_545000     0
Total_Power_Parity               0
Length: 118, dtype: int64
```

There are no null values so carrying forword with our analysis.

Step 5 Dumping unwanted columns

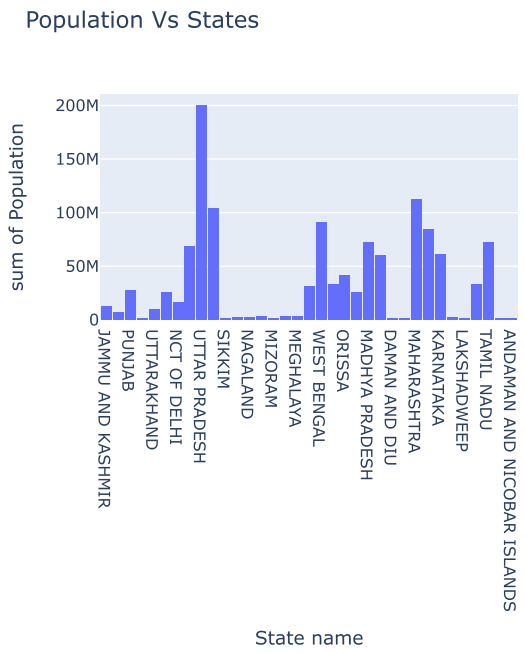
```
data.drop(['SC','Male_SC','Female_SC','ST','Male_ST','Female_ST','Male_Workers','Female_Workers','Hindus','Muslims','Christians','Sikh
                       ,'Housholds_with_Electric_Lighting'], axis=1, inplace= True)
```

```
data.columns
```

```
Index(['District code', 'State name', 'District name', 'Population', 'Male',
       'Female', 'Literate', 'Male_Literate', 'Female_Literate', 'Workers',
       'Main_Workers', 'Marginal_Workers', 'Non_Workers', 'Cultivator_Workers',
       'Agricultural_Workers', 'Household_Workers', 'Other_Workers',
       'Households_with_Internet', 'Households_with_Computer',
       'Rural_Households', 'Urban_Households', 'Households',
       'Below_Primary_Education', 'Primary_Education', 'Middle_Education',
       'Secondary_Education', 'Higher_Education', 'Graduate_Education',
       'Other_Education', 'Literate_Education', 'Illiterate_Education',
       'Total_Education', 'Age_Group_0_29', 'Age_Group_30_49', 'Age_Group_50',
       'Age not stated', 'Power_Parity_Less_than_Rs_45000',
       'Power_Parity_Rs_45000_90000', 'Power_Parity_Rs_90000_150000',
       'Power_Parity_Rs_45000_150000', 'Power_Parity_Rs_150000_240000',
       'Power_Parity_Rs_240000_330000', 'Power_Parity_Rs_150000_330000',
       'Power_Parity_Rs_330000_425000', 'Power_Parity_Rs_425000_545000',
       'Power_Parity_Rs_330000_545000', 'Power_Parity_Above_Rs_545000',
       'Total_Power_Parity'],
      dtype='object')
```

Step 6 Exploring for insights at State level

```
fig = px.histogram(data,
                   x="State name",
                   y = "Population",
                   title='Population Vs States')
fig.update_layout(bargap=0.1)
fig.show()
```



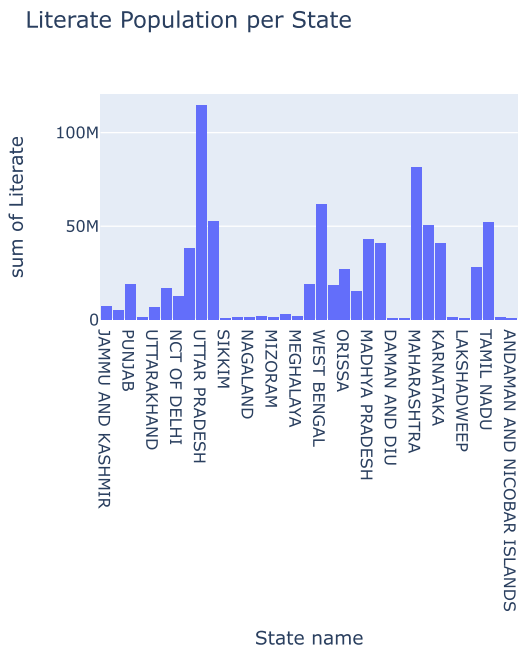
So there are many states which can be selected for our start up to launch their services in solely based on population count.

Most likely more business will be generssated from states like :

- Rajasthan
- Uttarpradesh
- Bihar
- West Bengal
- Madhya Pradesh
- Gujarat
- Maharastra
- Andhar Pradesh
- Karnataka
- Tamil Nadu

Note :- These are states with population greater than 50 Millions and this does not visualize whole scenario it is just a speculation based on Total population count of the above given states. Now, let's explore number of literate people residing in every state as literacy rate is directly Corrolated by regular medical check ups.

```
fig = px. histogram(data,
                    x = "State name",
                    y = "Literate",
                    title = "Literate Population per State")
fig.update_layout(bargap = 0.1)
fig.show()
```



Step 7 Exploring for Insights at District level

Firstly, we are going to make separate data frame for data of above listed states

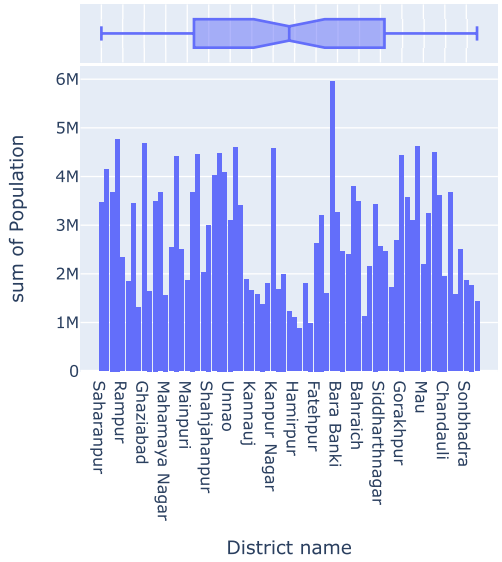
```
NCT_of_Delhi = data[data['State name'] == "NCT OF DELHI"]
Uttar_Pradesh = data[data['State name'] == "UTTAR PRADESH"]
West_Bengal = data[data['State name'] == "WEST BENGAL"]
Gujarat = data[data['State name'] == "GUJARAT"]
Maharashtra = data[data['State name'] == "MAHARASHTRA"]
Andra_Pradesh = data[data['State name'] == "ANDRA PRADESH"]
Karnataka = data[data['State name'] == "KARNATAKA"]
Kerala = data[data['State name'] == "KERALA"]
Tamil_Nadu = data[data['State name'] == "TAMIL NADU"]
```

It will be a very tedious task to write code for each and every state vise dataframes that we made recently. So we are going to define a function for that purpose.

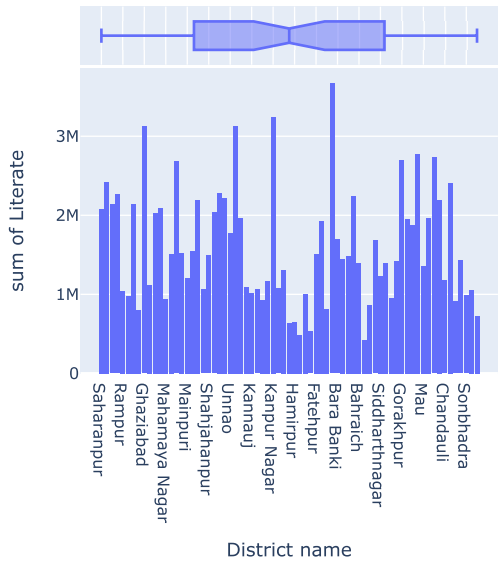
```
def Explore_districts_of(state):
    fig = px.histogram(state,
                       marginal = 'box',
                       x="District name",
                       y = "Population",
                       title='Population Vs Districts')
    fig.update_layout(bargap=0.1)
    fig.show()

    fig = px.histogram(state,
                       marginal = 'box',
                       x="District name",
                       y = "Literate",
                       title='Number of Literate Vs Districts')
    fig.update_layout(bargap=0.1)
    fig.show()

    fig = px.histogram(state,
                      marginal = 'box',
                      x = "District name",
                      y = "Households_with_Internet",
                      title = "Households with Internet in every District")
    fig.show()
```

```
Explore_districts_of(Uttar_Pradesh)
```

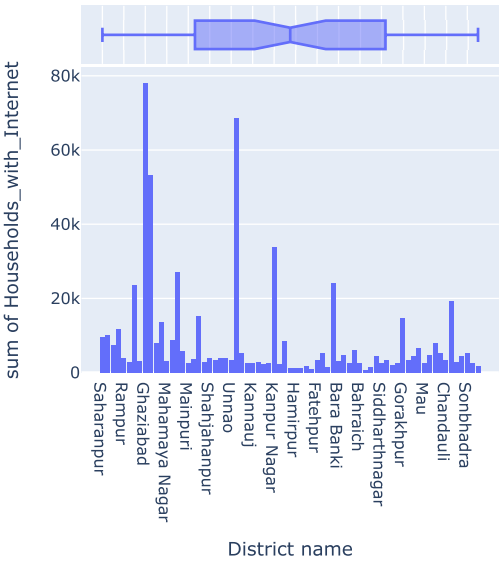
Population Vs Districts



Number of Literate Vs Districts



Households with Internet in every District



Step 8 Gathering Insights for few selected states

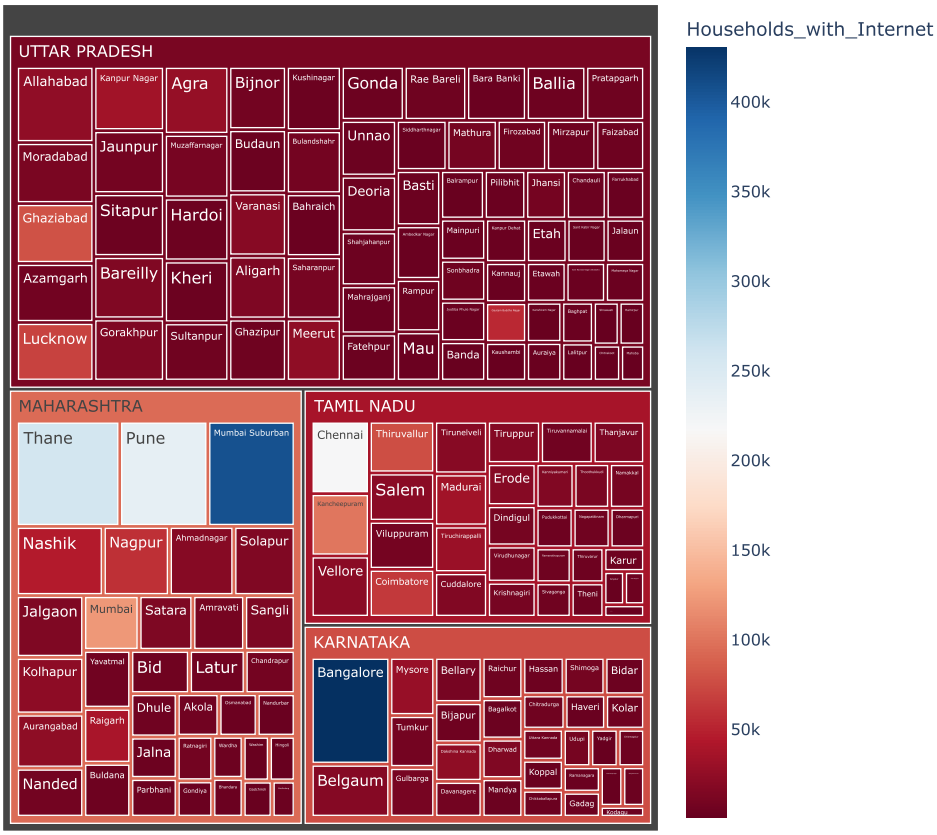
```
Selected_States = pd.concat([Uttar_Pradesh, Maharashtra, Tamil_Nadu, Karnataka], axis=0)
Selected_States
```

	District code	State name	District name	Population	Male	Female	Literate	Male_Literate	Female_Literate	Workers	...	Powe
131	132	UTTAR PRADESH	Saharanpur	3466382	1834106	1632276	2077108	1220114	856994	1037344	...	
132	133	UTTAR PRADESH	Muzaffarnagar	4143512	2193434	1950078	2417339	1448528	968811	1291644	...	
133	134	UTTAR PRADESH	Bijnor	3682713	1921215	1761498	2135393	1241471	893922	1088036	...	
134	135	UTTAR PRADESH	Moradabad	4772006	2503186	2268820	2263848	1357435	906413	1417811	...	
135	136	UTTAR PRADESH	Rampur	2335819	1223889	1111930	1043666	630408	413258	737261	...	
...	...	...	...	...	...	...	...	...	...	...	...	
579	580	KARNATAKA	Yadgir	1174271	590329	583942	510003	306751	203252	547696	...	
580	581	KARNATAKA	Kolar	1536401	776396	760005	1016219	564110	452109	717872	...	
581	582	KARNATAKA	Chikkaballapura	1255104	636437	618667	783222	442158	341064	639778	...	
582	583	KARNATAKA	Bangalore Rural	990923	509172	481751	688749	385311	303438	459891	...	
583	584	KARNATAKA	Ramanagara	1082636	548008	534628	674758	378461	296297	531459	...	

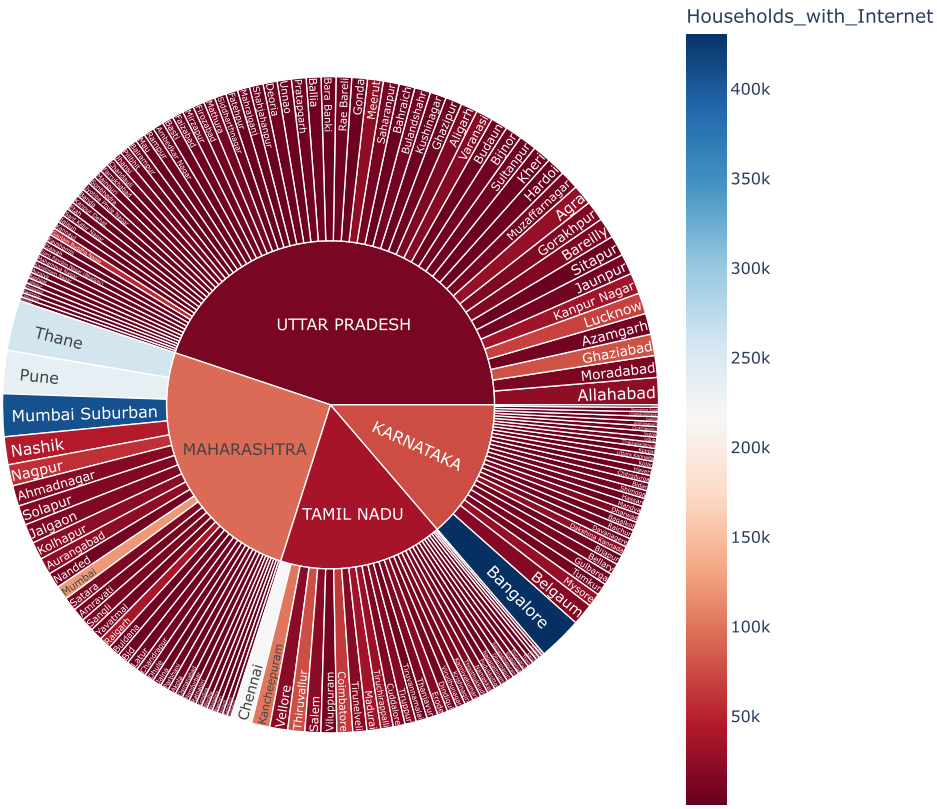
168 rows × 48 columns

```
fig = px.treemap(Selected_States,
                 path=['State name','District name'],
                 values='Population',
                 color='Households_with_Internet',
                 color_continuous_scale='RdBu',
                title = 'Finding out best Market')
fig.update_layout(bargap=1,autosize=False,
    width=800,
    height=800,)
fig.show()

fig = px.sunburst(Selected_States,
                 path=['State name','District name'],
                 values='Population',
                 color='Households_with_Internet',
                 color_continuous_scale='RdBu',
                title = 'Finding out best Market')
fig.update_layout(
    autosize=False,
    width=800,
    height=800)
fig.show()
```

Finding out best Market



Finding out best Market



larger the portion of district in above visuals shows larger total population and color inclination towards darker shades of blue means larger number of households with internet connection. So it seems there are five districts that looks like promising greater business opportunity for us. especially three of which are in Maharashtra.

Step 9 Recommendations based on our EDA

- Our company should encorrporate some advance data collection methods like foccus groups and mass public surveys to this states and specially to the states corresponding to five districts that show great promise of business gains.public surveys can be conducted online too by giving out incentives or discounts to customers that take part in it. By doing so we are also finalizing our customer base by marketing and also collecting data that can be used to create pyschographic profiles of the participants which will give us enough understanding of the local community, their values and their attitude towards online health services.
- This type of data collection needs to be done at a large scale to get rid of bias which is a very dangerous for our analysis.
- My personal judgement leans towards Maharashtran market as this state has more districts that are attractivve for our profits but also has quality population which has internet services and higher literacy rates.
- Marketing department should first penetrate larger cities which has denser population because more the density of population faster will be the word of mouth marketing like a wildfire spreading across dense forest.
- After establishing concrete business there we should move towards cities with lesser public and then towards the rural areas as rural segment is very hard to deal with for many reasons like providing fast customer service is very challenging, inventory storage in near by areas is very costly and also possibility of people adopting this change of online healthcare services is very less to allocate our resources to.

```
Selected_States.drop(['Power_Parity_Less_than_Rs_45000','Power_Parity_Rs_45000_90000','Power_Parity_Rs_90000_150000','Power_Parity_Rs_
Selected_States.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 168 entries, 131 to 583
Data columns (total 16 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   District code             168 non-null    int64 
 1   State name                168 non-null    object
 2   District name             168 non-null    object
 3   Population                168 non-null    int64 
 4   Male                      168 non-null    int64 
 5   Female                    168 non-null    int64 
 6   Literate                  168 non-null    int64 
 7   Households_with_Internet  168 non-null    int64 
 8   Households_with_Computer  168 non-null    int64 
 9   Rural_Households          168 non-null    int64 
 10  Urban_Households          168 non-null    int64 
 11  Households                168 non-null    int64 
 12  Age_Group_0_29            168 non-null    int64 
 13  Age_Group_30_49           168 non-null    int64 
 14  Age_Group_50              168 non-null    int64 
 15  Age not stated            168 non-null    int64 
dtypes: int64(14), object(2)
memory usage: 22.3+ KB
```

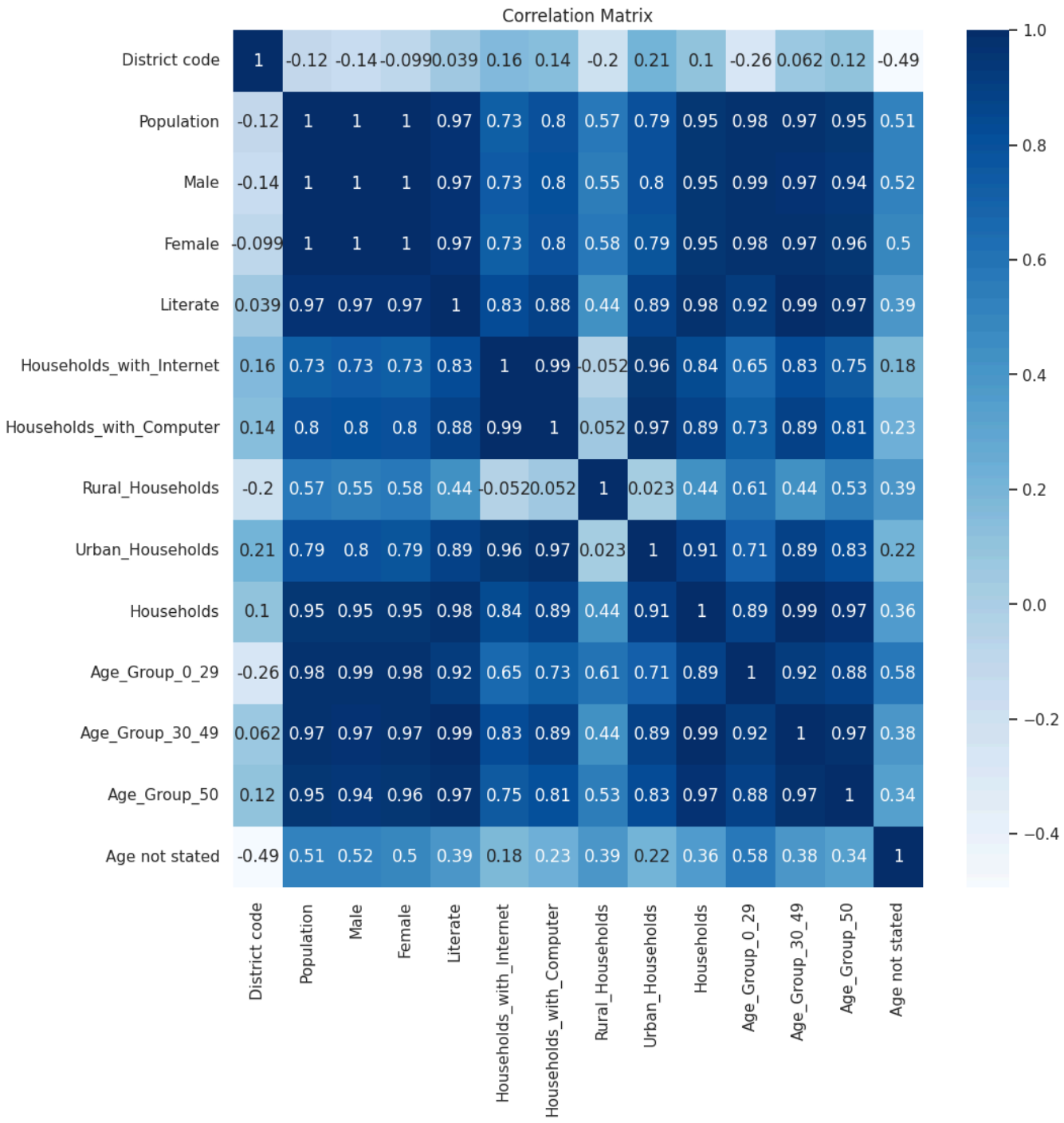
```
Selected_States.corr()
```

```
<ipython-input-163-15cfe9aa2c44>:1: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid
```

	District code	Population	Male	Female	Literate	Households_with_Internet	Households_with_Computer	F
District code	1.000000	-0.121331	-0.141542	-0.098584	0.038961	0.159186	0.142389	
Population	-0.121331	1.000000	0.999066	0.998844	0.969780	0.731503	0.800683	
Male	-0.141542	0.999066	1.000000	0.995832	0.966991	0.734823	0.802280	
Female	-0.098584	0.998844	0.995832	1.000000	0.970752	0.726202	0.797149	
Literate	0.038961	0.969780	0.966991	0.970752	1.000000	0.826475	0.882609	
Households_with_Internet	0.159186	0.731503	0.734823	0.726202	0.826475	1.000000	0.989140	
Households_with_Computer	0.142389	0.800683	0.802280	0.797149	0.882609	0.989140	1.000000	
Rural_Households	-0.204762	0.567404	0.554203	0.580840	0.440882	-0.051928	0.051724	
Urban_Households	0.209499	0.794356	0.797215	0.789432	0.887783	0.958281	0.972786	
Households	0.101705	0.951945	0.948948	0.953189	0.982464	0.838044	0.894741	
Age_Group_0_29	-0.260047	0.984909	0.986749	0.980700	0.918900	0.652522	0.726316	
Age_Group_30_49	0.061744	0.972116	0.969173	0.973254	0.992420	0.827668	0.885342	
Age_Group_50	0.119018	0.948381	0.940003	0.955617	0.973885	0.752498	0.813343	
Age not stated	-0.493138	0.509430	0.520352	0.496164	0.385775	0.181041	0.227474	

```
plt.figure(figsize=(11,11))
sns.heatmap(Selected_States.corr(), cmap='Blues', annot=True)
plt.title('Correlation Matrix')
```

```
<ipython-input-164-847717f52760>:2: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid
```

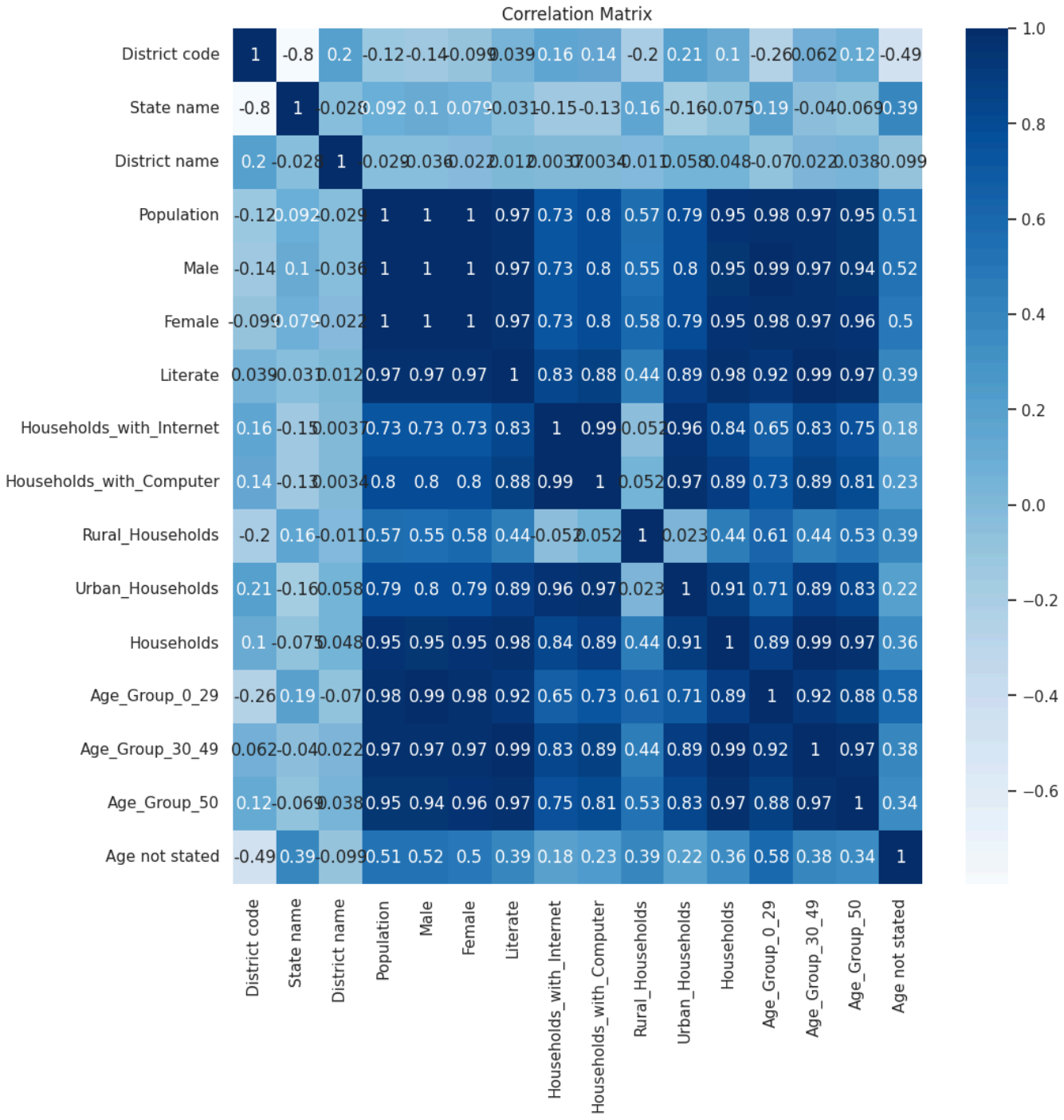



```
LB = LabelEncoder()
```

```
Selected_States['State name'] = LB.fit_transform(Selected_States['State name'])
Selected_States['District name'] = LB.fit_transform(Selected_States['District name'])
advance_data = Selected_States
scaler = StandardScaler()
segmentation_std = scaler.fit_transform(advance_data)
segmentation_std = pd.DataFrame(segmentation_std,columns=advance_data.columns)
advance_data.corr()
```

	District code	State name	District name	Population	Male	Female	Literate	Households_with_Internet	Househ
District code	1.000000	-0.795655	0.203548	-0.121331	-0.141542	-0.098584	0.038961	0.159186	
State name	-0.795655	1.000000	-0.027725	0.092411	0.104025	0.079290	-0.031222	-0.145091	
District name	0.203548	-0.027725	1.000000	-0.029396	-0.035803	-0.022206	0.012210	0.003727	
Population	-0.121331	0.092411	-0.029396	1.000000	0.999066	0.998844	0.969780	0.731503	
Male	-0.141542	0.104025	-0.035803	0.999066	1.000000	0.995832	0.966991	0.734823	
Female	-0.098584	0.079290	-0.022206	0.998844	0.995832	1.000000	0.970752	0.726202	
Literate	0.038961	-0.031222	0.012210	0.969780	0.966991	0.970752	1.000000	0.826475	
Households_with_Internet	0.159186	-0.145091	0.003727	0.731503	0.734823	0.726202	0.826475	1.000000	
Households_with_Computer	0.142389	-0.132109	0.003445	0.800683	0.802280	0.797149	0.882609	0.989140	
Rural_Households	-0.204762	0.163939	-0.011346	0.567404	0.554203	0.580840	0.440882	-0.051928	
Urban_Households	0.209499	-0.161085	0.058343	0.794356	0.797215	0.789432	0.887783	0.958281	
Households	0.101705	-0.075464	0.047573	0.951945	0.948948	0.953189	0.982464	0.838044	
Age_Group_0_29	-0.260047	0.188273	-0.069565	0.984909	0.986749	0.980700	0.918900	0.652522	
Age_Group_30_49	0.061744	-0.039677	0.022360	0.972116	0.969173	0.973254	0.992420	0.827668	
Age_Group_50	0.119018	-0.068558	0.038497	0.948381	0.940003	0.955617	0.973885	0.752498	
Age not stated	-0.493138	0.391986	-0.098979	0.509430	0.520352	0.496164	0.385775	0.181041	

```
plt.figure(figsize=(11,11))
sns.heatmap(advance_data.corr(), cmap='Blues', annot=True)
plt.title('Correlation Matrix')
```

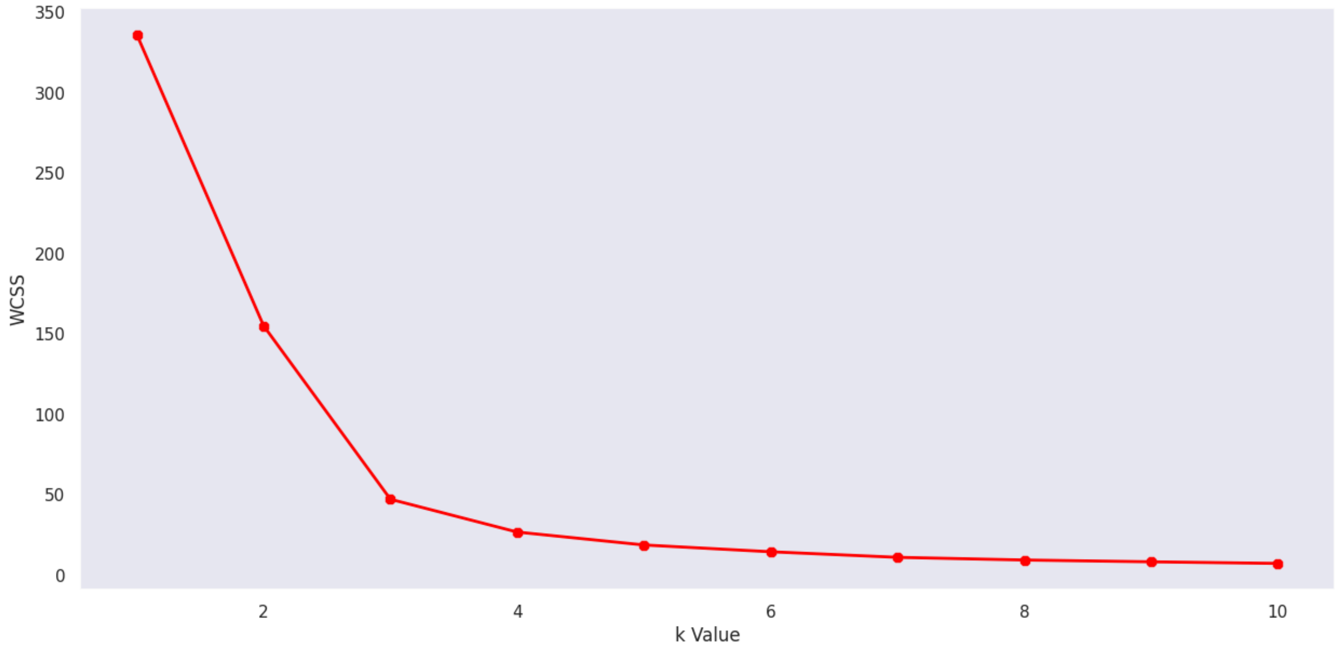



```
segmentation_std= pd.DataFrame(segmentation_std)
print(segmentation_std.max())
```

```
District code               1.179951
State name                  0.992915
District name               1.721771
Population                  5.202932
Male                        5.278704
Female                      5.107227
Literate                    5.424253
Households_with_Internet    7.381445
Households_with_Computer    7.382053
Rural_Households            3.433510
Urban_Households            6.289856
Households                  5.833210
Age_Group_0_29              4.936360
Age_Group_30_49             5.685461
Age_Group_50                4.571477
Age not stated              4.137599
dtype: float64
```

```
X1 = segmentation_std.loc[:, ["Population","Literate"]].values

from sklearn.cluster import KMeans
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++')
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(15,7))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2, color='red', marker="8")
plt.xlabel('k Value')
plt.ylabel('WCSS')

plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```



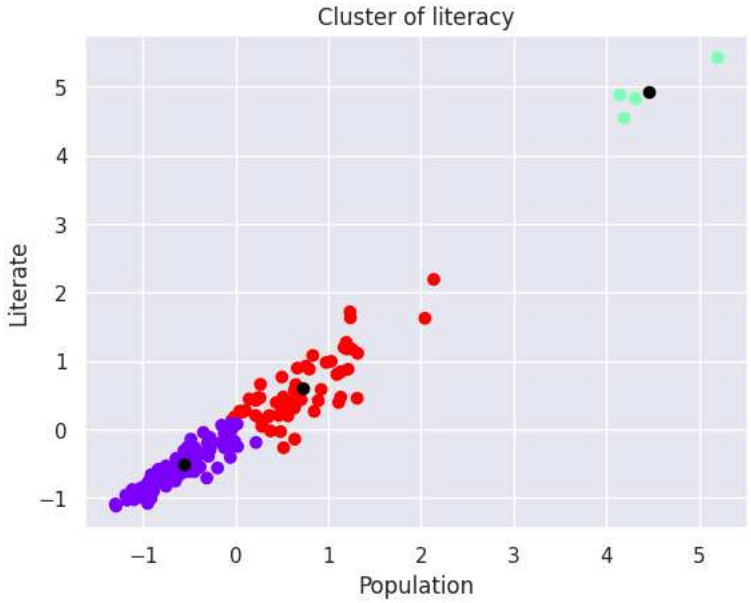

```
kmeans = KMeans(n_clusters= 3)
label = kmeans.fit_predict(X1)
# print(label)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
print(kmeans.cluster_centers_)
```

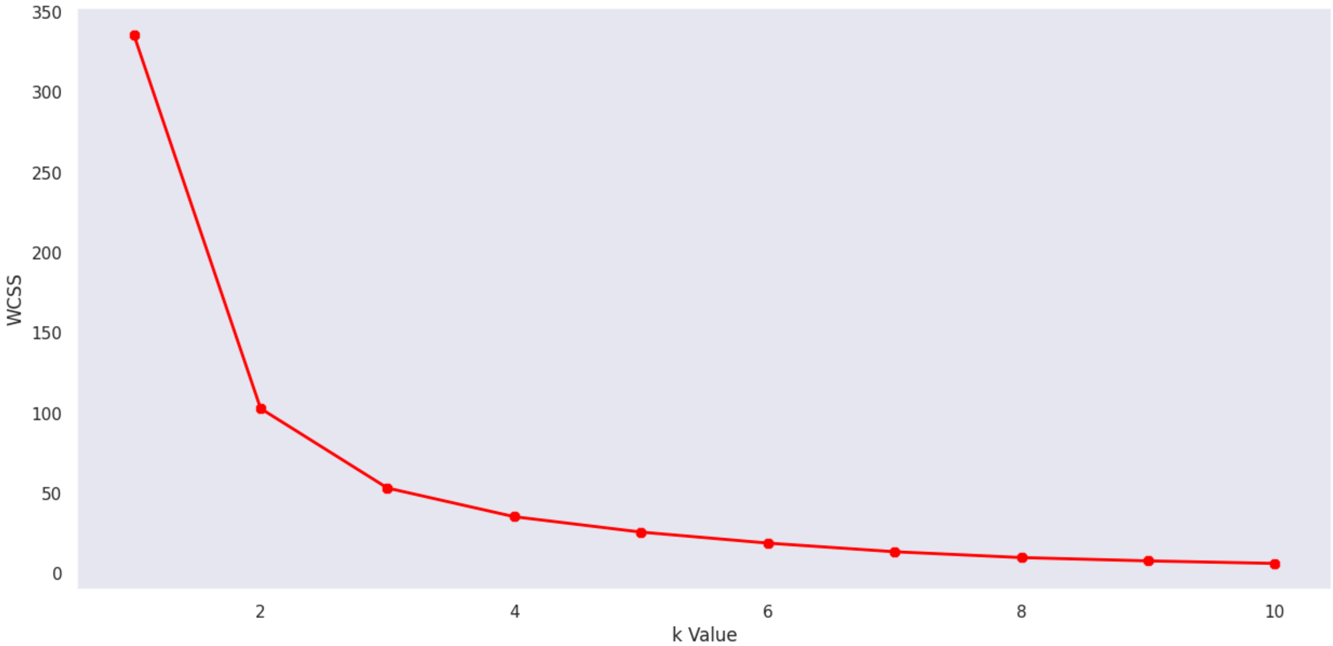
```
[[-0.56327527 -0.51218995]
 [ 4.46468283  4.91976113]
 [ 0.72152495  0.59677741]]
```

```
plt.scatter(X1[:,0], X1[:,1], c=kmeans.labels_,cmap= 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color='black')
plt.title('Cluster of literacy')
plt.xlabel('Population')
plt.ylabel('Literate')
plt.show()
```



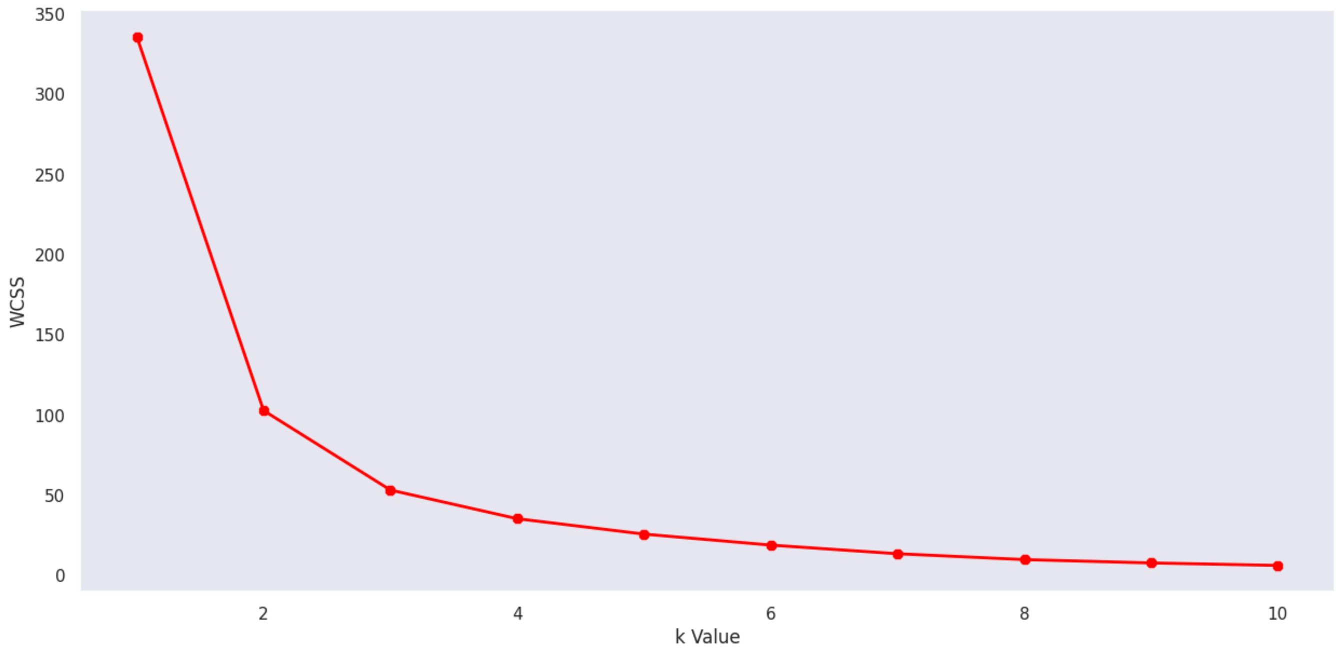

```
X1 = segmentation_std.loc[:, ["Households_with_Internet","Literate"]].values

from sklearn.cluster import KMeans
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++')
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(15,7))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2, color='red', marker="8")
plt.xlabel('k Value')
plt.ylabel('WCSS')

plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```



```
X1 = segmentation_std.loc[:, ["Households_with_Internet","Literate"]].values

from sklearn.cluster import KMeans
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++')
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(15,7))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2, color='red', marker="8")
plt.xlabel('k Value')
plt.ylabel('WCSS')

plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

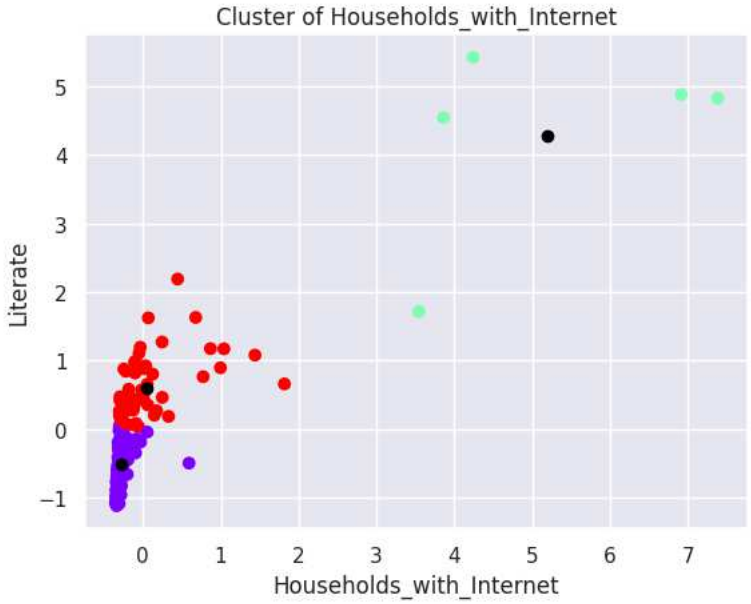



```
kmeans = KMeans(n_clusters= 3)
label = kmeans.fit_predict(X1)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

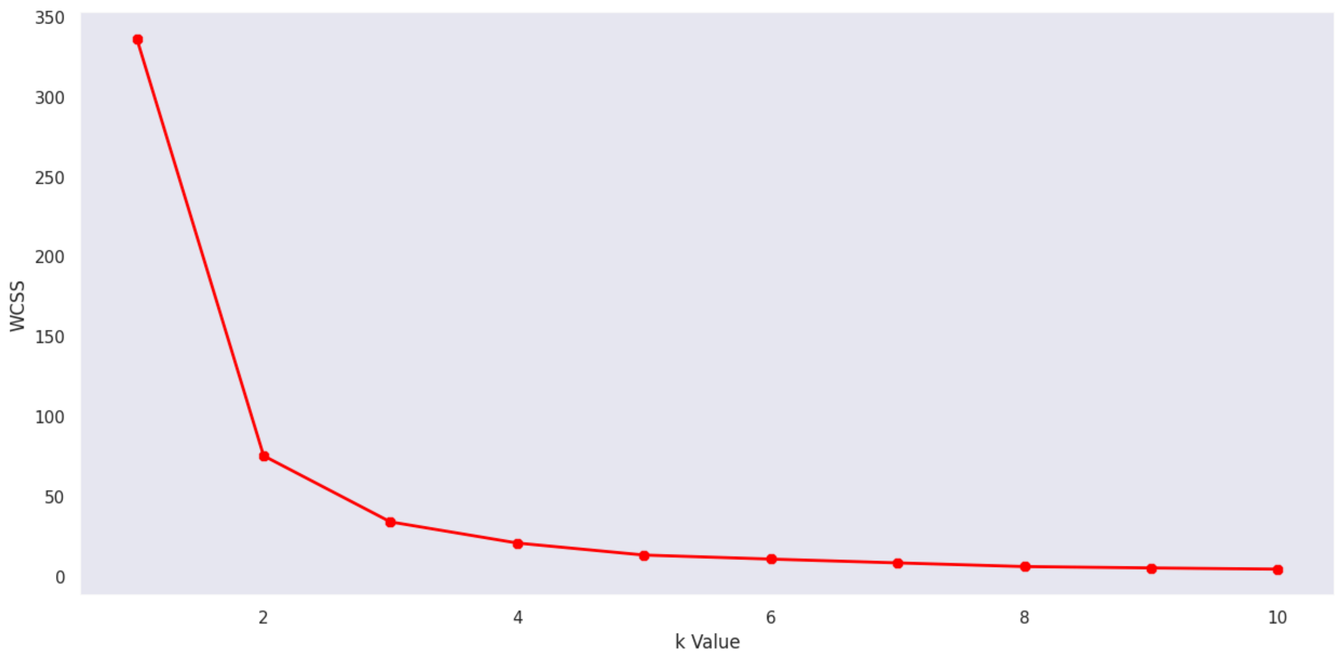
```
print(kmeans.cluster_centers_)
plt.scatter(X1[:,0], X1[:,1], c=kmeans.labels_,cmap= 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color='black')
plt.title('Cluster of Households_with_Internet')
plt.xlabel('Households_with_Internet')
plt.ylabel('Literate')
plt.show()
```

```
[[-0.27097743 -0.51980044]
 [ 5.18770938  4.27897896]
 [ 0.04886071  0.5912974 ]]
```




```
X1 = segmentation_std.loc[:, ["Urban_Households","Households_with_Computer"]].values

from sklearn.cluster import KMeans
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++')
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(15,7))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2, color='red', marker="8")
plt.xlabel('k Value')
plt.ylabel('WCSS')

plt.show()
```

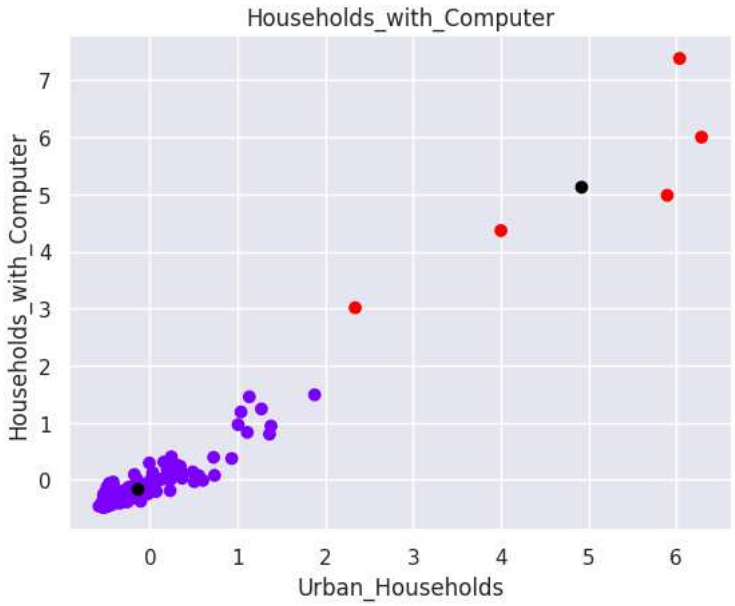
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```




```
kmeans = KMeans(n_clusters= 2)
label = kmeans.fit_predict(X1)
```

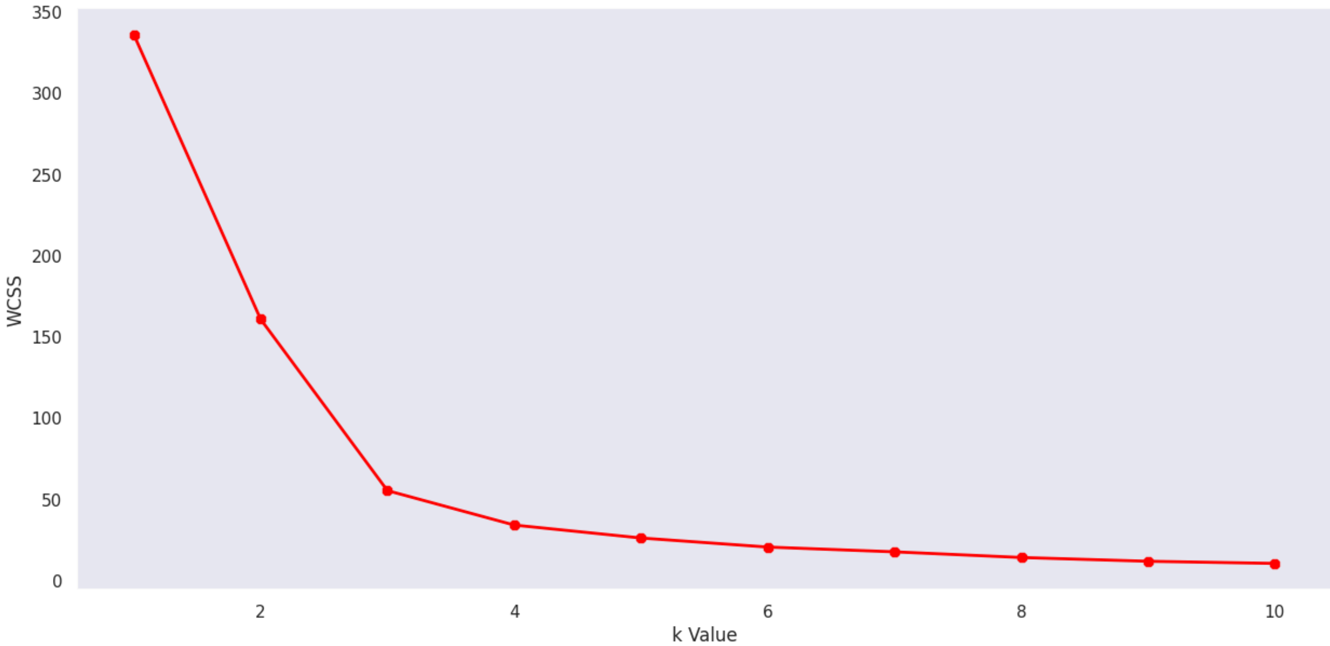
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
plt.scatter(X1[:,0], X1[:,1], c=kmeans.labels_,cmap= 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color='black')
plt.title('Households_with_Computer')
plt.xlabel('Urban_Households')
plt.ylabel('Households_with_Computer')
plt.show()
```




```
X1 = segmentation_std.loc[:, ["Age_Group_0_29","Literate"]].values

from sklearn.cluster import KMeans
wcss = []
for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, init='k-means++')
    kmeans.fit(X1)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(15,7))
plt.grid()
plt.plot(range(1,11),wcss, linewidth=2, color='red', marker="8")
plt.xlabel('k Value')
plt.ylabel('WCSS')

plt.show()
```

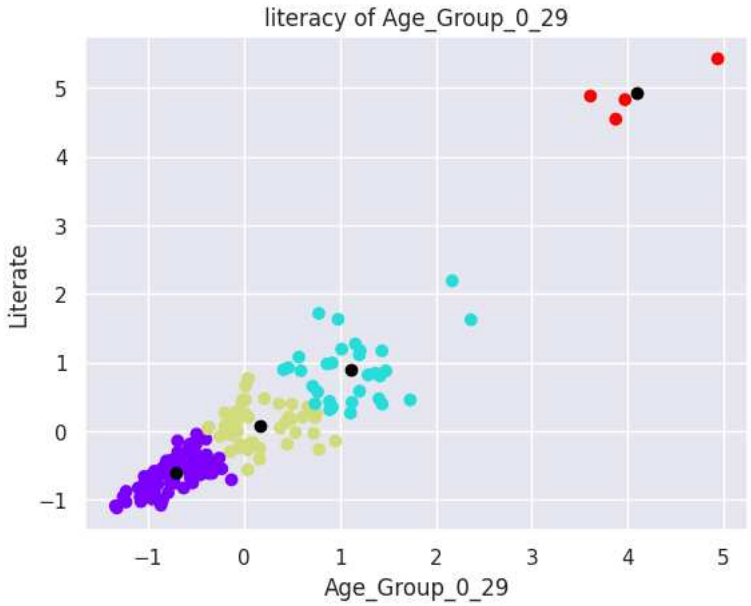
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning

/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```




```
kmeans = KMeans(n_clusters= 4)
label = kmeans.fit_predict(X1)
```

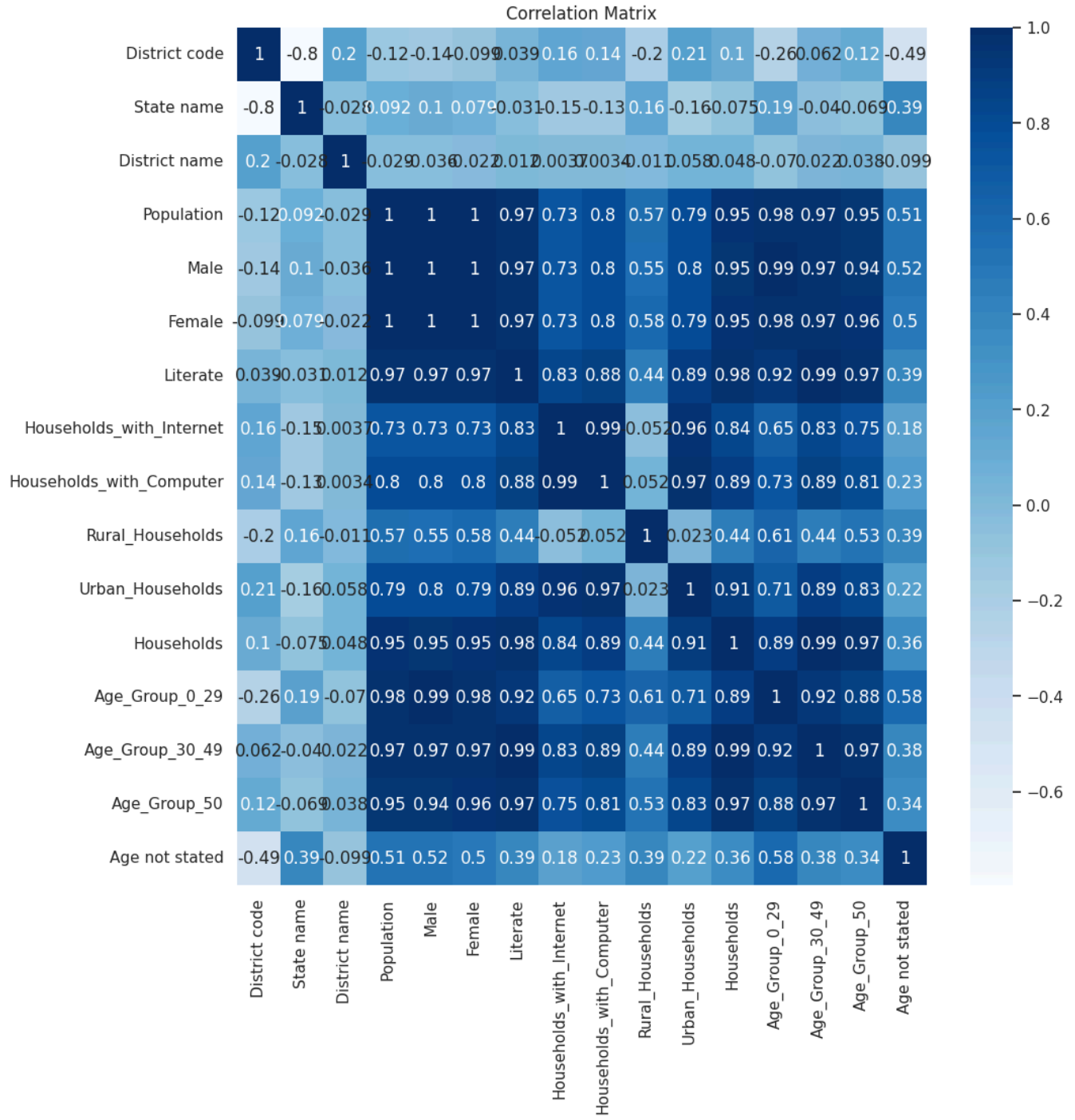
```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning:

The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
```

```
plt.scatter(X1[:,0], X1[:,1], c=kmeans.labels_,cmap= 'rainbow')
plt.scatter(kmeans.cluster_centers_[:,0], kmeans.cluster_centers_[:,1], color='black')
plt.title('literacy of Age_Group_0_29')
plt.xlabel('Age_Group_0_29')
plt.ylabel('Literate')
plt.show()
```

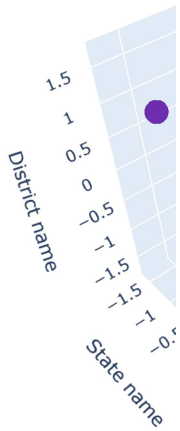



```
plt.figure(figsize=(11,11))
sns.heatmap(advance_data.corr(), cmap='Blues', annot=True)
plt.title('Correlation Matrix')
```

```
Text(0.5, 1.0, 'Correlation Matrix')
```



```
x = Selected_States[['District code',   'State name', 'District name', 'Population', 'Male', 'Female', 'Literate',
          'Households_with_Internet', 'Households_with_Computer', 'Rural_Households', 'Urban_Households',
          'Households', 'Age_Group_0_29', 'Age_Group_30_49',    'Age_Group_50']].values
km = KMeans(n_clusters = 15, init = 'k-means++', max_iter = 300, n_init = 10, random_state = 0)
km.fit(x)
labels = km.labels_
centroids = km.cluster_centers_
segmentation_std['labels'] =  labels
trace1 = go.Scatter3d(
    x= segmentation_std['Population'],
    y= segmentation_std['State name'],
    z= segmentation_std['District name'],
    mode='markers',
     marker=dict(
        color = segmentation_std['labels'],
        size= 10,
        line=dict(
            color= segmentation_std['labels'],
            width= 12
        ),
        opacity=0.8
     )
)
df = [trace1]

layout = go.Layout(
    title = 'population in States as well as District',
    margin=dict(
        l=0,
        r=0,
        b=0,
        t=0
    ),
    scene = dict(
            xaxis = dict(title  = 'Population'),
            yaxis = dict(title  = 'State name'),
            zaxis = dict(title  = 'District name')
        )
)

fig = go.Figure(data = df, layout = layout)
iplot(fig)
```



```
segmentation_std['labels'] =  labels
trace1 = go.Scatter3d(
    x= segmentation_std['Households'],
    y= segmentation_std['Rural_Households'],
    z= segmentation_std['Urban_Households'],
    mode='markers',
     marker=dict(
        color = segmentation_std['labels'],
        size= 10,
        line=dict(
            color= segmentation_std['labels'],
            width= 12
        ),
        opacity=0.8
     )
)
df = [trace1]

layout = go.Layout(
    title = 'Households in rural and urban',
    margin=dict(
        l=0,
        r=0,
        b=0,
        t=0
    ),
```

5. Benchmarking Alternate Products

For all the reasons stated throughout this report, dermatology has become an attractive space for app innovation. A study in 2017, reported a total of 526 dermatological mobile apps which corresponded to an 80.8% growth since 2014 (Flaten et al., 2018). Hence it is important to look over at the noteworthy available market alternatives as per (Mesko, 2023).

- CureSkin:
    - Targeted operating location: India
    - Features: Provides personalized skincare solutions using AI. Allows users to submit images of their skin conditions for analysis.
    - Strength: The app claims to offer customized treatment plans based on AI analysis and consultations with dermatologists.
    - Consideration: The effectiveness of features like tracking treatment progress and reminders for medication vary with individual use. Promotes its own product line.
    - Languages: 3 Indian Languages – Hindi, Kannada and Telugu only
- FirstDerm:
    - Targeted operating location: Global
    - Features: Offers remote dermatology consultations and skin condition information.
    - Strength: Access to dermatologists for consultation
    - Consideration: May require a fee for consultations
    - AI usage: Limited
    - Languages: 6 global languages
- Skin Vision:
    - Targeted operating location: Europe predominantly
    - Features: Provides skin cancer risk assessment using AI.
    - Strength: Focus on early detection of skin cancer
    - Consideration: Requires a subscription for full access.
    - Languages: English, Dutch and German only

- DermEngine:
    - Targeted operating location: Global
    - Features: Offers AI-driven skin imaging and analysis.
    - Strengths: Focus on skin imaging and AI analysis, Anytime (24/7)
    - Consideration: May cater more to healthcare providers
    - Languages: English, French, Italian, Spanish and Portuguese only
- Ping An Good Doctor:
    - Targeted operating location: China
    - Features: Comprehensive healthcare app including dermatology services.
    - Strengths: Part of a larger telemedicine platform.
    - Consideration: Not a specialist app - More expansive than dedicated dermatology apps as they tackle a host of non- skin related conditions.
    - Languages: Chinese only

Healthcare gap in Dermatology in India: Explainable deep-learning based AI early-diagnostic (detection) app that has high explainability of outputs to dedicated for India, to address the cultural and linguistic diversity. Additionally, serving as a link between users, healthcare specialists and insurance institutions.

6. Applicable Regulations

Developing and deploying a dermatological diagnosis app using deep learning in India requires adherence to several regulations, encompassing both government and environmental considerations. Ensuring compliance with these regulations throughout the business venture is not only a legal requirement but also essential for building trust with users and stakeholders. Key applicable regulations include (Das et al., 2023):

A. Data Protection and Privacy:

- Personal Data Protection Bill (PDPB): India is in the process of enacting comprehensive data protection legislation. The PDPB aims to regulate the processing of personal data and impose obligations on entities handling such

data. Compliance with this law is critical to ensure the protection of user health data.

- General Data Protection Regulation (GDPR): If the app processes data of users residing in the European Union, compliance with GDPR is necessary. Even though it's an EU regulation, its extraterritorial scope affects businesses worldwide.

B. Health Data Regulations:

- Electronic Health Records (EHR) Standards: Ensure compliance with any national and/or state-level standards for electronic health records as the app will be storing and managing user health-related information.

- Medical Council of India (MCI) Guidelines: Since the app aims to collaborate with dermatological professionals, it must adhere to guidelines set by the Medical Council of India to maintain ethical standards in healthcare practice.

C. Drug and Cosmetic Act:

- Central Drugs Standard Control Organization (CDSCO): The app might provide information related to pharmaceutical products, it should comply with the regulations governed by CDSCO under the Drug and Cosmetic Act.

D. Telemedicine Guidelines:

- Telemedicine Practice Guidelines: The Ministry of Health and Family Welfare in India has released guidelines for telemedicine. The app will involve virtual consultations, hence adherence to these guidelines is critical.

E. Cybersecurity Regulations:

- Indian Computer Emergency Response Team (CERT-In): Dealing with sensitive health data means compliance with cybersecurity regulations is necessary to protect user data from cyber threats. CERT-In provides guidelines and standards for information security.

F. Ethical AI and Algorithmic Transparency:

- NITI Aayog's AI Policy Framework: The National Institution for Transforming India (NITI Aayog) has detailed the guidelines on ethical AI. App developers should ensure transparency in algorithms, fairness, and accountability in AI systems.

G. Accessibility Standards:

- Web Content Accessibility Guidelines (WCAG): As a health-related app, to ensure that it is accessible to all individuals it must be considerate of users with disabilities by following WCAG standards. This would include considerations for users with visual or auditory impairments.

H. Consumer Protection Laws:

- Consumer Protection Act: Adhere to consumer protection laws to ensure fair and transparent practices, including providing accurate information about the app's capabilities and limitations.

7. Applicable Constraints

A. Technical Expertise: Developing and implementing deep learning algorithms requires specialized technical expertise in AI, machine learning, and image recognition.

B. Data limitations: A detailed and extensive health dataset is necessary to develop an accurate and result-explainable model. Creating, obtaining the necessary permissions and meeting necessary regulations for handling sensitive confidential health data will be tricky. Poorly labelled data will also lead to the outputs reflecting those biases and inaccuracies. Additionally to meet the explainability requirement the data must be labelled at the physician level.

C. Legal expertise: The extensive list of healthcare-related regulations means it might be necessary to consult legal experts to ensure adherence.

D. Model accuracy measure: Such a metric is especially tricky considering the app seeks to explain conclusions drawn to a high degree.

E. Accessibility features: Multilingual Support is necessary to reflect and cater to the linguistic diversity in India, making the app accessible to a wider audience. Especially necessary since a large target audience lies in rural regions. Poor connectivity and networks in these areas mean the app would need to be optimized for low bandwidth, ensuring usability in regions with limited internet connectivity.

8. Business Opportunity

Provide individualized skin cancer detection services and products based on differences in the population and prevalence of skin cancer by location. This will guarantee that activities are appropriate and culturally relevant for the local community. Create telehealth apps that make use of AI-enhanced skin cancer detection capabilities. This will enable medical professionals to assess problematic lesions from a distance and advise primary care physicians. This will facilitate easier access to expert dermatological care.

Permit insurance companies to detect patients who pose a higher risk by adding them to their profile. enables patients to find companies that, if medical intervention is required, can provide support for their care based on a professional diagnosis. Additionally, if applicable, it would enable them to maximize the use of their present plans at an earlier stage of therapy.

Integration with Healthcare Providers:

- Physician Collaboration: The application must to enable communication and coordination between dermatologists and other medical professionals. This indicates that the software enhances rather than replaces conventional healthcare services.

- Referral System: Provide a smooth referral process for patients in need of in-person consultations to guarantee a comprehensive approach to medical care.

Combining insurance companies:

Cooperate with nearby insurance companies: Permit insurance companies to recognize high-risk individuals who add to their profile. gives patients the option to choose businesses that, in the event that medical intervention is required, can assist with their care. Moreover, it would enable patients to optimize the utilization of their existing plans, if relevant, during an earlier phase of therapy.

Final Product Prototype/ Product Details:

A. Algorithms of interest:

As a diagnosis tool, the primary focus lies on Classification based algorithms (Chan et al., 2020):

- **Convoluted Neural Networks (CNN)**: A branch of deep learning that mimics how neurons process information by adding more convolutions to the traditional Artificial neural networks (ANNs). It breaks down the image into its fundamental component pixels. The model proceeds to compare and contrast sub-class features of the input image. Finally pooling information across to classify input image. A branch of CNN called region-based CNN (r-CNN) can hone in on a desired object within the image. This in particular is of crucial importance in skin disease detection using AI.

- **k-nearest neighbours (KNN)**: Used for data classification and regression based on the number of k neighbours. Can be used to identify at risk patient if their data is close to that of a statistically diagnosed patient (data point).

- **Support vector machine (SVM)**: They are used in data classification by finding a hyperplane to differentiate between groups. Serving a similar role as a classifier to identify at-risk patients based on training data.

- **Logistic regression**: Several risk factors will be of binary format hence it is worth considering a discriminative model such as logistic regression. It distinguishes between classes such as binary data (true or false).

- **Random Forest:** A simple yet powerful tool of ensemble learning used for classification that utilizes the strength in many approaches to construct several decision trees during training returning the most common denominator. Effective in reducing overfitting to a single curated model.

B. Team required to develop:

- Deep ML engineers
- Healthcare professionals
- Business analyst
- Software engineer
- UI/UX developer
- Cloud engineer
- Big Data Researcher
- Legal expert

C. A structure that makes sense:

The Data Science department reports to the Product (Gavish, 2022)

- Allows the product to be the driver, this structure allows full alignment of goals and desirables. Creates a level of transparency to the product head which helps achieve business outcomes effectively.
- Prerequisite: Linkage between the teams to dumb down the technical jargon. A product head that understands the importance of understanding the underlying infrastructure to an extent.

9. Concept Generation and Development

A. Market Research – Expanding on the current report:

- Identify the current landscape of dermatological diagnosis apps in India.
- Understand user needs, pain points, and preferences.
- Analyse competitors and existing solutions to identify gaps.

B. Define Objectives:

- Clearly define the objectives of the app, such as early detection, accessibility, or remote consultations.
- Identify specific dermatological conditions the app will focus on.

C. Detailed User/Consumer Survey:

- Create detailed user personas to understand the characteristics, preferences, and challenges of potential users in India.

D. Ideation Sessions:

- Conduct brainstorming sessions with a multidisciplinary team, including AI experts, UX/UI designers, dermatologists and legal experts.
- Problem-Solution Fit: Addressing problems identified
- The interdisciplinary approach encourages diverse perspectives to generate a wide range of ideas.

E. User-Friendly Design:

- Develop ideas for a user-friendly interface that accommodates users with varying levels of tech literacy.

- Ensure the design is culturally sensitive and inclusive of diverse populations in India.

F. AI Algorithm Integration:

- Explore different deep-learning algorithms for dermatological image analysis.
- Consider continuous learning mechanisms to improve accuracy over time.
- Focus on Explainable AI to maximise healthcare transparency

G. Collaboration with Healthcare Professionals:

- Identify exact opportunities for collaboration with dermatologists and/or other healthcare providers.
- Develop features that facilitate seamless communication between users and healthcare professionals.

H. Data Security and Privacy:

- Devise strategies to address data security and privacy concerns.
- Consider encryption, secure storage, and compliance with the relevant regulations mentioned.

I. Multilingual Support:

- Ensure the app supports multiple languages to address the linguistic diversity in India.

J. Offline Functionality:

- Look into app functionality in areas with limited internet connectivity.

K. Pilot Testing:

- Develop a prototype of the app to conduct pilot testing with a selected small user group.
- Gather feedback to refine the concept based on user experiences.

L. Productization

- Stabilize the model for the use case. Then scale the model and data collection to an open (not curated) bigger user base.
- Check for outliers

M. Iterative Refinement:

- Continuously improvement of the app based on user feedback, technological advancements, and changes in the healthcare landscape.

10. Final Product Prototype/ Product Details

Step 1: Business Fundamentals

A) Feasibility:

This project can be created and distributed for public use within a few years. As the number of cases grows, rulers and urban residents will benefit from this. The full description is available in the business model.

B) Viability:

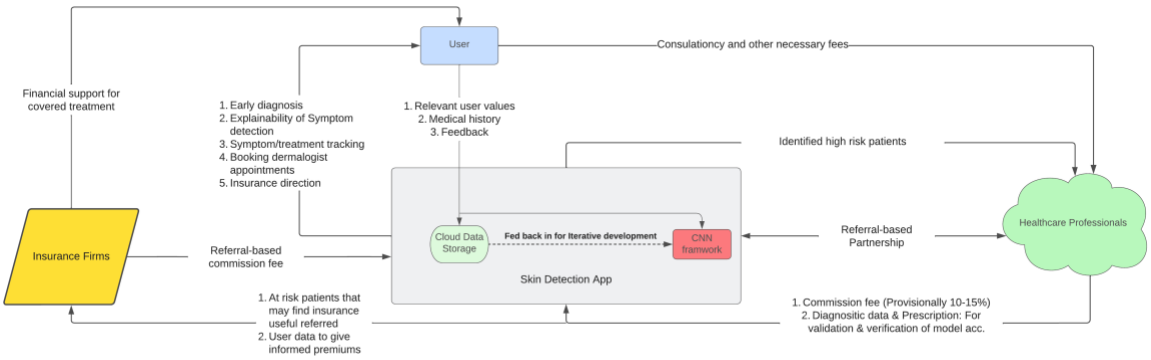
In India, initiatives to tackle skin disorders include public health programs, cleanliness and skincare education, increased access to healthcare facilities, and research into effective treatments and prevention methods. Additionally, dermatologists and other healthcare professionals play a vital role in diagnosing and treating a variety of skin problems. We have a limited number of dermatologists, but the number and availability

of dermatologists will need to rise in the future as people become more interested in UV radiation and, according to surveys, use chemicals more frequently.

C) Monetization:

This service is monetizable because it is directly related to people's skin issues (these days), which individuals should never overlook when considering which businesses to utilize.

Step 2: Final Product Prototype Diagram



Step 3: Business Modelling

A teledermatologist's block diagram consists of several parts and procedures to enable remote dermatological consultations. Typically, the diagram has components like image analysis systems, patient information management, communication technology, and expert consultations. Based on the search results, the following is a quick summary: A suggested teledermatology approach's block diagram shows the parts of a smartphone-based teledermoscopy system. Automatic ECG data processing is the main focus of a telemedicine system block diagram used to identify cardiac abnormalities. These diagrams depict the integration of medical imaging equipment, communication tools, and professional interactions to efficiently deliver remote dermatological care. They also highlight the technology infrastructure and processes involved in teledermatology consultations.

An application that connects with the current healthcare infrastructure, is safe, user-friendly, and sensitive to cultural differences is necessary for successful implementation. The application satisfies this requirement, which could enhance dermatological care outcomes and accessibility for a variety of Indian demographics.

The rate of adoption, accuracy, and accessibility of online dermatological diagnostics will only increase. Boulders that are still important are the scarcity of data and the importance of model bias. Nonetheless, there is a chance for a constantly accessible, quick, and reliable response to the most recent research. Moreover, their labour is unpaid, in contrast to that of doctors. In dermatology, this would enable us to detect potential issues as soon as feasible. Being an early adopter in the field of artificial intelligence disruption makes sense given the potential for scalable growth.

11. Financial Modelling

Financial modelling for a dermatological diagnosis app involves projecting revenue, costs, and profitability over a certain period. To understand the growth trend of the industry we observe the projected average revenue in Skin treatment in India.

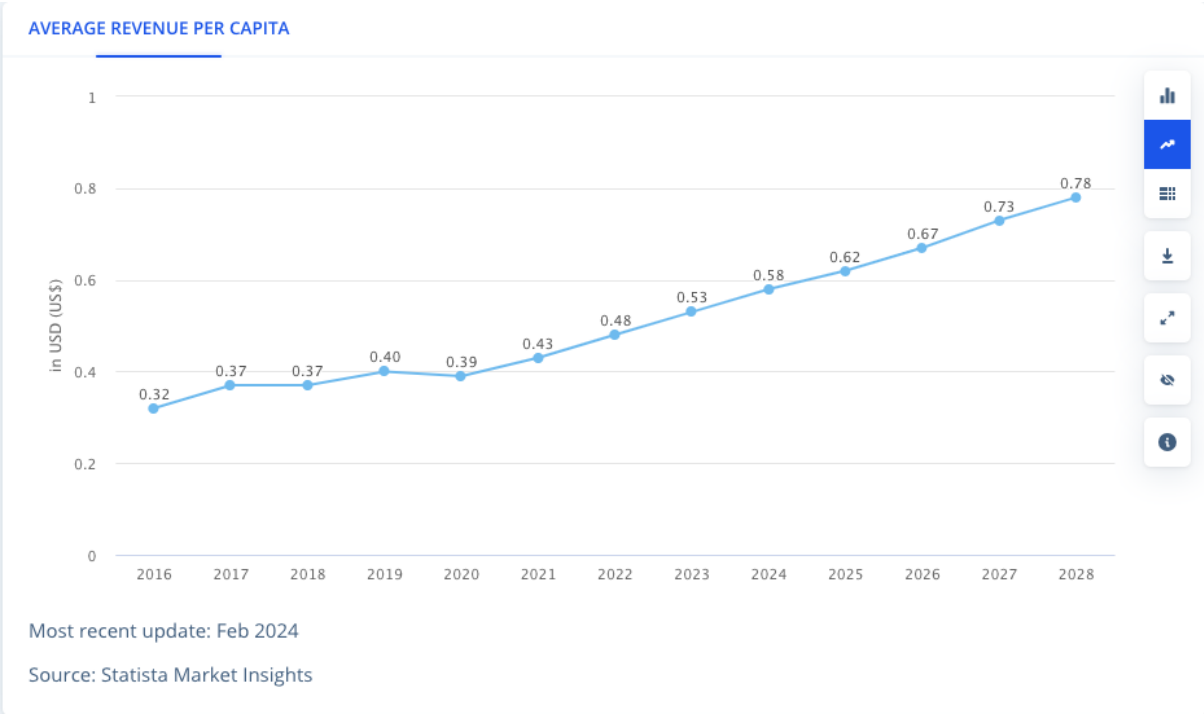


Fig. Projected Average Revenue per Capita for Skin treatment in India (2016-2028) [6]

The projected average revenue in Skin treatment since 2020 paint an extremely linear growth.

- Let total profit = y
- Price of product = m
- Total sale as a function of time = x(t)
- Total production & maintenance cost = c

Assumptions:

1. User Base Projection:
    - Start with the current population of smartphone users in India and estimate the percentage of individuals interested in dermatological diagnosis.
    - Assume a linear growth rate in the user base over the projection period.
2. Revenue Model:
    - Consultation Fees: Revenue generated from virtual consultations with dermatologists.
    - Data Sales to Insurance Firms: Revenue earned by selling aggregated and anonymized user data to insurance firms for risk assessment.
3. Cost Structure:
    - Development Costs: Initial investment in app development, AI algorithm integration, and infrastructure setup.
    - Operating Costs: Ongoing expenses such as server maintenance, marketing, customer support, and regulatory compliance.
    - Dermatologist Fees: Payments to dermatologists for virtual consultations.
4. Insurance Firm Partnerships:
    - Revenue generated from partnerships with insurance firms, which may involve a percentage of premiums based on referrals or access to user data.

Provisional Financial Model Outputs:

User Base and Revenue Projection:

- Year 1: 10,000 users
- Year 2: 20,000 users
- Year 3: 30,000 users
- Consultation Fees: INR 500 per consultation
- Data Sales to Insurance Firms: INR 300,000 per year

Cost Projection:

Potential production costs would include –

- Fixed Costs: The total cost of operating the business, including rent, salaries, and other recurring expenses.
- Variable Costs: The cost of providing the services, such as the cost of supplies and equipment.
- Servers and software cost
- Office cost: Can be assumed null as a start-up with a work-from-home model

Development Costs: INR 4,000,000
Operating Costs: INR 500,000 per year

Profitability Analysis:

Net profit by subtracting total costs from total revenue:

- Year 1: INR 800,000
- Year 2: INR 9,800,000
- Year 3: 14,800,000

Financial Equation: y = 500*x1(t) + (300,000*t - 500,000*t - 4,000,000)

Note: Operational cost and insurance based earnings are time related as they are recurring annually.

Key Assumptions:

- Number of consultations yearly is equal to the number of users at the time
    - Implies most current users have at least one consultation
    - Need for consultation on a yearly base
- A constant operating cost – unlikely as a growing business has growing needs (example: larger servers to store more data from a growing consumer base)
- A constant inflow from insurance firms referral and data sales

Room for growth: Additional subscription-based brackets for premium service and potential for exponential growth in the industry considering the growing public intrest.

Additionally, profitability metrics such as return on investment (ROI) and payback period could be utilised to assess the business growth and turnover further.

Investment Requirements:

- Initial Investment: INR 4,000,000
- Breakeven Analysis: Determine the time it takes to recoup the initial investment based on projected cash flow.

Considerations:

- Validate assumptions with market research and industry experts.
- Continuously update the financial model with actual data to refine projections and make informed business decisions.
- Monitor key performance indicators (KPIs) such as user acquisition cost, customer lifetime value, and revenue growth to track the app's performance over time.

12. Conclusion

The findings from the business and financial modelling, coupled with health insurance and dermatological disease demographical segmentation, underscore a compelling opportunity for a dermatological diagnosis app using deep learning in India. By leveraging AI technology, the app can address critical gaps in dermatological care, particularly in underserved areas, leading to improved early detection and treatment outcomes.

The next phase in the evolution of AI, particularly in the health industry, involves unravelling the complexities of the "black-box" algorithms to enhance transparency and trust. With the increasing digitalization and data tracking, larger and more detailed datasets will become available, paving the way for more sophisticated and powerful models.

The accessibility, accuracy, and adoption rate of online dermatological diagnosis are poised to continue rising. However, challenges such as data limitations and model bias remain significant hurdles. Nevertheless, the potential for an ever-available, rapid, and consistently accurate diagnostic tool presents a compelling opportunity to revolutionize dermatological care.

Moreover, the segmentation analysis highlights the importance of factors such as BMI, smoking, and family history in connecting to necessary help and determining insurance premiums, further emphasizing the value of early detection and intervention provided by the app.

In conclusion, the scalability and potential for growth in the field of AI disruption in dermatology make it an opportune time to be an early adopter. By addressing the healthcare gap, fostering transparency, and harnessing the power of AI, the dermatological diagnosis app has the potential to significantly impact healthcare outcomes and improve accessibility for diverse demographics in India.

13. References

[1] Technology adoption life cycle (2023a) Wikipedia. Available at: https://en.wikipedia.org/wiki/Technology_adoption_life_cycle

[2] Electric two-wheelers in India. Available at: https://aeee.in/wp-content/uploads/2022/07/ICA-AEEE-Whitepaper-2022.pdf

[3] Electric car vector art, icons, and graphics for free download Vecteezy. Available at: https://www.vecteezy.com/free-vector/electric-car

[4] Dolnicar, S. (2018) Market segmentation analysis. Springer Nature.

[5] Sheth, J.N. Segmenting the health care market, Segmenting the Health Care Market. Available at: https://www.jagsheth.com/wp-content/uploads/2014/02/segmenting_the_health_care_market.pdf

[6] Skin treatment - india: Statista market forecast (no date) Statista. Available at: https://www.statista.com/outlook/hmo/otc-pharmaceuticals/skin-treatment/india#revenue