District Level Healthcare Planning
&
Pregnancy Outcome Prediction

PROJECT REPORT

Submitted by:

Shubham Arora (ID: 11841080)

Rhythm Gupta (ID:11840920)

Akash Likhar (ID: 11840110)

Guided by:

Dr. Gagan Raj Gupta

INDIAN INSTITUTE OF TECHNOLOGY BHILAI
DS250 - Data Analytics And Visualization
WINTER 2020

Motivation/Objective:

According to the Overall Health System Performance Report 2020, generated by the World Health Organisation, India ranks 112th amongst 191 countries. This is something which really needs to be improved because having proper healthcare facilities is one of the key factors which are necessary for the development of a country.

Moreover, the maternal healthcare facilities are also not in good condition. In tier 1 and tier 2 cities, we do have good and well equipped hospitals but they are costly when seen from the point of view of poor and middle class households. Government hospitals, on the other hand, are relatively cheap but they lack basic amenities.

In India, people are not aware about various common diseases, their symptoms, preventive measures etc. For example, people are not aware about AIDS and how it can be prevented. Moreover, most of the population, especially in rural areas, don't know about various contraceptive measures which are present in the market. In some cases, it has been observed that people are not aware about the schemes which have been released by the government for their welfare.

These are some major issues which if not taken into account and solved, can decrease the rate at which our nation will develop.

Therefore, with the help of data analysis, we have tried to rank the districts on the basis of healthcare facilities available. This will give us a clear picture about the regions which require more attention. Also, a model has been generated in order to predict the outcome of pregnancy (live birth or stillbirth).

Dataset:

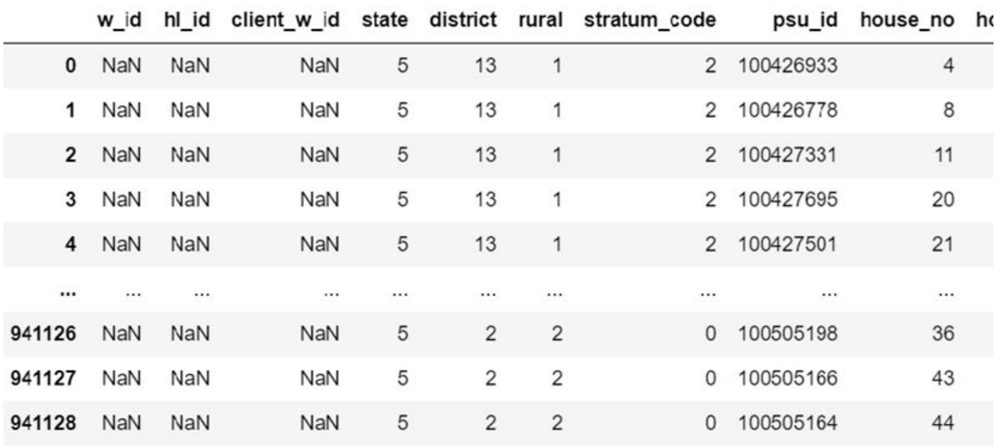
Source: Health Management Information System

The data set which was downloaded initially (for the state of Uttarakhand (5.csv)) contained 9,41,131 rows and 202 columns.

Dataset description file was also present at the same place, which contained the meaning of each column and mapping of each integer (used in the dataset) with its meaning.

Data Cleaning:

The dataset which was downloaded initially contained a lot of columns out of which many were irrelevant for our objective. Following image shows the raw dataset :



	w_id	hl_id	client_w_id	state	district	rural	stratum_code	psu_id	house_no	hc
0	NaN	NaN	NaN	5	13	1	2	100426933	4	
1	NaN	NaN	NaN	5	13	1	2	100426778	8	
2	NaN	NaN	NaN	5	13	1	2	100427331	11	
3	NaN	NaN	NaN	5	13	1	2	100427695	20	
4	NaN	NaN	NaN	5	13	1	2	100427501	21	
...	...	...	...	...	...	...	...	...	...	
941126	NaN	NaN	NaN	5	2	2	0	100505198	36	
941127	NaN	NaN	NaN	5	2	2	0	100505166	43	
941128	NaN	NaN	NaN	5	2	2	0	100505164	44	

We were working on 2 separate tasks, one was planning **District Level Healthcare Programs** and the other one was **Prediction of Outcome of Pregnancy**, so cleaning for both the tasks was done separately.

Data Cleaning for Dataset:

After thoroughly reading about the data in each column and understanding whether it was necessary to keep it or not for further procedures, the number of columns was reduced to 100 from 204. Some irrelevant columns which were removed were - w_id, hl_id, client_w_id, state, psu_id, house_no, date_of_birth, date_of_marriage etc. The NaN values in each column were also taken care of.

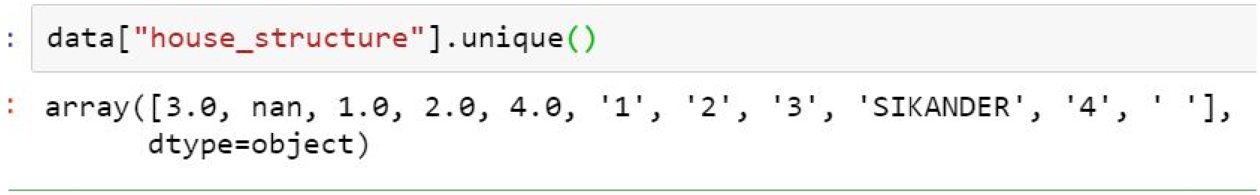
Steps involved in cleaning:

1. Number of NaN values for each column was calculated, and this information has been used in further steps.
2. Next, for each of the remaining 100 columns, we tried to find the cause of the NaN value, and replaced it with either -1 for *not applicable, not stated* or *others*.
3. Many columns were there which were interrelated, for example the columns 'marital_status' and 'is_currently_pregnant' had a relation. Such relations were taken into consideration while cleaning to replace it with something more appropriate. If we replaced NaN values in any column with *not applicable* then we mathematically checked if we were correctly replacing it or not.

   For example, in *is_using_any_fp_method* & *reason_for_not_using_fp_method*, if we are replacing NaN values in *reason_for_not_using_fp_method* with *not applicable* (*fp* means family planning), then:

   # NaN in reason_for_not_using_fp_method ~ # Yes in is_using_any_fp_method

4. Some columns even contained some irrelevant values (probably because of human error), like 'SIKANDER' in the image given below.



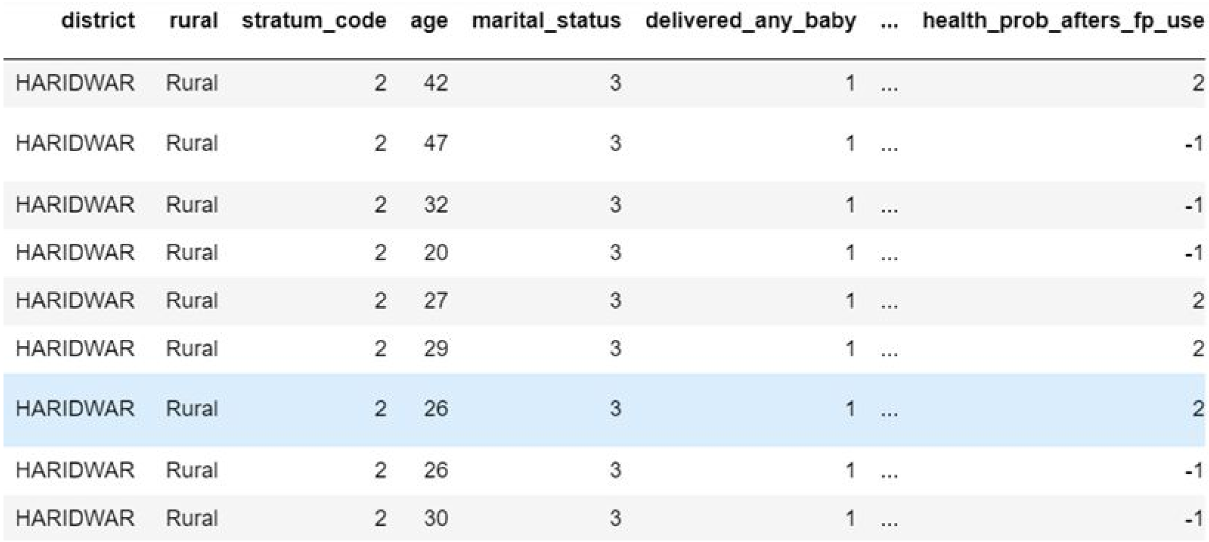
```
data["house_structure"].unique()
```

```
array([3.0, nan, 1.0, 2.0, 4.0, '1', '2', '3', 'SIKANDER', '4', ' '],
      dtype=object)
```

We found such values for each column, and replaced it with NaN.

5. Applying step 2, 3 and 4 on each column, we finally clean our dataset.
   (Size of dataset reduced from (403MB to ~230MB)

Analysing these relations amongst 100 columns was the most difficult part and consumed a lot of time. Image below shows the dataset after cleaning:



district	rural	stratum_code	age	marital_status	delivered_any_baby	...	health_prob_afters_fp_use
HARIDWAR	Rural	2	42	3	1	...	2
HARIDWAR	Rural	2	47	3	1	...	-1
HARIDWAR	Rural	2	32	3	1	...	-1
HARIDWAR	Rural	2	20	3	1	...	-1
HARIDWAR	Rural	2	27	3	1	...	2
HARIDWAR	Rural	2	29	3	1	...	2
HARIDWAR	Rural	2	26	3	1	...	2
HARIDWAR	Rural	2	26	3	1	...	-1
HARIDWAR	Rural	2	30	3	1	...	-1

Data Cleaning for Pregnancy Outcome Prediction (Live Birth or Stillbirth):

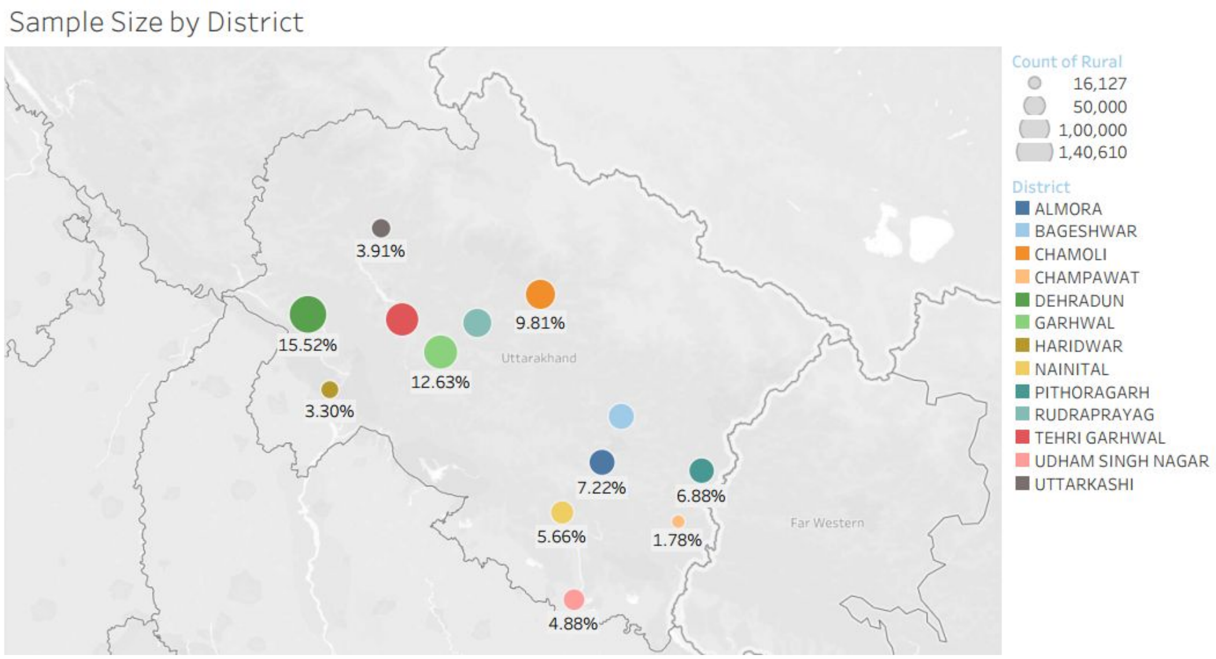
The columns which were to be removed for this part were different from the ones which were removed in the previous step. The number of columns was reduced to 82 after thorough analysis. Also, only those rows were kept in which the interview was successful. A function was made to do the cleaning part because cleaning was to be done on two datasets, smaller one for training the model and the larger one on which predictions were to be made.

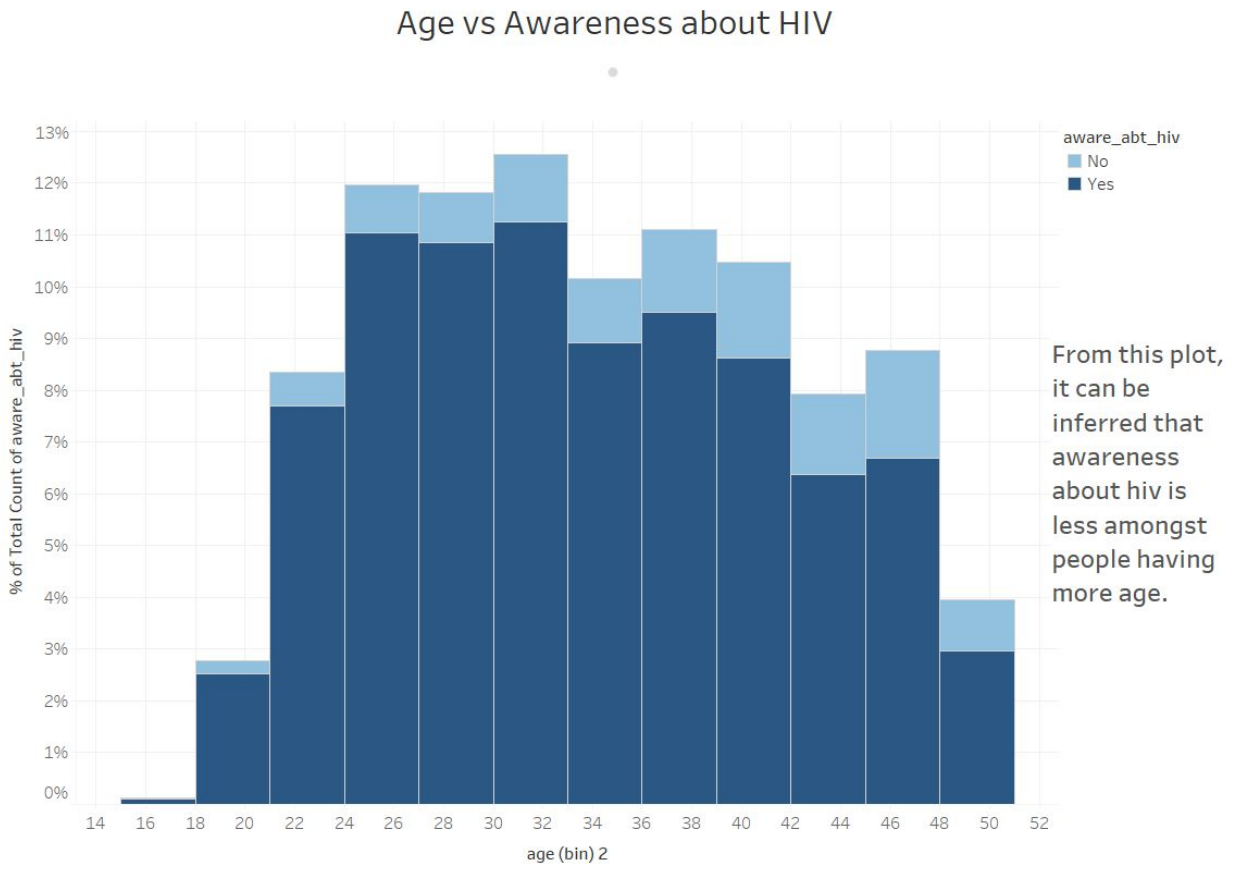
Visualization:

In order to generate interactive plots, Tableau was used for the purpose of visualization.

Moreover, for visualization, a separate dataset was created after cleaning the initial cleaned dataset.
NOTE: For accessing the interactive Tableau workbooks containing visualizations with their description, please download the dataset from this link.

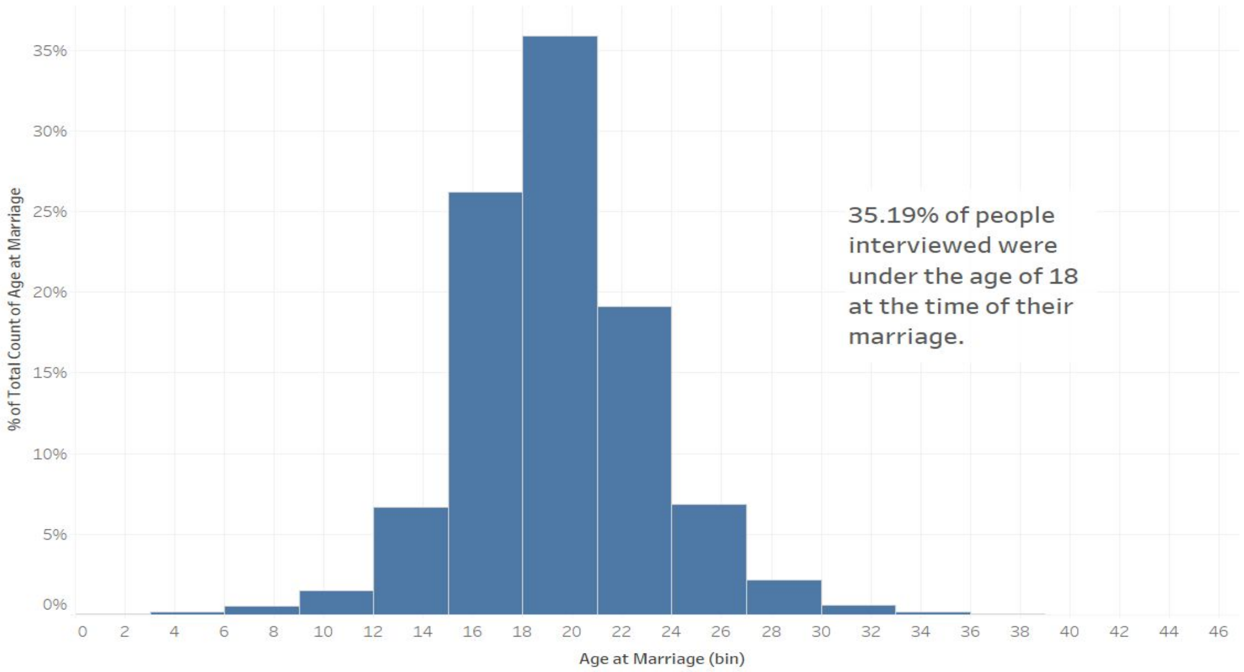
Some important visualizations:


Sample Size by District


Age vs Awareness about HIV

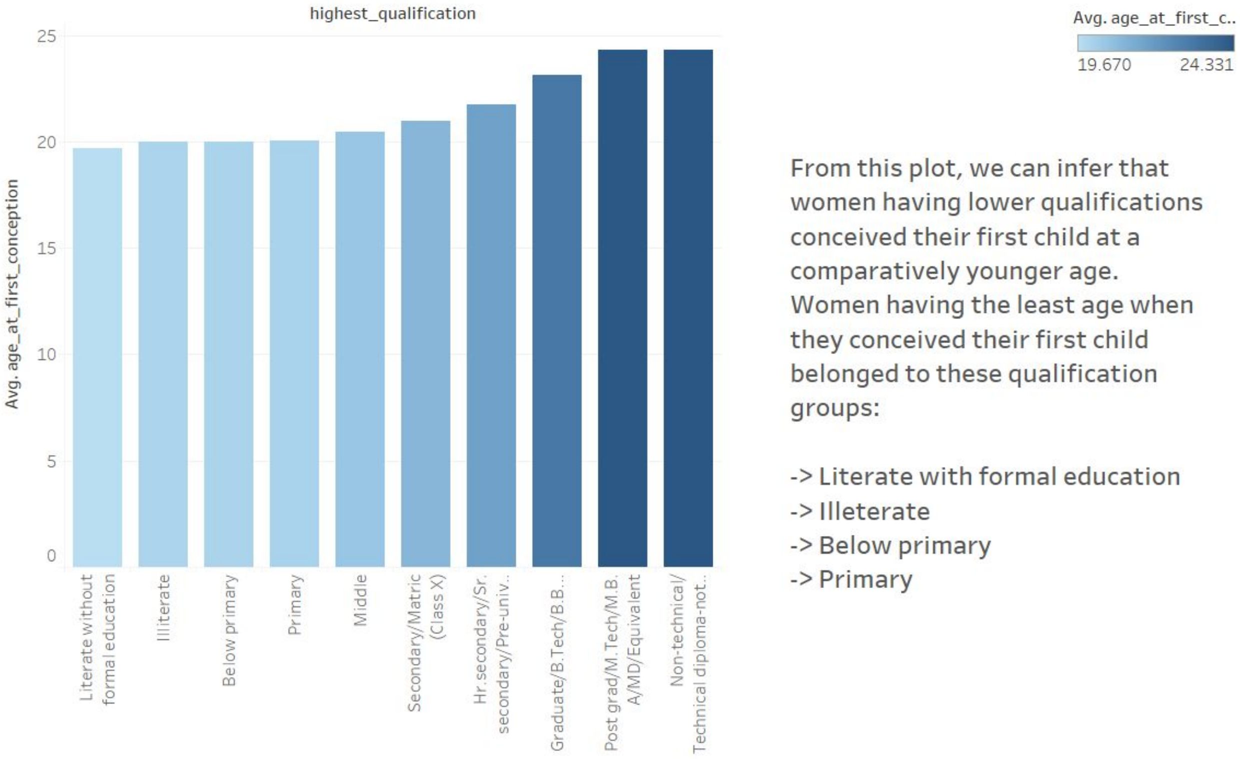
From this plot, it can be inferred that awareness about hiv is less amongst people having more age.

Age at Marriage



35.19% of people interviewed were under the age of 18 at the time of their marriage.

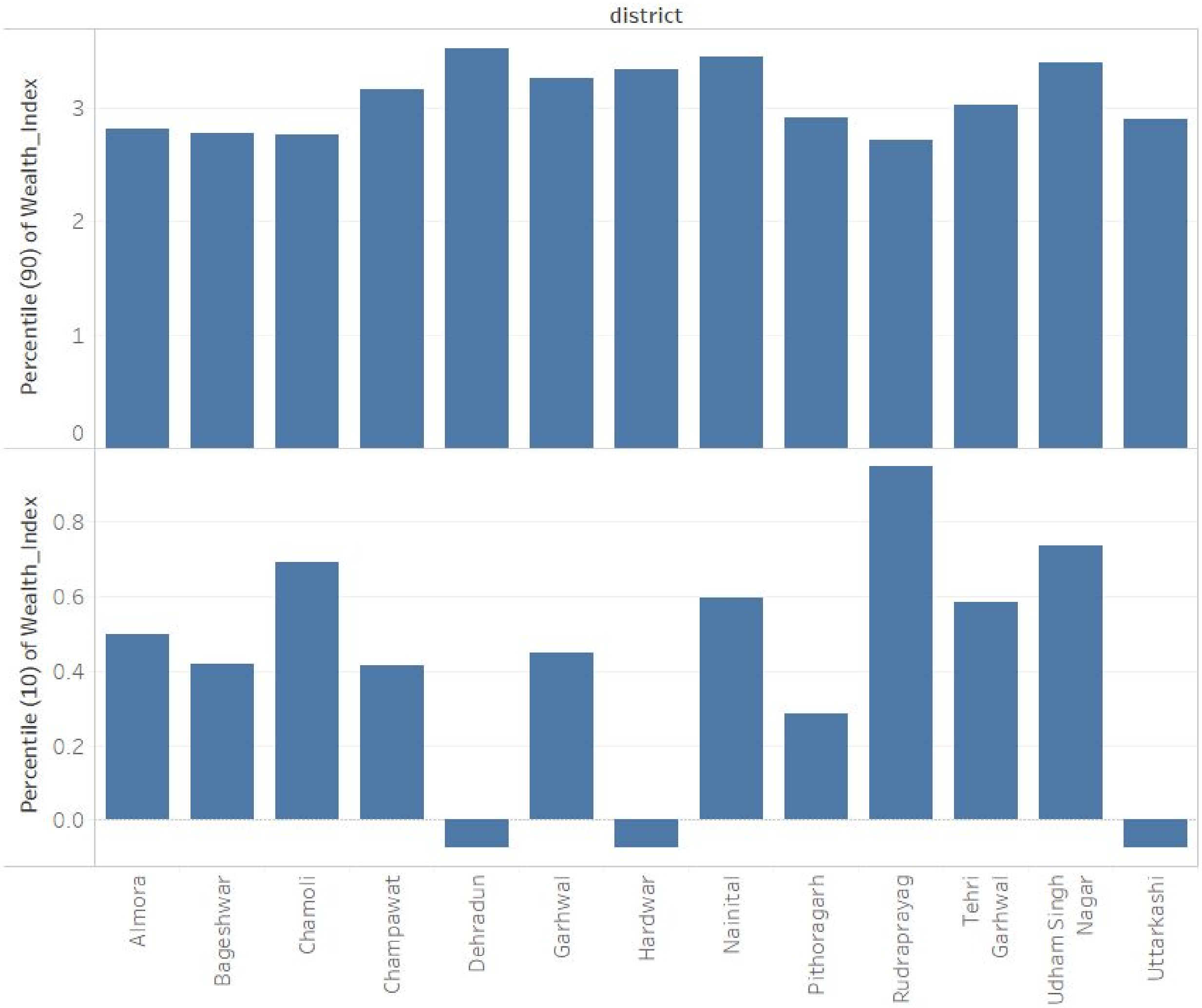
Highest Qualification vs Average Age at First Conception



From this plot, we can infer that women having lower qualifications conceived their first child at a comparatively younger age. Women having the least age when they conceived their first child belonged to these qualification groups:

-> Literate with formal education
-> Illeterate
-> Below primary
-> Primary

Districtwise Percentile Distribution of Wealth Index



Wealth Index Calculation:

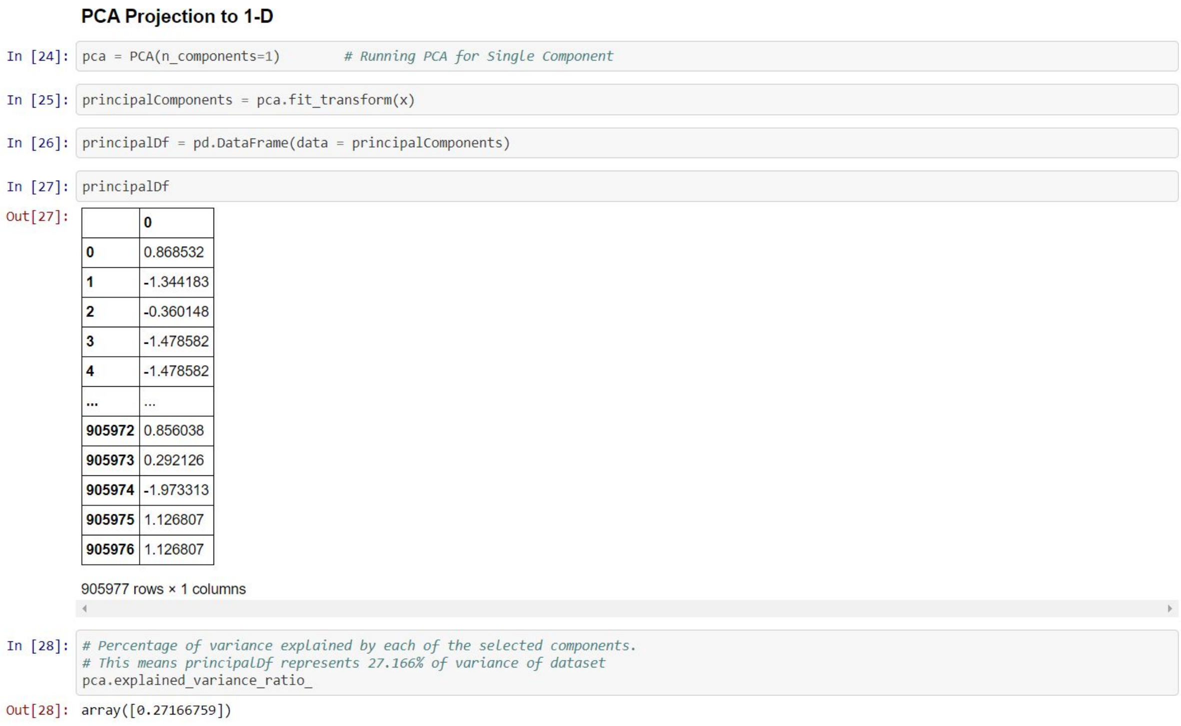
Wealth Index calculation depends on 21 variables selected from the dataset as per previous AHS results. These 21 columns are processed further and most of them are converted to bool type.

Objective was to calculate the wealth index for each household in our dataset for which assigning weightage to each of 21 columns was to be done. For this we used PCA with a single component to project the dataset to a single dimension.

Steps:

1. After performing PCA, “components_” was used to get maximum variance in the dataset. From this we got a multiplication factor for each of the 21 columns.
2. Then we multiplied each column with their respective multiplication factor to get the score for that column in our dataset.

3. Then we summed along each row in our dataset to obtain a ‘Wealth Index’ for each sample in our dataset.



PCA Projection to 1-D

```
In [24]: pca = PCA(n_components=1)        # Running PCA for Single Component

In [25]: principalComponents = pca.fit_transform(x)

In [26]: principalDf = pd.DataFrame(data = principalComponents)

In [27]: principalDf
Out[27]:
```

	0
0	0.868532
1	-1.344183
2	-0.360148
3	-1.478582
4	-1.478582
...	...
905972	0.856038
905973	0.292126
905974	-1.973313
905975	1.126807
905976	1.126807

905977 rows × 1 columns

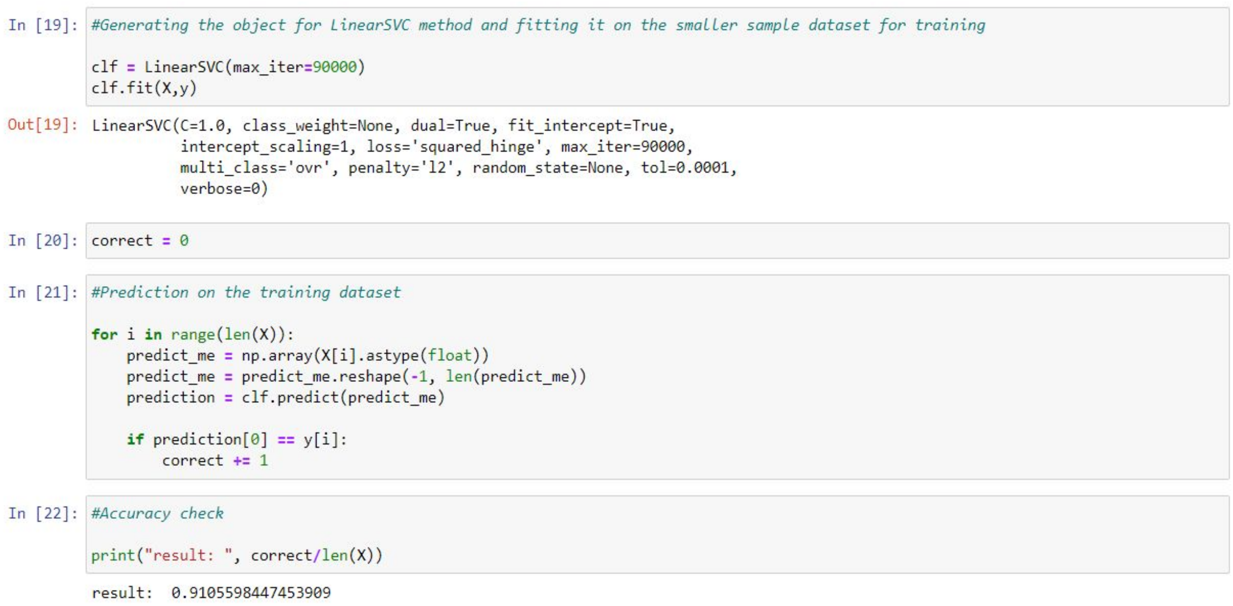
```
In [28]: # Percentage of variance explained by each of the selected components.
         # This means principalDf represents 27.166% of variance of dataset
         pca.explained_variance_ratio_
Out[28]: array([0.27166759])
```

<u>Pregnancy Outcome Prediction</u>:

The objective was to develop a model which would be able to predict the outcome of pregnancy - live birth or stillbirth, after taking various features into its consideration.

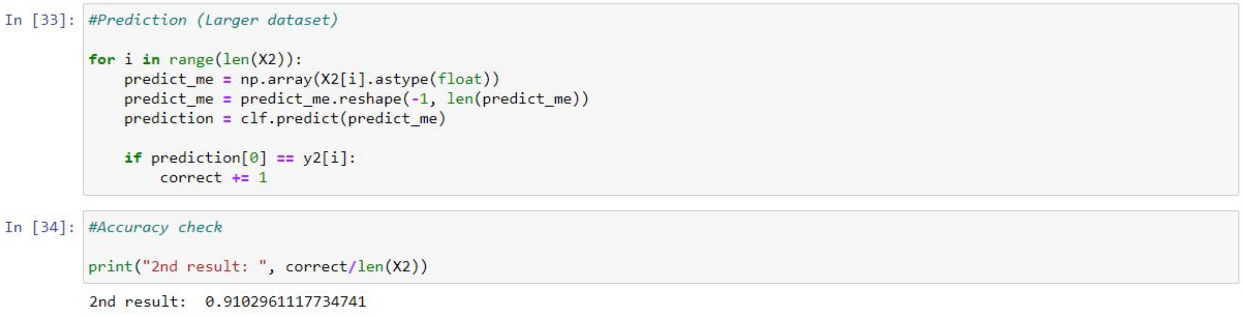
<u>Steps</u>:

1. First, the dataset was cleaned using a function and the number of columns was reduced to 82. Also the data was converted into numeric format from the earlier string format.
2. The initial dataset was very large (9,41,131 rows and 202 columns) and training/fitting a model on such a large dataset would have taken very much time. Hence a fraction of the initial dataset (containing 25000 rows) was selected for training/fitting the model.
3. Then the cleaning function was run on the fractional dataset and it was divided into two parts, ‘X’ (containing the feature variables) and ‘y’ (containing the prediction variable - ‘outcome_pregnancy’).
4. The numpy array ‘X’ was then standardized using ‘preprocessing.scale’.
5. The first model that was taken into consideration was ‘LinearSVC’ (**S**upport **V**ector **C**lassifier). The LinearSVC model was first trained on the fractional dataset and the accuracy was calculated.



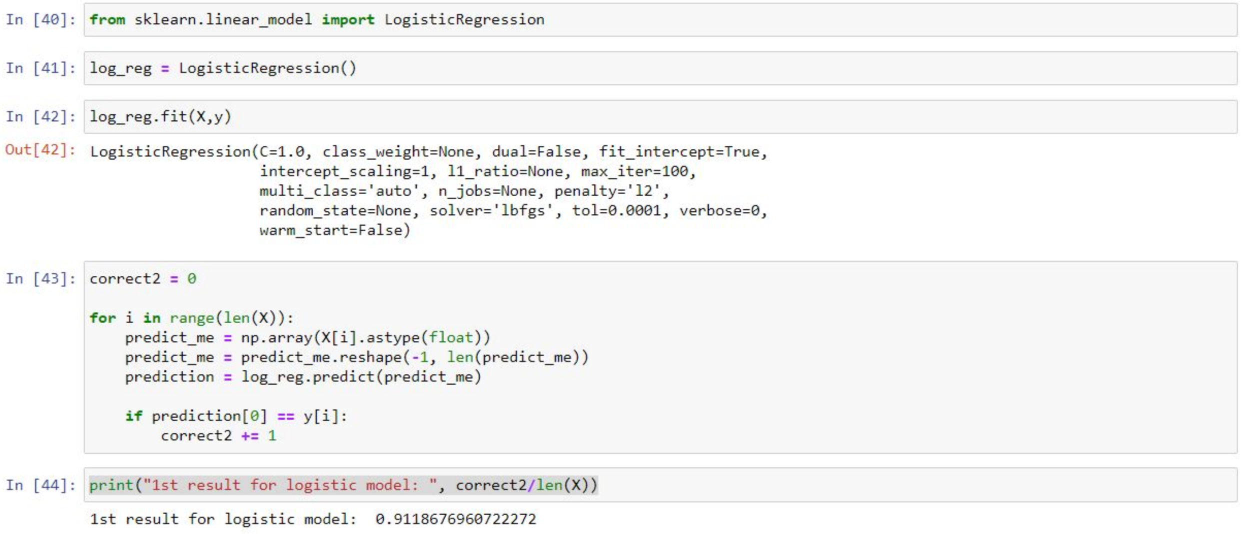
```
In [19]: #Generating the object for LinearSVC method and fitting it on the smaller sample dataset for training

         clf = LinearSVC(max_iter=90000)
         clf.fit(X,y)

Out[19]: LinearSVC(C=1.0, class_weight=None, dual=True, fit_intercept=True,
                   intercept_scaling=1, loss='squared_hinge', max_iter=90000,
                   multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
                   verbose=0)

In [20]: correct = 0

In [21]: #Prediction on the training dataset

         for i in range(len(X)):
             predict_me = np.array(X[i].astype(float))
             predict_me = predict_me.reshape(-1, len(predict_me))
             prediction = clf.predict(predict_me)

             if prediction[0] == y[i]:
                 correct += 1

In [22]: #Accuracy check

         print("result: ", correct/len(X))

         result:  0.9105598447453909
```

6. Now the LinearSVC model was used for prediction on the initial dataset and the accuracy that was achieved was: ‘**0.9103**’.



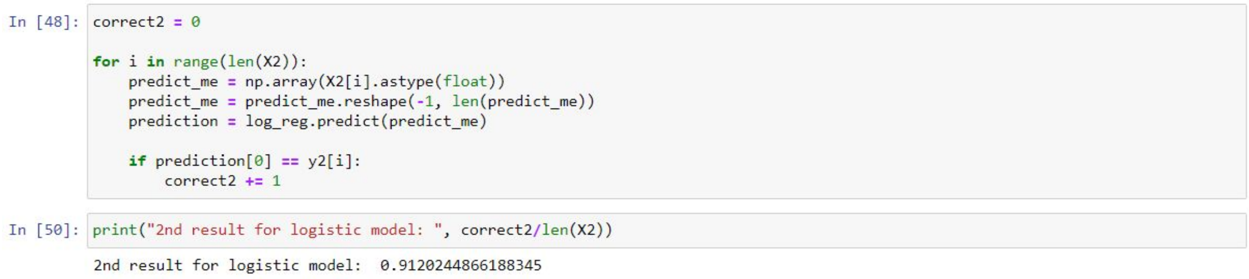
```
In [33]: #Prediction (Larger dataset)

         for i in range(len(X2)):
             predict_me = np.array(X2[i].astype(float))
             predict_me = predict_me.reshape(-1, len(predict_me))
             prediction = clf.predict(predict_me)

             if prediction[0] == y2[i]:
                 correct += 1

In [34]: #Accuracy check

         print("2nd result: ", correct/len(X2))

         2nd result:  0.9102961117734741
```

7. The second model which was taken into consideration was the ‘Logistic Regression’ model. This model was also first trained on the fractional dataset and the accuracy was calculated.



```
In [40]: from sklearn.linear_model import LogisticRegression

In [41]: log_reg = LogisticRegression()

In [42]: log_reg.fit(X,y)

Out[42]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, l1_ratio=None, max_iter=100,
                            multi_class='auto', n_jobs=None, penalty='l2',
                            random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                            warm_start=False)

In [43]: correct2 = 0

         for i in range(len(X)):
             predict_me = np.array(X[i].astype(float))
             predict_me = predict_me.reshape(-1, len(predict_me))
             prediction = log_reg.predict(predict_me)

             if prediction[0] == y[i]:
                 correct2 += 1

In [44]: print("1st result for logistic model: ", correct2/len(X))

         1st result for logistic model:  0.9118676960722272
```

8. Now the Logistic Regression model was used for prediction on the initial dataset and the accuracy that was achieved was: ‘**0.912**’.



```
In [48]: correct2 = 0

         for i in range(len(X2)):
             predict_me = np.array(X2[i].astype(float))
             predict_me = predict_me.reshape(-1, len(predict_me))
             prediction = log_reg.predict(predict_me)

             if prediction[0] == y2[i]:
                 correct2 += 1

In [50]: print("2nd result for logistic model: ", correct2/len(X2))

         2nd result for logistic model:  0.9120244866188345
```

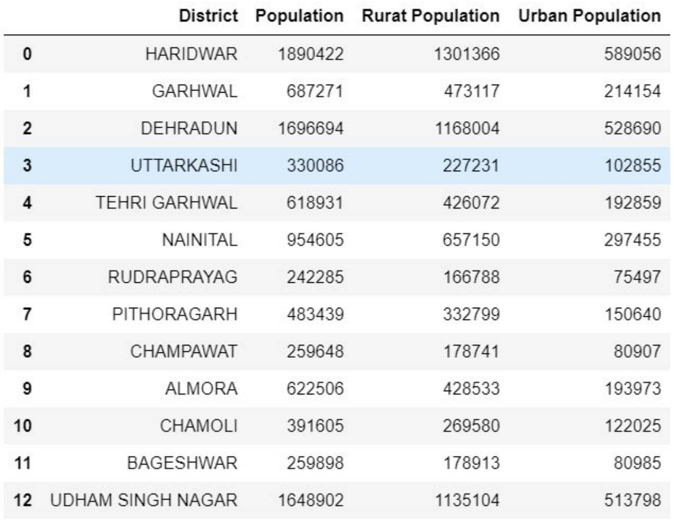
Scoring:

3 scores were calculated for each district.

1. Score of rural area
2. Score of urban area
3. Score of whole district

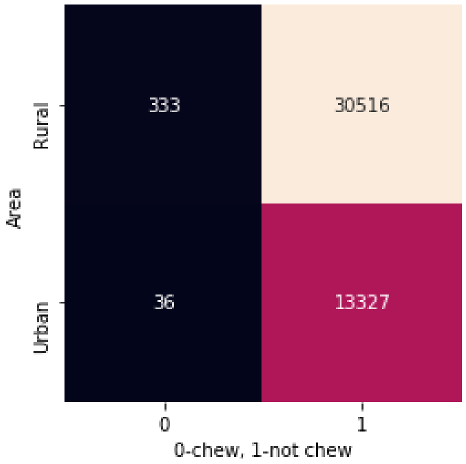
Steps in scoring:

1. Each of the 33 columns were modified such that they only contained values - either 0 (means negative response) or 1 (positive response) (or -1 if not applicable).
2. Then for each district we found the population from the 2011 census data using **this**.
3. To calculate the scores separately for rural and urban populations, we used approximation. According to **this** report, 68.84% of the population according to the 2011 census is from rural areas. So, to divide the population of districts of Uttarakhand between rural and urban, we used the ratio 68.84:31.16.

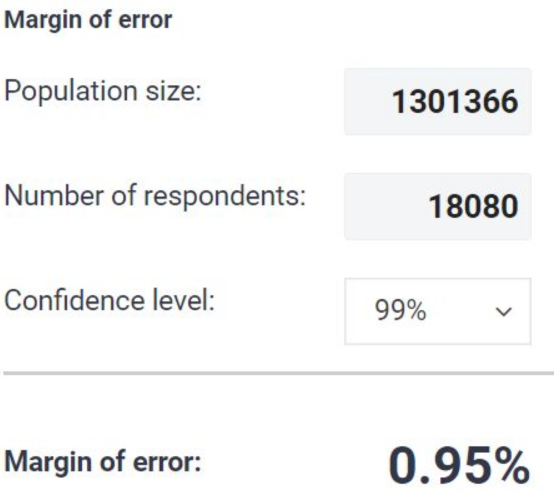


	District	Population	Rurat Population	Urban Population
0	HARIDWAR	1890422	1301366	589056
1	GARHWAL	687271	473117	214154
2	DEHRADUN	1696694	1168004	528690
3	UTTARKASHI	330086	227231	102855
4	TEHRI GARHWAL	618931	426072	192859
5	NAINITAL	954605	657150	297455
6	RUDRAPRAYAG	242285	166788	75497
7	PITHORAGARH	483439	332799	150640
8	CHAMPAWAT	259648	178741	80907
9	ALMORA	622506	428533	193973
10	CHAMOLI	391605	269580	122025
11	BAGESHWAR	259898	178913	80985
12	UDHAM SINGH NAGAR	1648902	1135104	513798

4. After calculating this, for each district and each of 33 columns we chose, we calculated the confusion matrix, and found the number of people who give a negative response. The score of that area is the fraction of people who are unaware of that particular problem. Following image shows the confusion matrix of *chew* vs *area*.



5. Next, we calculated the aggregate score of the region by taking the mean (mean is taken because every column corresponds to a problem which is as important as any other one, that's why we gave weight to each column as 1, rather than calculating weights for separate columns using PCA or any other technique).
6. This score is only for a subset of the population, the original score (score for all population), may vary. To take this factor into account, we calculated the margin of error for each district's rural and urban area separately. To calculate this we use **this**, which will take the total population, number of people responding to the survey and confidence interval. (All our calculations are at 99% confidence). Following image shows the margin of error calculation for Haridwar district rural area.



Margin of error

Population size: 1301366

Number of respondents: 18080

Confidence level: 99%

Margin of error: 0.95%

7. If x is the score we calculated for any district and y is the margin of error, then score range for that district is,

$$score = x \pm y$$

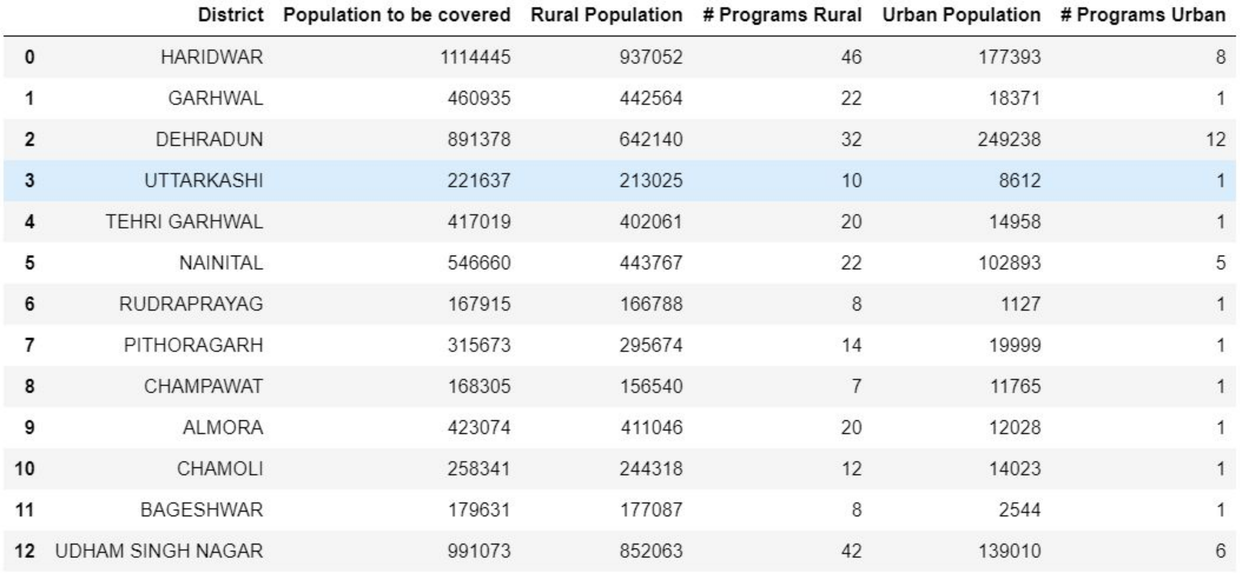
We only take the positive margin into consideration to be on the safer side.

$$score = min(1,\ x + y)$$

8. Next step is to calculate the population count with lack of knowledge and awareness about health problems.
9. To calculate the number of healthcare programs required, first we found in government programs that how many people are covered in one such program, which is around 20,000. So, taking

$$number\ of\ people\ per\ program = 20000$$

we calculated the number of programs required by just dividing the total population to be covered under this program with 20,000. Following image shows the population to be covered under such programs, how many from rural areas, how many from urban areas, and the number of programs required in rural and urban areas.

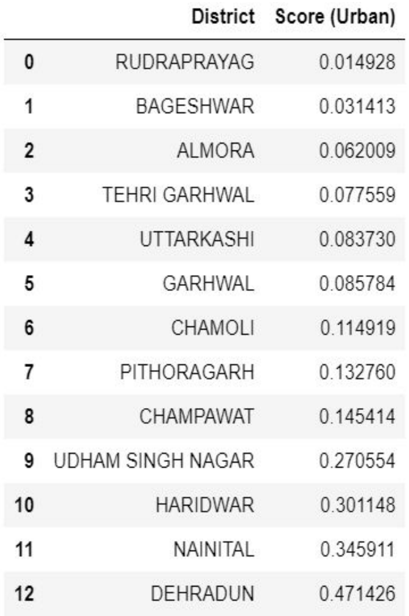


	District	Population to be covered	Rural Population	# Programs Rural	Urban Population	# Programs Urban
0	HARIDWAR	1114445	937052	46	177393	8
1	GARHWAL	460935	442564	22	18371	1
2	DEHRADUN	891378	642140	32	249238	12
3	UTTARKASHI	221637	213025	10	8612	1
4	TEHRI GARHWAL	417019	402061	20	14958	1
5	NAINITAL	546660	443767	22	102893	5
6	RUDRAPRAYAG	167915	166788	8	1127	1
7	PITHORAGARH	315673	295674	14	19999	1
8	CHAMPAWAT	168305	156540	7	11765	1
9	ALMORA	423074	411046	20	12028	1
10	CHAMOLI	258341	244318	12	14023	1
11	BAGESHWAR	179631	177087	8	2544	1
12	UDHAM SINGH NAGAR	991073	852063	42	139010	6

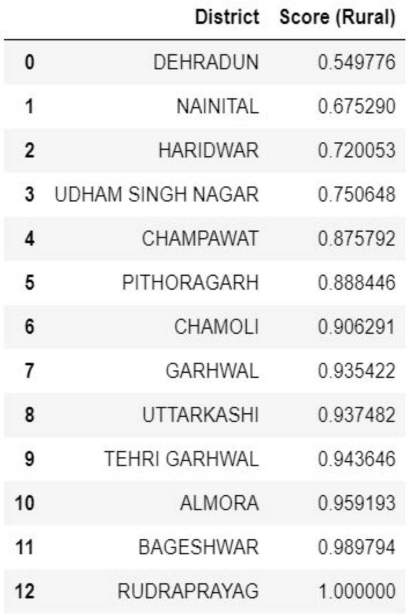
Ranking:

Just like scoring, 3 rankings were calculated for each district:

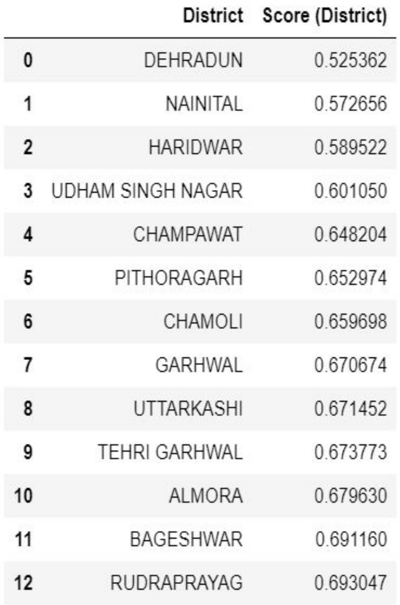
1. Rank of rural area
2. Rank of urban area
3. Rank of whole district



	District	Score (Urban)
0	RUDRAPRAYAG	0.014928
1	BAGESHWAR	0.031413
2	ALMORA	0.062009
3	TEHRI GARHWAL	0.077559
4	UTTARKASHI	0.083730
5	GARHWAL	0.085784
6	CHAMOLI	0.114919
7	PITHORAGARH	0.132760
8	CHAMPAWAT	0.145414
9	UDHAM SINGH NAGAR	0.270554
10	HARIDWAR	0.301148
11	NAINITAL	0.345911
12	DEHRADUN	0.471426



	District	Score (Rural)
0	DEHRADUN	0.549776
1	NAINITAL	0.675290
2	HARIDWAR	0.720053
3	UDHAM SINGH NAGAR	0.750648
4	CHAMPAWAT	0.875792
5	PITHORAGARH	0.888446
6	CHAMOLI	0.906291
7	GARHWAL	0.935422
8	UTTARKASHI	0.937482
9	TEHRI GARHWAL	0.943646
10	ALMORA	0.959193
11	BAGESHWAR	0.989794
12	RUDRAPRAYAG	1.000000



	District	Score (District)
0	DEHRADUN	0.525362
1	NAINITAL	0.572656
2	HARIDWAR	0.589522
3	UDHAM SINGH NAGAR	0.601050
4	CHAMPAWAT	0.648204
5	PITHORAGARH	0.652974
6	CHAMOLI	0.659698
7	GARHWAL	0.670674
8	UTTARKASHI	0.671452
9	TEHRI GARHWAL	0.673773
10	ALMORA	0.679630
11	BAGESHWAR	0.691160
12	RUDRAPRAYAG	0.693047

Inferences:

1. In every district, in rural areas more than 50% of the population is unaware of health related problems, with 9 districts out of 13 having more than 80%, and 1 even having almost all its population as unaware.
2. Condition is better in urban areas as compared to rural areas, with all the districts having less than 50% of population as not aware of health related problems. Dehradun, unexpectedly, is the area with the most number of unaware people despite being the best in rural areas.
3. Similar to Dehradun, Rudraprayag is worst in rural areas, but best in urban areas.
4. Since almost 70% of the total population is from rural areas, rural areas should have more weight in calculating the score of districts. From the table above we can see that this is the case, i.e. ranking of districts is the same as their rural ranking.
5. Rural areas require more government attention as compared to urban areas.
6. **Refer to Tableau worksheets for inferences/relations of other columns.**

Problems Faced:

1. Doing dimensionality reduction is one of the main problems we faced during the project. All columns in our dataset are categorical, hence using PCA is definitely not a good idea.
   Finally, we apply PCA by making our variables in such a way that it performs like continuous values.
2. Data Cleaning was also a major problem because data contains a lot of NaN values, irrelevant columns. But after thorough analysis of each column of the dataset, we were able to clean it.

What we Learnt:

This project turned out to be really interesting for me. Not only did it continuously keep my interest but I learnt a lot too. Before this project, I knew nothing about tableau, but now I know quite a bit about it and after using it for generating some wonderful plots like - geoplots, treemaps, packed bubbles etc., I am even more fascinated by it. I also gained some knowledge about LinearSVC and various classification models.

~Rhythm Gupta

I learnt a lot of things from this project, especially Tableau which helps in making interactive and beautiful visualizations in very little effort.
Another thing that this project explained to me is how difficult it is to analyse the survey results and why 60% of data scientists are cleaning and organising data.

~Shubham Arora

This project provided me with hands-on experience of working on a real world dataset. I was able to learn new softwares like Tableau and also understood the importance of the Data Cleaning part of analysing surveys. I also gained some knowledge on PCA and understood various aspects regarding Health Surveys and how to extract and process more information from them.

~Akash Likhar

References:

1. https://nrhm-mis.nic.in/hmisreports/AHSReports.aspx
2. https://www.checkmarket.com/sample-size-calculator/#sample-size-margin-of-error-calculator
3. https://www.censusindia.co.in/districts
4. https://indiafacts.in/india-census-2011/urban-rural-population-o-india/
5. https://www.kaggle.com/rajanand/ahs-woman-1
6. https://www.censusindia.gov.in/vital_statistics/AHSBulletins/AHS_Baseline_Factsheets/Rajasthan.pdf
7. https://dhsprogram.com/topics/wealth-index/Wealth-Index-Construction.cfm

Contribution of team members:

Team Coordinator: Shubham Arora

Name	Data Cleaning	Data Visualization	Wealth Index Calculation	Prediction Model	Scoring and Ranking	Report
Shubham Arora	**Yes**	**Yes**	**No**	**No**	**Yes**	**Yes**
Rhythm Gupta	**Yes**	**Yes**	**No**	**Yes**	**No**	**Yes**
Akash Likhar	**Yes**	**Yes**	**Yes**	**No**	**No**	**Yes**