Step 0: Explain the problem you are going to solve using the selected data.

Using 23 different SDOH, this application is predicting whether a person with given social determinants is prone to any chronic condition or not. It can help healthcare payers to understand and anticipate health conditions of their incoming customers based on their unmet social needs.

SDOH and Insrance Payers

Social Determinant of health are conditions in the environments in which people are born, live, learn, work, play, worship and age that affect a wide range of health, functionality and quality of life outcomes and risks.

Insurance payers understand that, there are important factors beyond health care, although only 10% of the National Health Expenditure is spent on tackling these unmet social needs. With a desire to transform the health Industry from one of sick care to wellcare, US healthcare payers aims to address these unmet social needs affecting people with chronic condition(s) on Medicaid and Medicaid-Medicare Insurance plans.

Step 1.1: Find a public dataset in a domain you like. This dataset must be have the following features.

1- there is a possibility of finding versions of the data.
2- possibility of change in the data.
3- possibility of receiving future updated on the data.
4- have at least two protected features.

Source of data: Sources for Data on SDOH | Social Determinants of Health | CDC
1 – Yes
2 – Yes
3 – Yes
4 – Gender and Ethnicity

Step 1.2: Define some ML metrics to evaluate your model.

- Accuracy - It is the ratio of number of correct predictions to the total number of input samples.
  Accuracy = $\frac{\text{Total Correct Predictions}}{\text{Total Sample}}$

- Confusion Matrix - A matrix as output and describes the complete performance of the model.
  For example:
  Lets assume we have a binary classification problem. We have some samples belonging to two classes : YES or NO. Also, we have our own classifier which predicts a class for a given input sample. On testing our model on 165 samples ,we get the following result.

n=165	Prediction = No	Prediction = Yes
Actual = No	50	10
Actual = Yes	5	100

  • True Positives: The cases in which we predicted YES and the actual output was also YES.
  • True Negatives: The cases in which we predicted NO and the actual output was NO.
  • False Positives: The cases in which we predicted YES and the actual output was NO.
  • False Negatives: The cases in which we predicted NO and the actual output was YES.

  Accuracy for the matrix can be calculated by taking average of the values lying across the “main diagonal” i.e

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalSample}$$

- Precision

It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

- <u>Recall</u>

  It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

$$Precision = \frac{TruePositives}{TruePositives + FalseNegatives}$$

- <u>F1-Score</u>

  F1 Score is the Harmonic Mean between precision and recall.
  It tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Step 1.3: Define some business metrics to evaluate your model.

- <u>False Negatives</u> = 0

  Since we are dealing with chronic disease data, there should not be any False Negative. As explained above, false negatives are those outputs(not diagnosed with chronic disease) but actually they were.
  This will help in diagnosing each and every patient who is prone to any kind of chronic disease and thus, preventive measure could be taken in time. This will not only impact positively on health of citizens but its associated financial cost too.

  Though this is next to impossible in practical terms, but achievable using ML techniques.

- <u>False Positives</u> < 5%

  This means that only False positives, and that too <5%, could be entertained which can fall under the umbrella of our goals.
  It is so because any datapoint which is identified positively will be entertained by a human executive. It has an associated cost will be bore by the organization.

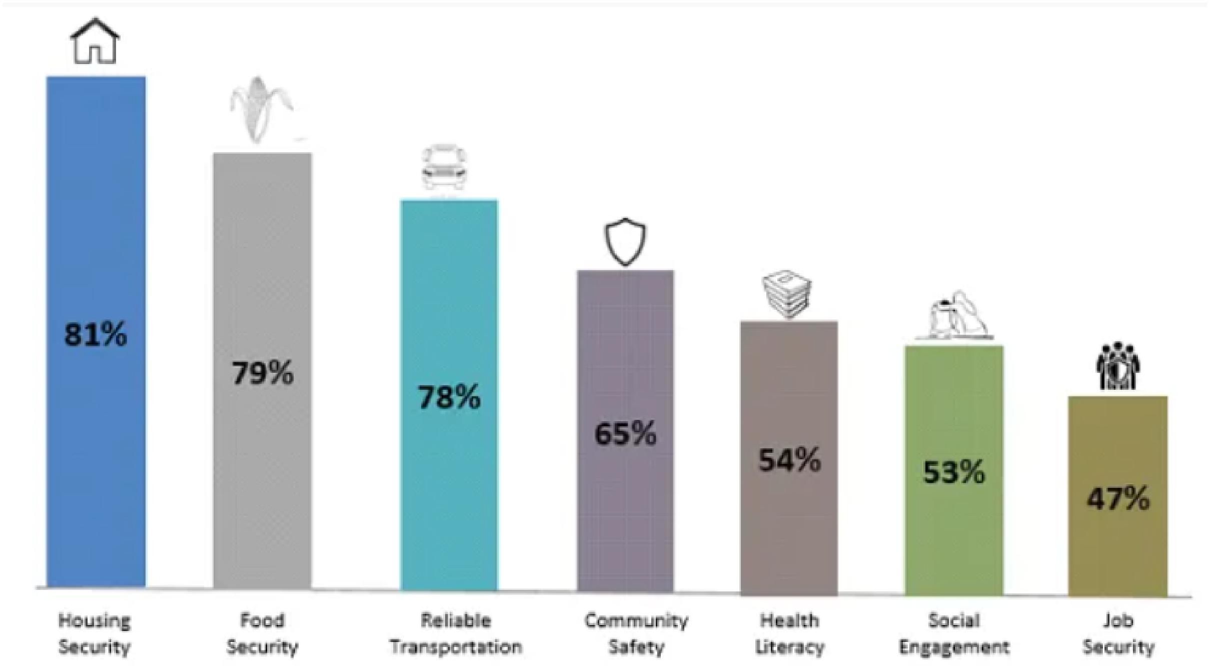
Step 1.4: Define some software metrics to evaluate your model.

- Extremely low latency – the model should provide results in almost real time.
- Available computation space for upcoming versions
- There shouldn’t be any cyclomatic complexity in the code and every module should follow SRP(Single Responsibility Principle)
- Miminal code smells without any architecture smells.

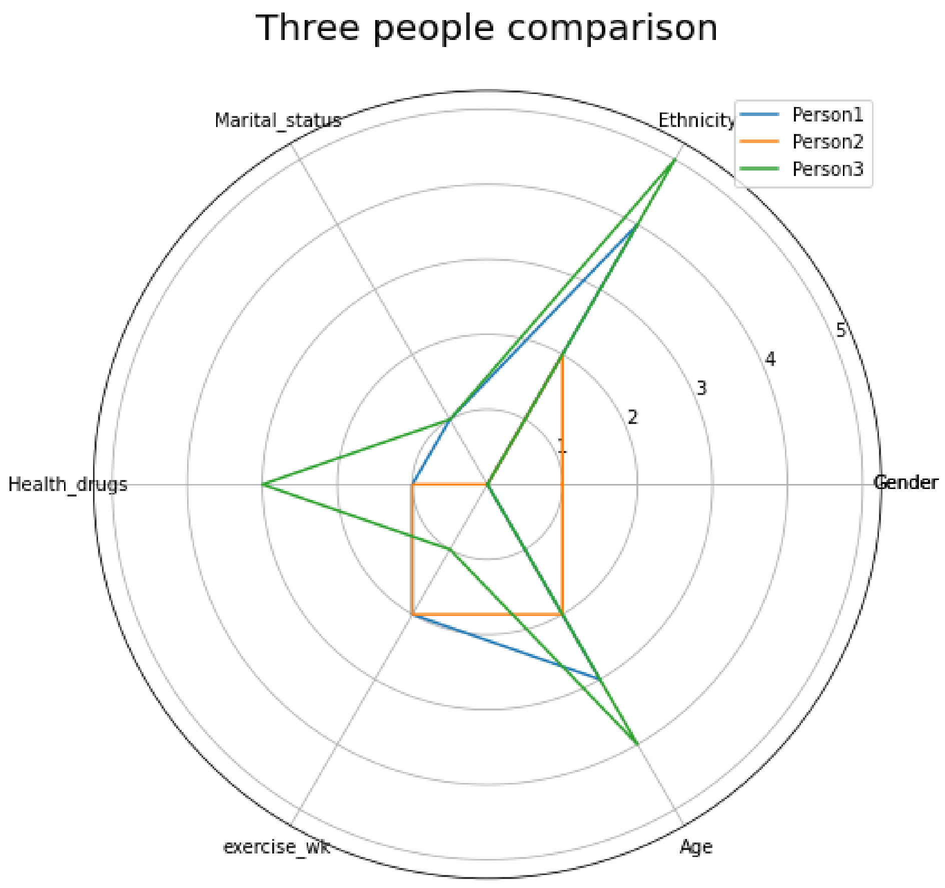
Step 2.1: Describe the dataset objective and the features made you decide on selecting it.

Objective: The dataset objective was to predict the probability of getting a chronic disease based on certain factors of Social Determinants of Health.

Majority of the respondents with chronic condition(s) consistently ranked Housing security or Food security and Reliable transportation as their most pressing social needs.
In addition to this, Community safety, Health Literacy, Social Engagement and Job security are important factors to be considered.



Step 2.1: Describe the quality of dataset using a radar chart with enough explanation.



The goal of the radar chart is to visually represent one or more groups of values over multiple variables. In our case we have 20+ variables containing data about different people. Visualizing comparison isn't the right fit in our case. For example in the above chart we can see the polygons created aren't giving any insight . Radar charts can get confusing fast — comparing more than a handful of observations leads to a mess no one wants to look at.

Step 2.2: upload your data in a public repo on https://git-lfs.github.com/

You can clone the data using:
git clone https://github.com/Chanpreet-Singh/csci-5901.git
or
simply by clicking here.

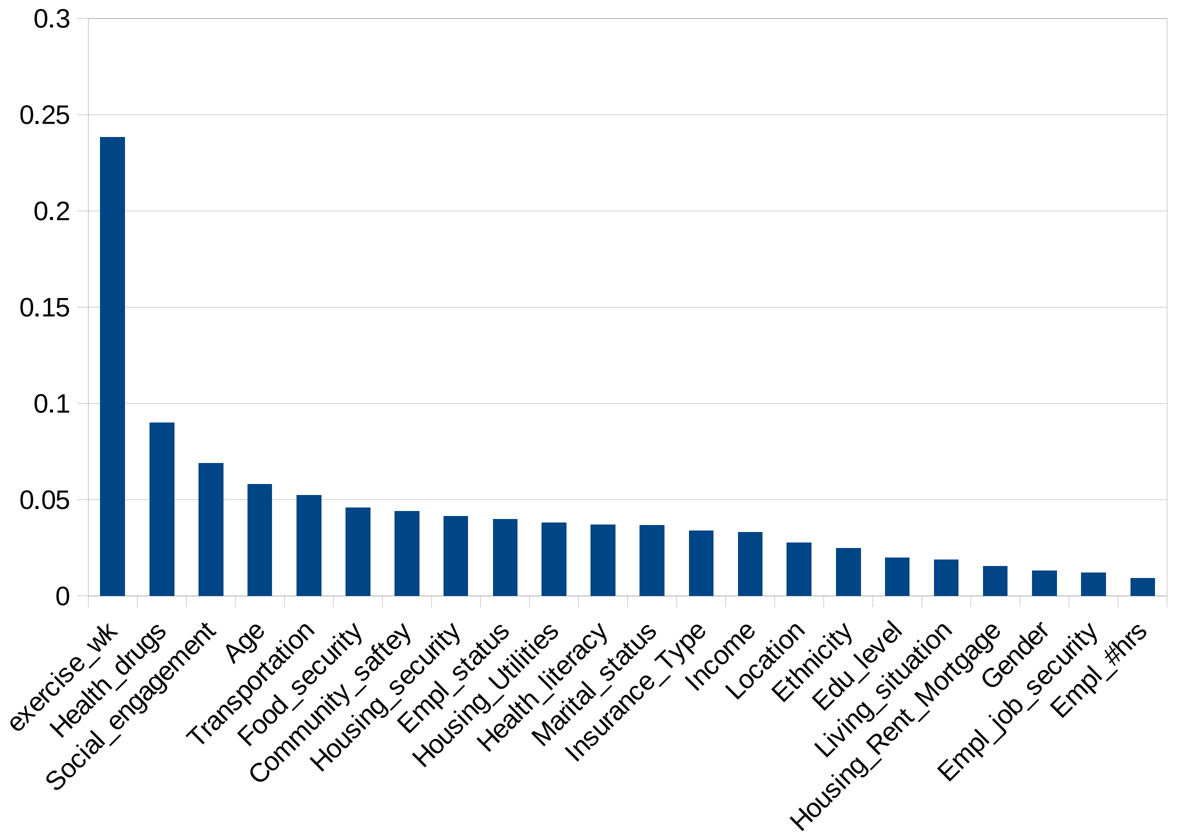
Step 3: What are the features in your dataset? what is the target variable?

Features

- exercise_wk
- Health_drugs
- Social_engagement
- Age
- Transportation
- Food_security
- Community_saftey
- Housing_security
- Empl_status
- Housing_Utilities
- Health_literacy
- Marital_status
- Insurance_Type
- Income
- Location
- Ethnicity
- Edu_level
- Living_situation
- Housing_Rent_Mortgage
- Gender
- Empl_job_security
- Empl_#hrs

Target Variable: Diagnosed

Step 4: Among the features what are the features with more predictive value?



This feature importance column graph shows that ‘excercise_wk’, ‘Health_drugs’, ‘Social_engagement’, ‘Age’ are the top 4 important features in predicting target variable – diagnosed. The column graph was prepared with the help of feature importance table using Random Forest Classification Technique.

Table1: Feature importance table

exercise_wk	0.238445
Health_drugs	0.089996
Social_engagement	0.068936
Age	0.058217
Transportation	0.052227
Food_security	0.045908
Community_saftey	0.044062

Housing_security	0.041522
Empl_status	0.039843
Housing_Utilities	0.038143
Health_literacy	0.037139
Marital_status	0.036904
Insurance_Type	0.033962
Income	0.03309
Location	0.027659
Ethnicity	0.02494
Edu_level	0.019901
Living_situation	0.018941
Housing_Rent_Mortgage	0.015504
Gender	0.013254
Empl_job_security	0.012026
Empl_#hrs	0.009382

Step 5: Identify all protected features? (for example in some domains we can say Gender is a predictive feature.)

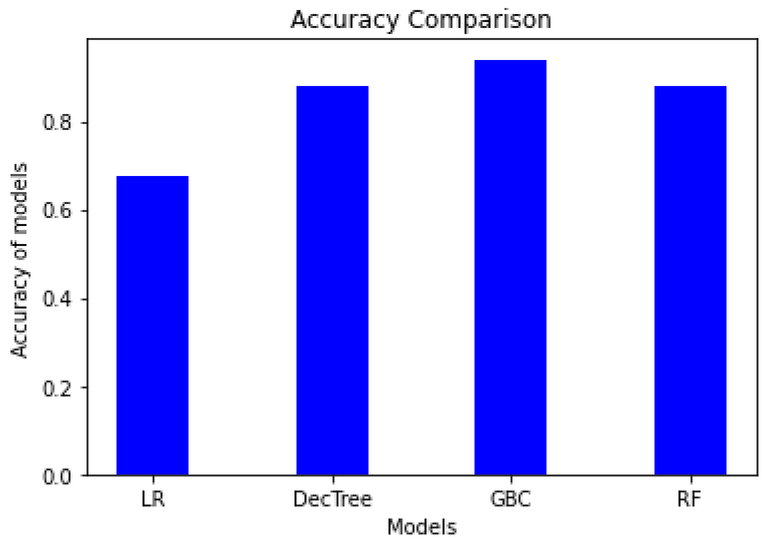
In our case, the protected features are:

- Gender
- Ethnicity

Step 6: Build a model to predict your target value.

For details, please refer to the code-notebook attached.

Step 7: Explain what model you utilized and the reason of choosing it.



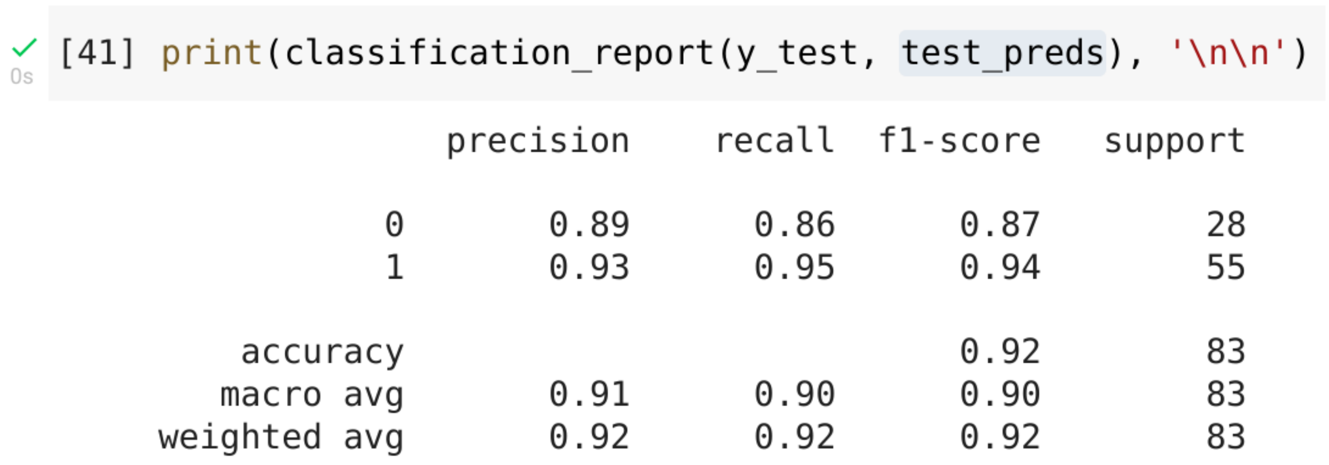
Gradient Boosting Classifier is giving us best accuracy among all hence we will be going ahead with this model only for final results.

It is also satisfying our the most prominent business metric that False Negative should be tending to 0. The confusion matrix of the test data in GBC is:

n=83	Prediction = No	Prediction = Yes
Actual = No	24	4
Actual = Yes	3	52

FN = 3/83 = 0.036%

Step 8: Using the identified ML metrics evaluate your model.



```
[41] print(classification_report(y_test, test_preds), '\n\n')
```

```
              precision    recall  f1-score   support

           0       0.89      0.86      0.87        28
           1       0.93      0.95      0.94        55

    accuracy                           0.92        83
   macro avg       0.91      0.90      0.90        83
weighted avg       0.92      0.92      0.92        83
```

For details, please refer to the code-notebook attached.

Step 9: Perform error analysis on your dataset and try to improve the performance of your solution by investigating the samples in your dataset.

Tag / hypothesis	ML Metric	Baseline	Gap to baseline	% of data
Chronic disease are highly dependent on SDOH	F1-score	0.95, 0.95	0.78, 0.83	70%
Chronic disease are highly dependent on SDOH	F1-score	0.95, 0.95	0.86, 0.89	80%

On training the model, we expected F1-score to be 0.95, 0.95 for 0,1 respectively but was 0.78, 0.83 when trained on 70% of data. Amazingly, F1-score was improved to 0.86, 0.89 when trained on 80% of the data.

Step 10: Evaluate the fairness of your model. Use subsets of data to assess the fairness in regards to the protected features.

As explained in step-4, the feature importance table shows that gender and ethnicity which are our protected features lie very below in the table. Hence, intrinsically they do not impact much on the target variable.

Step 11: Build a pipeline to ingest your data, train the model and deploy it in a specific folder.

For details, please refer to the code-notebook.

Step 12: Suggest a way to calculate the software and business metrics for your pipeline.

The best way to design any machine learning model is CRISP-DM(Cross-Industry Standard Process for Data Mining) and I have applied CRISP-DM to complete the task.

The software metrics can be tested using Jmeter and various other automation tools to test the latency. The code smells can be minimized if we follow design principles.