CS205 Final Project - Spring 2019

Soundarya Tekkalakota
Department of Computer Science
University of California, Los Angeles
Los Angeles, CA 90024
soundaryat@ucla.edu

Abstract

The aim of this project is to build a predictive model to predict being positive to cancer or malignancy. CDC NHANES provides access to a large number data sets to perform analytics on. Machine Learning models were used on carefully chosen and pre processed causal factors. They provided qualitative and conclusive evidence on how Cancer is caused by and the degree to which each factor contributes to the risk.

1 Introduction

Prediction or prognosis of Cancer involves risk assessment that is based on trying to predict the likelihood of developing a type of cancer prior to the occurrence of the disease.It typically involves a hospital setting with multiple physicians from different specialties using different subsets of bio markers and multiple clinical factors. Early prognosis has now become a necessity in cancer research. Machine Learning is one of the most promising technologies that could be utilised today in this regard. The ability of ML tools to detect key features from complex data sets makes them viable to be used in these scenarios. The predictive models applied in this project are based on various supervised ML techniques as well as on different input features and data samples.

2 Preliminaries

2.1 Data Set

The project is based on the Data Set provided by CDC NHANES. It is a program of surveys designed to assess the health and nutritional status of adults and children in the United States. It consists of surveys on demographic, socioeconomic, dietary, and health-related components. I have explored the data set to determine which of the surveys can be mined for relevant information. I further explored each individual category to identify features that could be used to construct a predictive model that will be able to predict having cancer or malignancy.

2.2 Target Variable

The target variable for this project is MCQ220. It can be found under Questionnaire Data -> Medical Conditions -> MCQ220 - Ever told you had cancer or malignancy. For the purpose of running ML models on the dataset, we need to split the data set into training and testing subsets.

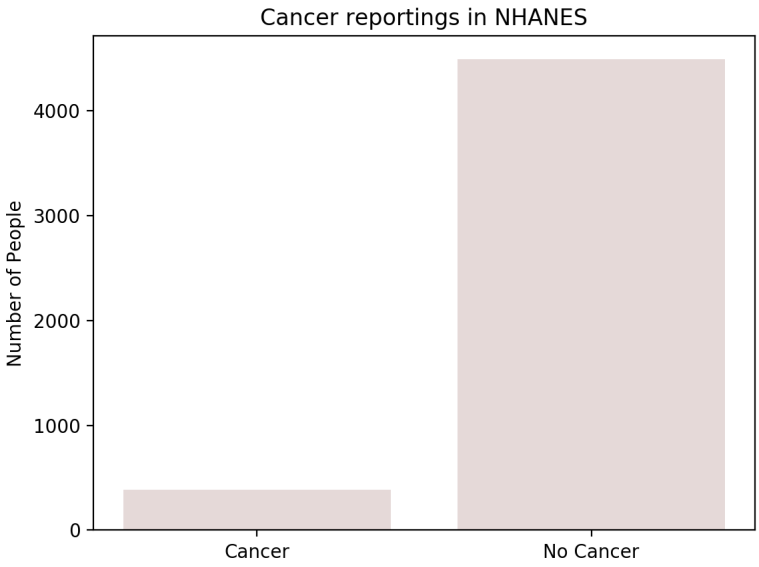
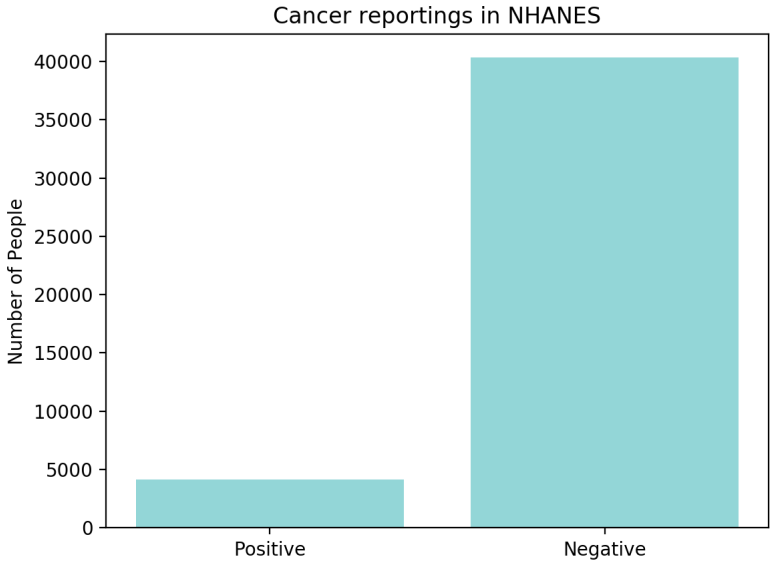


Figure 1 shows the distribution of the "MCQ220" feature also known as the ground truth label in biased data set. while Figure 2 shows that of the unbiased data set. There is acute skewness in the lable as a very small percentage of the patients (around 9.2%) responded saying they have/had been informed of malignancy. This makes our training problem even more difficult. Training on this data, I realized that is simply naive. Even if the model ignores all the positive labels and outputs false for all predictions, it would still have a 90% accuracy. This is misleading because it does not identify any of the positive cases correctly. To tackle this problem, I ran each experiment on both a balanced and an imbalanced data set.



dataset.png dataset.png

2.2.1 Balanced data set

To create a balanced data set for training and testing, I ensured the distribution of both the labels (testing positive for cancer and negative for cancer) were equal. This forced me to make both the training set and the testing set a very small subset of the actual data set since the number of records with a ground truth label testing positive for malignancy was very low.

2.2.2 Imbalanced data set

To create an imbalanced training and test set, I used SKlearn's (insert reference) train_test_split function. This approach is called imbalanced because it directly divides the data into training and testing sets without accounting for the skew in the data set

2.3 Machine Learning:

ML, a branch of Artificial Intelligence, relates the problem of learning from data samples to the general concept of inference. Every 'learning' process consists of two phases: (i) estimation of unknown dependencies in a system from a given data set and (ii) use of estimated dependencies to predict new outputs of the system. ML has also been proven an interesting area in bio medical research with many applications. Different techniques and algorithms are used to derive a decision/prediction for a given set of observations.

2.4 Supervised and Unsupervised learning:

There are two main common types of ML methods- supervised learning and unsupervised learning. In supervised learning, a labeled set of training data is used to estimate or map the input data to the desired output. In contrast, under the unsupervised learning methods no labeled examples are provided and there is no notion of the output during the learning process. As a result, it is up to the learning scheme/model to find patterns or discover the groups of the input data. In supervised learning this procedure can be thought as a classification problem. The task of classification refers to a learning process that categorizes the data into a set of finite classes.

2.5 Approaches employed:

The following are some of the techniques used to run Machine Learning Models on the pre-processed data set features:

2.5.1 Logistic Regression:

Logistic regression models the probabilities for classification problems with two possible outcomes. It is an extension of the linear regression model for classification problems. It is a better alternative to linear regression. This is because instead of fitting a straight line or hyperplane, the logistic regression model uses the logistic function to squeeze the output of a linear equation between 0 and 1.

However, Logistic Regression struggles with restrictive expressiveness. Interactions must be added manually and so there may be other models that might have a better predictive performance in this regard. Another disadvantage of the logistic regression model is that the interpretation is more difficult because the interpretation of the weights is multiplicative and not additive. If there is a feature that would perfectly separate the two classes, the logistic regression model can no longer be trained.

2.5.2 Random Forests:

Random forests are bagged decision tree models that split on a subset of features on each split. A decision tree splits the data set into smaller data groups based on the features until there are data points under just one label. The splits are decided according to a purity measure. Decision trees are now combined/bagged to create a forest with a sampling technique called bootstrapping. In Bootstrapping we randomly sample with replacement from the data set. The final prediction value of the forest would be the average value of all the decision trees.

Random forests are known for their versatility. It can handle binary features, categorical features, and numerical features. There is very little pre-processing that needs to be done. The data does not need to be re scaled or transformed. They are also parallelizable. Given the vast number of features I had from the massive data set, this makes it ideal to use Random Forests in the project.

2.5.3 Support Vector Machines:

Support Vector Machines is yet another supervised learning model. An SVM training algorithm will assign the set of training examples, each marked as belonging to one or the other of two categories. An SVM constructs a hyperplane or set of hyperplanes in a high dimensional space. This can help with classification or outlier detection. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class.

However, SVMs require full labelling of data which can be a tedious task. Although not an issue in this project, an SVM is intended for two-class tasks. So, algorithms for multi-class tasks need to reduce it to binary.

2.5.4 K Nearest Neighbours:

Yet another supervised learning method, KNN classification input consists of the k closest training examples in the feature space and the output is the class membership of the training example. It is based on feature similarity. It considers how closely out-of-sample features resemble my training set and this in turn determines how we classify a given data point. The class membership is decided by the majority vote of the neighbors. The object will be assigned to the class most common among its k nearest neighbors. KNN doesn't take any assumptions about data. This can be useful, for example, for nonlinear data. It is a straight forward algorithm which has relatively high accuracy. However, it is not as competitive in comparison to better supervised learning models because it is sensitive to irrelevant features and the scale of the data.

2.5.5 K Means

K means is used for cluster analysis of data.This can be semi-supervised or unsupervised learning. The objective of K Means is to group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset. K-means stores k centroids that it uses to define clusters. K-Means finds the best centroids by alternating between (1) assigning data points to clusters based on the current centroids (2) chosing centroids (points which are the center of a cluster) based on the current assignment of data points to clusters. A cluster is a collection or group of data points aggregated together because of certain similarities. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid. This results in a partitioning of the data set. K means like KNN is very sensitive to noisy features in the data set.

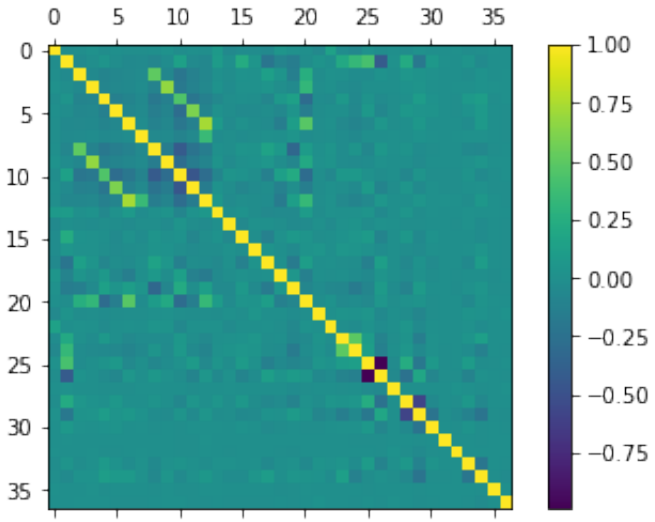
2.6 Feature Selection and Evaluation:

I will now elaborate on the intuition behind choosing the relevant features under various categories for the purpose of this project. I used the following measures to assess which features to choose quantitatively in addition to applying my own intuition.

2.6.1 Correlation

Correlation refers to a mutual relationship or association between quantities. Often, correlation is the first step to understanding relationships and subsequently building better statistical models. It can show whether and how strongly pairs of variables are related. I plotted a correlation matrix below which is a table showing correlation coefficients between variables. Each cell in the table shows the correlation between two variables.

The main result of a correlation is called the correlation coefficient. It ranges from -1.0 to +1.0. The closer r is to +1 or -1, the more closely the two variables are related. If r is close to 0, it means there is no relationship between the variables. If r is positive, it means that as one variable gets larger the other gets larger. If r is negative it means that as one gets larger, the other gets smaller.



I found that it is a nice way to summarize data in order to see patterns. The yellow indicates that the variables are highly correlated with themselves which is to be expected. There are other features that are slightly higher in correlation than others. This indicates a stronger relationship/higher positive correlation. There are also some features with a very high negative correlation(indicated by blacker shade). I could either choose to drop or retain these features. But I chose to retain them because there were not as many strongly positively correlated features as negative ones and they both would ultimately lead to similar effects on the predictions.

There are ways to measure Correlations between variables. One of the most popular method is using Pearson's correlation Coeffient.

Pearson's Correlation Coeffient: This is a measure of linear correlation or covariance of the two variables divided by the product of their standard deviations. The correlation coefficient ranges from 1 to 1. A value of 1 implies that a linear equation describes the relationship between X and Y perfectly, with all data points lying on a line for which Y increases as X increases. A value of 1 implies that all data points lie on a line for which Y decreases as X increases. A value of 0 implies that there is no linear correlation between the variables.

2.6.2 Mutual Information

Mutual Information quantifies the amount of information one can obtain about one variable by observing the other random variables. The concept of mutual information is intricately linked to that of entropy of a random variable, a fundamental notion in information theory that quantifies the expected "amount of information" held in a random variable. Mutual Information is positive in general. A MI of zero means that the two variables are independent. High mutual information indicates a large reduction in uncertainty while a low mutual information indicates a small reduction. I applied this to eliminate those features with zero MI as they wouldn't contribute to the dependencies.

```
[4.75867422e-03 4.13734490e-02 8.24486208e-03 1.48392863e-02
 4.04132656e-03 6.16856881e-04 9.22953939e-04 0.00000000e+00
 2.92002431e-03 3.62490419e-04 3.52300236e-03 2.99191928e-03
 1.56930290e-03 4.45217989e-03 2.36807047e-03 7.49067988e-04
 2.83430750e-03 2.75104525e-03 3.28391235e-03 4.94090901e-03
 0.00000000e+00 1.37997555e-03 9.43069003e-04 0.00000000e+00
 1.96871079e-03 3.70728018e-05 9.38261726e-04 3.95887267e-03
 4.33204781e-04 3.64953516e-03 5.92547436e-04 1.67530971e-03
 0.00000000e+00 0.00000000e+00 1.88433940e-03 7.42161517e-04
 0.00000000e+00 1.09478360e-03 6.49511640e-04 0.00000000e+00
 0.00000000e+00 0.00000000e+00 0.00000000e+00 0.00000000e+00
 0.00000000e+00 0.00000000e+00 9.68095198e-03 4.71417442e-03
 3.94815715e-03 2.14343325e-03 1.03991717e-03 8.15373899e-04
 0.00000000e+00 4.21984734e-04 4.47053347e-04 2.75481257e-03
 0.00000000e+00 1.08610940e-03 2.66229175e-03 4.51896373e-03
 1.45326525e-02 1.10086282e-02 8.47184787e-05 2.67310912e-04
 1.66705621e-03 0.00000000e+00 1.56521232e-03 5.55871591e-04
 4.02324327e-04 2.34639971e-03 2.04358560e-03 0.00000000e+00
 1.75752025e-03 1.80878438e-05 3.40207931e-03 5.73675693e-03
 1.35701299e-02 1.22165477e-02 3.38293531e-03 1.95729496e-03
 5.77404182e-04 3.17322891e-03 0.00000000e+00 3.33976848e-04]
```

MI and manually checked if they contained NANs

I had originally collected around 70 features from all the categories. I calculated the Mutual Information on all these features and found the array to be sparsely populated with many zero MIs. I filtered out these features to get the array below. The array consists of forty features with a few zeroes indicating a certain degree of uncertainty between these variable relationships.

2.6.3 F test:

These are the uni-variate linear regression tests. F tests are widely used during feature selection as a scoring function. This is done in 2 steps: The correlation between each regressor and the target is computed and it is converted to an F score then to a p-value. The following are the values for f-test that compute the ANOVA F-value for the provided features. A low F-value shows a case where the group means are close together (low variability) relative to the variability within each group. The high F-value shows a case where the variability of group means is large relative to the within group

```
[ 3.41790432e-01  3.88785619e+03  2.06295080e+01  2.16759907e+01
  1.05152538e+01  2.95709878e+00  7.68519018e+01  7.73989261e+00
  7.00548622e+00  4.30306995e+00  1.08599315e+01  1.60805985e-01
  4.32057111e+01  3.30128513e-01  7.09156066e+01  1.74135571e+01
  1.17963315e+01  2.49973094e+01  2.11110927e+02  2.11060911e+02
  4.00144300e-01  2.85420901e-03  1.14738060e+01  8.81431872e+00
  1.31412903e-01  4.82774468e+00  1.10702256e+00  1.96654412e+01
  8.52246947e-01  7.95355363e+01  1.92473682e+01  4.15755241e-01
  1.34517829e+01  1.93780052e+01  8.28930745e+01  3.02015872e+00
  2.64744931e+01  2.71325609e+01  1.49934631e+01  8.99378699e+00
  7.31951558e+00  1.74666424e-01  2.37486508e-01  2.39929791e-03
  2.60090768e-01  1.18680606e-02  3.89963017e+02  5.05680173e-02
  6.02275291e+01  5.33897450e+00  6.23132769e+00  1.04881962e+01
  1.53012505e+01  1.10897113e+01  2.98529298e-02  4.24729826e+00
  8.97401399e+01  9.14143679e+00  1.25644446e+00  1.62269655e-01
  1.41560319e+00  4.35514672e+02 -4.56548963e+00  2.71973805e+01
  5.66936382e+01  4.04979312e+00  2.98930069e+01  4.87029159e+00
  1.03284536e-01  1.75459309e+00  4.75583998e+00 -6.79349320e+00
  1.74482311e+02  1.15630958e-01  1.67121653e+02  3.08720244e+02
  9.43490460e+02  3.56882464e+02  8.27792043e+01  3.01214489e+01
 -4.30978682e+01  3.01760638e+01  1.90268195e+00  3.99153958e+01]
```

variability.

2.7 Feature Evaluation:

The following is a description of all the features I had chosen and reasoned as to why I discarded a few of them.

Demographics Data: .

The demographics file provides individual, family, and household-level information on the topics such as gender(RIAGENDR), Age(RIDAGEYR),Race and Ethnicity variable (RIDRETH1),

Education Level(DMDEDUC2),Country of Birth(DMDBORN4), Family Income(INDFMIN2) as certain types of cancer affect certain demographics under or above a certain age bracket with higher probability. [5] Females and males are also at different risk of cancer [8]. Race and ethnicity do impact cancer as studied here [9].

Demographics Data	
Feature Code	Feature Name
RIAGENDR	Gender
RIDAGEYR	Age
RIDRETH1	Race/ethnicity
INDHHINC	Annual Household Income
DMDEDUC2	Education Level
INDFMIN2	Annual Family Income
INDFMPIR	Ratio of family income to poverty
DMDEDUC2	Education level - Adults
DMDEDUC3	Education level - Child
DMDBORN4	Country of Birth

Dietary Data: .

The objective of the dietary interview component is to obtain detailed dietary intake information from NHANES participants. The dietary intake data are used to estimate the types and amounts of foods and beverages (including all types of water) consumed during the 24-hour period prior to the interview (midnight to midnight), and to estimate intakes of energy, nutrients, and other food components from those foods and beverages. The intake of Sodium is studied and seems to be an influencing factor [4].

Dietary Data	
Feature Code	Feature Name
DBD100	How often add salt to food at table
DR1320Z	Total plain water drank yesterday (gm)
DR2TSODI	Sodium (mg)
DR1ICHOL	Cholesterol

Examination Data: .

NHANES body measures data are used to monitor trends in infant and child growth, to estimate the prevalence of overweight and obesity in U.S. children, adolescents, and adults, and to examine the associations between body weight and the health and nutritional status of the U.S. population. There have been studies indicating a strong correlation between calories intake and subsequent body measurements and cancer. [1]

Examination Data	
Feature Code	Feature Name
BMXBMI	BMI
BMXWAIST	Width of the Waist
BMXHT	Height (mg)
BMXWT	Weight
OHDEXSTS	Oral Health

Laboratory Data: .

In addition to features that describe the lifestyle, diet and body measurements as possible contributors to cancer, quantitative evidence by lab tests show that the presence of certain compounds such as flouride, lead or long term exposure to mercury increase risk of cancer. [7] Apolipoprotein E (ApoE) has been recently identified as a potential tumor-associated marker in ovarian cancer by serial analysis of gene expression.[3]

Laboratory Data	
Feature Code	Feature Name
URXUAS	Arsenic-Total-Urine
LBDBPBSI	Lead,Cd,Total Hg-Blood
LBDBCDSI	Lead,Cd,Total Hg-Blood
LBDTHGSI	Lead,Cd,Total Hg-Blood
LBDBSESI	Lead,Cd,Total Hg-Blood
LBDBMNSI	Lead,Cd,Total Hg-Blood
WTSH2YR	Mercury:Inorganic, Ethyl and Methyl-Blood
LBDIHGSI	Mercury: Inorganic, Ethyl and Methyl – Blood
ORXHPV	Human Papillo-mavirus (HPV) - Oral Rinse
URXUMA	Urine
URDACT	Urine
LBDAPBSI	Apolipoprotein
LBDWFL	Fluoride, water (mg/L) average 2 values

Questionnaire Data: . This is a massive sub category of the dataset which consists of many important factors that impact the possibility of cancer. Out of the many, I have filtered out features

like long term Stomach or intestinal illness(HSQ510) that cause gastrointestinal cancers, [2], sunburn [10],how poor sleep quality lowers immunity and can cause cancer [6] etc.

Questionnaire Data	
Feature Code	Feature Name
HUQ010	Health Condition
DEQ038G	Sunburn
HIQ011	Health Insurance
ECQ020	Mother Smoked
DBQ700	Diet
LBXHBC	Heapatitis B
OCQ210	Usually work 35 or more hours per week
MCQ100	Told have high blood pressure
WHD080S	Ate less sugar, candy, sweets
WHD080T	Ate less junk food or fast food
SLQ030	How often do you snore?
ALQ120U	days drink alcohol per wk, mo, yr
IMQ011	Received Hepatitis A vaccine
IMQ020	Received Hepatitis B 3 dose series
BPQ020	Blood pressure
BPQ080	Has doctor told you that you High Cholesterol level
DIQ010	Has doctor told you that you have Diabetes
HSQ510	Stomach or intestinal illness
PAQ685	air quality bad
SLQ060	ever been told that you have a sleep disorder
IMQ060	Received HPV vaccine (Females)
IMQ070	Received HPV vaccine (Males)
DBQ301	Community/Government meals delivered

DBD900	of meals from fast food or pizza place
DBD905	of ready-to-eat foods in past 30 days
DBQ360	Attend kindergarten thru high school
ALQ101	Alcohol consumption
ALQ120Q	alcohol consumption
PAQ605	vigorous work activity
PAQ620	vigorous work activity 2
MCQ160J	doctor told was overweight
SLD010H	sleep
SMQ020	smoking
SMD030	smoking
HOD050	how many rooms do you have in the house
RHQ070	Age range at last menstrual period
SMQ665A	Marlboro variety
SMQ040	Do you now smoke cigarettes?
SMD100TR	FTC Tar Content
SMD100NI	FTC Nicotine Content
SMD100CO	FTC Carbon Monoxide Content
WHD080Q	Ate more fruits, vegetables, salads

2.7.1 Feature Elimination:

After selecting these tabulated features in the first round, to aid in less relevant feature elimination, I used Mutual Information and observed them for missing values(NaNs). Firstly, I dropped features that had too many missing values. Then, I calculated the Mutual Information Score for each feature and dropped many of the features based on low mutual information score values.

Eliminating empty features Out of the set of features, many of the features were empty i.e. most of the entries were N/A or missing. For example under Questionnaire, "How healthy is the diet" DBQ700SAS, "Education Level-Child" (DMDEDUC3) also had too many missing values. "Age range at last menstrual period" (RHQ070SAS) has only a total of 53 records with values. I manually looked up each feature and ensured it has enough values to be a part of the training dataset. From 84 features, the dataset size dropped to contain only 57 features.

Eliminating features based on Mutual Information Score When I calculated the Mutual Information Score for each feature on the dataset containing 57 features, it resulted in an extremely sparse matrix. I then dropped some more of the features that corresponded to a 0 mutual information score,

and obtained better results when I used the new filtered dataset on all my models. After this my dataset consisted of these features-

- MCQ220- Target variable-cancer or malignancy
- RIAGENDR-Gender
- RIDAGEYR-Age
- RIDRETH1-Race/ethnicity
- PAQ620-Moderate Work Activity
- DBD100- How often add salt to food at table
- SMQ040-Do you now smoke cigarettes?
- LBDAPBSI-Apolipoprotein
- HIQ011-Health Insurance
- LBXHBC -Hepatitis B
- DMDEDUC2- Education Level-adults
- DMDBORN4-Country of birth
- SLQ030-How often do you snore?
- ALQ120U-days drink alcohol per wk, mo, yr
- IMQ011- Received Hepatitis A vaccine
- IMQ020-Received Hepatitis B vaccine
- BPQ020-Blood pressure
- DIQ010-Has doctor told you that you have Diabetes
- SLQ060-ever been told by a doctor that has sleep disorder

However, I realised that removing features affected my accuracy and recall for the worse. Which meant that the fault lied not in features but the way they were processed. So, I reviewed the old feature set, and changed the pre-processing functions, added a few combinations of pre processings as discussed below to improve my metrics without sacrificing feature information.

2.8 Data Pre-Processing:

After I had chosen the features above, I had to pre-process them. Data pre-processing is an integral step in Machine Learning as the quality of data and the useful information that can be derived from it directly affects the ability of a model to learn; therefore, it was extremely important that I pre-processed the data before feeding it into a model. Some popular pre-processing methods that are commonly applied are:

2.8.1 Handling Null Values:

Any real world data set has a lot of null values or NaNs. No model can handle these NULL or NaN values on its own so intervention is needed.There are various ways for us to handle this problem. The easiest way to solve this problem is by dropping the rows or columns that contain null values. This is done by df.dropna() method call. I had dropped features like DBQ700SAS, DMDEDUC3, HSQ510 and RHQ070SAS as they had too many Null values. However it is not the best option to remove the rows and columns from our dataset as it can lead to loss of valuable information. A better alternative is to use imputation.

2.8.2 Imputation:

Imputation is simply the process of substituting the missing values of the data set by another value, preferably the mean. The method preproc-impute in my project is meant to handle this. In my data set, I had sparsely populated features like LBDAPBSI (Apolipoprotein) and Total water drank yesterday(DR1-320Z) and Annual household income (INDHHIN2) that I imputed with the mean.

2.8.3 Standardization:

It is another integral pre-processing step.In Standardization values are transformed such that the mean of the values is 0 and the standard deviation is 1. So in simple terms I just calculated the mean and standard deviation of the values and then for each data point I just subtract the mean and divide it by standard deviation. The method preprocreal in my project is meant to handle this.

I normalized the features such as Annual Household income(INDHHIN2), BMI(BMXBMI),height, weight, total cholesterol, alcohol consumption(ALQ120Q) that had a large range of values by applying this preprocessing technique.

2.8.4 Handling Categorical Variables:

Handling categorical variables is another integral aspect of Machine Learning. Categorical variables are basically the variables that are discrete and not continuous.

Ordinal categorical variables These variables can be ordered.

Nominal categorical variables These variables cannot be ordered.

The correct way of handling nominal categorical variables is to use One-Hot Encoding. The easiest way to use One-Hot Encoding is to use the getdummies() method.In this,n columns are created where n is the number of unique values that the nominal variable can take. Out of the n columns only one column can have value = 1 and the rest all will have value = 0 for that variable. In other words, the variable is binary vectorized.

Features such as Smoking(SMQ020), How often add salt to food at table(DBD100),Diet are some of the features that could be one hot encoded.

2.8.5 Pre-processing using Binning :

Sometimes, it is useful to group values of features into sub ranges/bins. Data binning (also called bucketing) is a data pre-processing technique used to reduce the effects of minor observation errors. The original data values which fall in a given small interval, a bin, are replaced by a value representative of that interval, often the central value. It is a form of quantization. Bining is the process of transform numerical variable into categorical counterparts.

I've identified features like Age, Arsenic(URXUAS), Lead, Cadmium, Total Mercury, Selenium Manganese in blood(LBDTHGSI) that could benefit from binning.Numerical variables are usually discretized in the modeling methods based on frequency tables (e.g., decision trees). Moreover, binning may improve accuracy of the predictive models by reducing the noise or non-linearity. Finally, binning allows easy identification of outliers, invalid and missing values of numerical variables.

2.8.6 Cut-off for preprocessing :

In many of the features, the dataset has a lot features values above which they were either "refused to answer", "Not available" or "Missing". I set cut off values in order to limit values to the appropriate bins ranges. Most features that needed cut-off had "yes" or "no" and it made sense to cut off at that level.

2.9 Predictive Modeling Evaluation and Results:

In this section, I will describe the results of the ML models we have discussed in the preliminaries. Results can broadly be described using Accuracy, Precision and Recall. For result visualization, I have plotted ROCs and Confusion matrices for each type of classifier.

Accuracy: Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right.

$$Accuracy = Number\ of\ correct\ predictions/total\ number\ of\ predictions \quad (1)$$

Accuracy alone doesn't tell the full story when you're working with a class-imbalanced data set, like this one, where there is a significant disparity between the number of positive and negative

labels.In other words, my model accuracy indicate no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.

Precision: Precision answers what proportion of positive identifications was actually correct.

$$Precision = TruePositive/TruePositive + FalsePositive \tag{2}$$

A model that produces no false positives has a precision of 1.0 while a model that predicts more false positives will have a lower precision.

Recall: Recall tries to answer what proportion of actual positives was identified correctly.

$$recall = TruePositive/TruePositive + FalseNegative \tag{3}$$

A model that produces no false negatives has a recall of 1.0.

F1 Score: To fully evaluate the effectiveness of a model, one must examine both precision and recall.F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution

ROC curve: An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate. AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).It measures the entire two-dimensional area underneath the curve. It is a measure of how well a parameter can distinguish between two diagnostic groups. Often used as a measure of quality of the classification models. A random classifier has an area under the curve of 0.5, while AUC for a perfect classifier is equal to 1.

Confusion Matrix: A confusion matrix is a table that is often used to describe the performance of a classification model or classifier on a set of test data for which the true values are known. The confusion matrix has the following labels:

- true positives (TP): These are cases in which we predicted yes (they have the disease), and they do have the disease.
- true negatives (TN): We predicted no, and they don't have the disease.
- false positives (FP): We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- false negatives (FN): We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

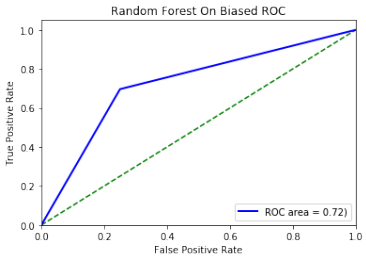
2.9.1 Biased Data set:

The following are the results on Biased data set with Random Forest, SVC and Logistic Regression classifier models:

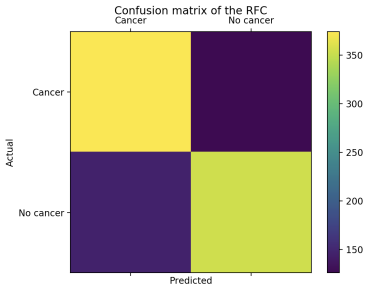
Random Forest Biased: .

```
accu_tst_RFC 0.728
Precision recall score RFC (0.708, 0.7375, 0.7224489795918367, None)
              precision    recall  f1-score   support

           0       0.72      0.75      0.73       500
           1       0.74      0.71      0.72       500

   micro avg       0.73      0.73      0.73      1000
   macro avg       0.73      0.73      0.73      1000
weighted avg       0.73      0.73      0.73      1000
```

The accuracy of Random Forest on biased data set came to 72 percent. This is impressive as it is above the baseline accuracy. The more important factors like recall(0.75) and precision(0.72) are also



(a) ROC



(b) Confusion Matrix

Figure 1: Random Forests Biased

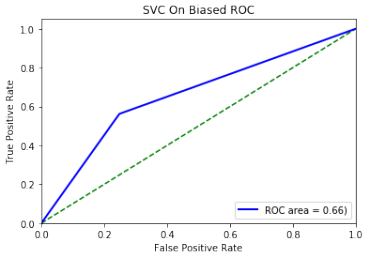
```
accu_tst_SVC 0.657
Precision recall score SVC (0.562, 0.6938271604938272, 0.6209944751381217, None)
              precision    recall  f1-score   support

           0       0.63      0.75      0.69       500
           1       0.69      0.56      0.62       500

   micro avg       0.66      0.66      0.66      1000
   macro avg       0.66      0.66      0.65      1000
weighted avg       0.66      0.66      0.65      1000
```

high. As indicated by the confusion matrix, the classifier is able to predict the true positive and true negatives-the possibility of having cancer and correctly predicting cancer and the possibility of not having cancer and correctly predicting that. This is shown by the lighter shaded TP and TN.

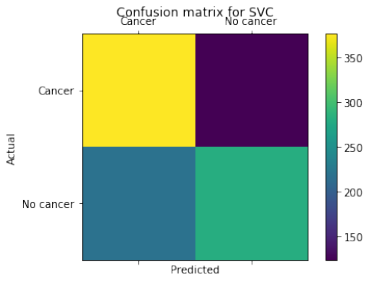
The area under the curve of a ROC is an indicator of the classifier's classification ability. RFC with biased data performs significantly well with a AUC=0.72.

SVC Biased: .

The accuracy of SVC is 65.7 percent on the biased data set. Although this is lower than the baseline, it has a good precision(0.69) and recall(0.75). The strength of these metrics is supported by the confusion matrix with higher number of true positive and true negatives. AUC is 0.66 to reflect the same idea.

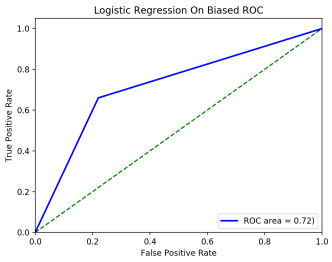


(a) ROC

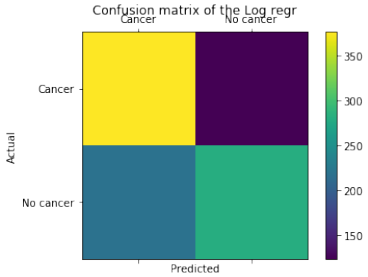


(b) Confusion Matrix

Figure 2: SVC Biased



(a) ROC



(b) Confusion Matrix

Figure 3: Logistic regression Biased

LR Biased: .

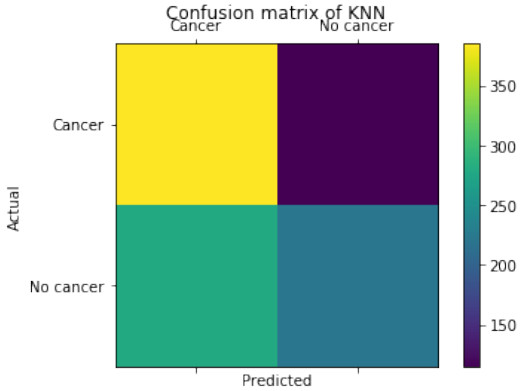
```
accu_tst_LR 0.72
Precision recall score Log regression (0.66, 0.75, 0.702127659574468, None)
              precision    recall  f1-score   support

           0       0.70      0.78      0.74       500
           1       0.75      0.66      0.70       500

   micro avg       0.72      0.72      0.72      1000
   macro avg       0.72      0.72      0.72      1000
weighted avg       0.72      0.72      0.72      1000
```

Logistic Regression performs as well as SVC and Random Forests on biased dataset with an accuracy of 72 percent and a precision and recall of 0.78 and 0.70 respectively. Confusion matrix and Area under the ROC shown below indicate its performance.

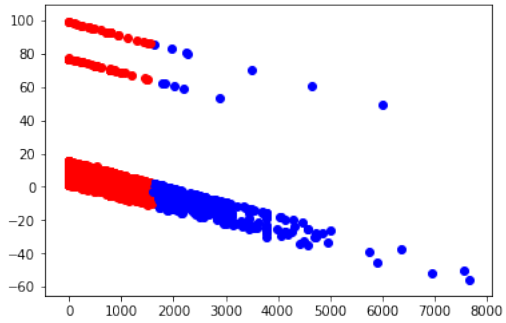
KNN Biased: .

```
acc_KNN 0.606
              precision    recall  f1-score   support

           0       0.58      0.77      0.66       500
           1       0.66      0.44      0.53       500

   micro avg       0.61      0.61      0.61      1000
   macro avg       0.62      0.61      0.60      1000
weighted avg       0.62      0.61      0.60      1000
```

KNN shows promising results with a good recall of 0.77 although precision is on the lower end at 0.58. The ROC shows the curve I had plotted was being very close to being a chance classifier. The confusion matrix below however shows the reflection of the moderately well performing model.



K Means with SVD: .



K Means shows two clusters one for cancer being present as shown in red and another for cancer being absent and shown in blue. The clusters are distributed in that manner as the data set is biased. However, there seems to be some merit to this clustering as it shows a similar distribution between the cancer and no cancer predictions.

MLP classifier: .

acc_Neural 0.687

	precision	recall	f1-score	support
0	0.58	0.77	0.66	500
1	0.66	0.44	0.53	500
micro avg	0.61	0.61	0.61	1000
macro avg	0.62	0.61	0.60	1000
weighted avg	0.62	0.61	0.60	1000

This is a Multi-layer Perceptron classifier.This model optimizes the log-loss function using LBFGS or stochastic gradient descent. MLPClassifier trains iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters. I obtained 68 percent accuracy with a reliable recall of 0.77 and a precision of 0.58.

2.9.2 Unbiased Data set:

The following are the results on running the same models as above on the unbiased data set:

Random Forests: .
Random forest on the unbiased data set has given 90 percent accuracy but has failed and given a 0.01 recall. This is clearly shown in the ROC curve and the confusion matrix as well.

acc_tst_RFC 0.9074868860276586

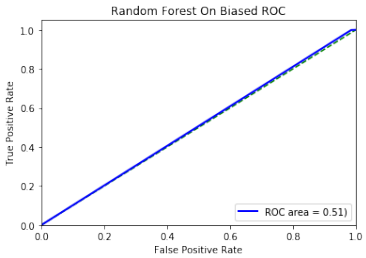
	precision	recall	f1-score	support
0	0.62	0.01	0.03	1366
1	0.91	1.00	0.95	13313
micro avg	0.91	0.91	0.91	14679
macro avg	0.77	0.51	0.49	14679
weighted avg	0.88	0.91	0.87	14679

Logistic Regression: .

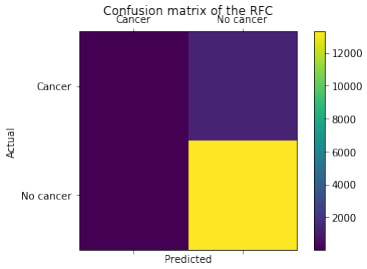
acc_LR 0.69330335853941

	precision	recall	f1-score	support
0	0.20	0.76	0.32	1366
1	0.97	0.69	0.80	13313
micro avg	0.69	0.69	0.69	14679
macro avg	0.58	0.72	0.56	14679
weighted avg	0.89	0.69	0.76	14679

Logistic Regression gave a 70 percent accuracy with a much better precision and recall of 0.20 and 0.76 respectively. However, it is interesting to see that it performs well enough to classify between

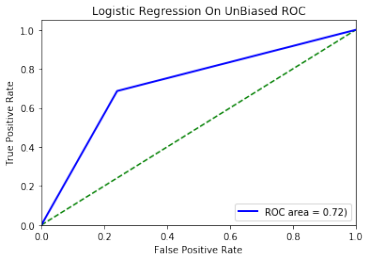


(a) ROC

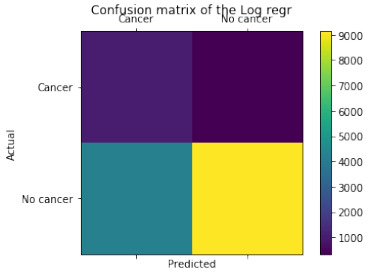


(b) Confusion Matrix

Figure 4: Random Forest UnBiased



(a) ROC



(b) Confusion Matrix

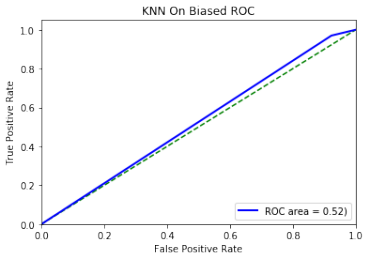
Figure 5: Logistic Regression UnBiased

cancer and no cancer as shown by the ROC.It is also notable that the true negatives are being classified more accurately than the true positives as shown in the confusion matrix.

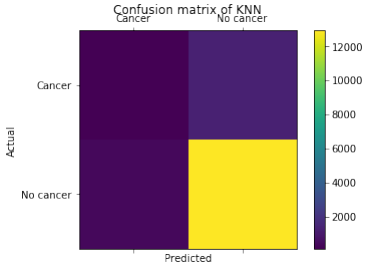
K nearest Neighbors: .

```
acc_KNN 0.8867089038762859
              precision    recall  f1-score   support

           0       0.21      0.08      0.11      1366
           1       0.91      0.97      0.94     13313

    micro avg       0.89      0.89      0.89     14679
    macro avg       0.56      0.52      0.53     14679
weighted avg       0.85      0.89      0.86     14679
```

K nearest neighbors performed slightly better than Log Regression on the unbiased data set at 89 percent accuracy and a recall of 0.08 and precision of 0.21.



(a) ROC



(b) Confusion Matrix

Figure 6: KNN Unbiased

MLP Classifier: .
MLP classifier did well on the unbiased data as well. It gave me a 91 percent accuracy over 5000 epochs with a decent recall of 0.08 and a precision of 0.21.

```
acc_Neural 0.9064650180530008
              precision    recall  f1-score   support

           0       0.21      0.08      0.11      1366
           1       0.91      0.97      0.94     13313

   micro avg       0.89      0.89      0.89     14679
   macro avg       0.56      0.52      0.53     14679
weighted avg       0.85      0.89      0.86     14679
```

2.10 Conclusion:

As per our problem statement, we are trying to predict the potential possibility of cancer. So, predicting positive took precedence over predicting negative. In this direction, I took a subsample of the dataset to ensure equal distribution of classes. But this subsample is biased and doesn't reflect the real dataset. In the NHANES dataset, the proportion of positive classes was only 9.2%. Therefore, by subsampling to create a balanced dataset, we would be only including 18.4% of the actual data.

Accuracy measures the fraction of predictions that the model got right and represents this as a percentage. In our model, there are only 9.2% cases of cancer. Therefore, even if the model simply predicted all patients to not have cancer, the accuracy would still be 90.8%. This is exactly how my models worked on the unbiased data set. So, I also judged my models not only based on accuracy but also on other metrics like Precision and recall.

Overall, I believe that incorporating analytics via Machine Learning Models to predict something so major and diverse as cancer will become an industry standard given that there is a growing deficit of physical health care providers and only an ever growing patient base. Combining computing and predictive powers with the age-old know-how of doctors is bound to bring more effective, targeted healthcare into the world.

References

[1] Demetrius Albanes. Caloric intake, body weight, and cancer: a review. *Nutrition and cancer*, 9(4):199–217, 1987.

[2] Kunio Aoki. Epidemiology of stomach cancer. In *Gastric Cancer*, pages 2–15. Springer, 1993.

[3] Yu-Chi Chen, Gudrun Pohl, Tian-Li Wang, Patrice J Morin, Björn Risberg, Gunnar B Kristensen, Albert Yu, Ben Davidson, and Ie-Ming Shih. Apolipoprotein e is required for cell proliferation and survival in ovarian cancer. *Cancer research*, 65(1):331–337, 2005.

[4] Birger Jansson. Potassium, sodium, and cancer: a review. *Journal of environmental pathology, toxicology and oncology: official organ of the International Society for Environmental Toxicology and Cancer*, 15(2-4):65–73, 1996.

[5] GR Newell, MR Spitz, and JG Sider. Cancer and age. 16(1):3–9, 1989.

[6] Michael J Sateia and Bianca J Lang. Sleep and cancer: recent developments. *Current oncology reports*, 10(4):309–318, 2008.

[7] Allan H Smith, Claudia Hopenhayn-Rich, Michael N Bates, Helen M Goeden, Irva Hertz-Picciotto, Heather M Duggan, Rose Wood, Michael J Kosnett, and Martyn T Smith. Cancer risks from arsenic in drinking water. *Environmental health perspectives*, 97:259–267, 1992.

[8] Declan Walsh, Sinead Donnelly, and Lisa Rybicki. The symptoms of advanced cancer: relationship to age, gender, and performance status in 1,000 patients. *Supportive Care in Cancer*, 8(3):175–179, 2000.

[9] Elizabeth Ward, Ahmedin Jemal, Vilma Cokkinides, Gopal K Singh, Cheryll Cardinez, Asma Ghafoor, and Michael Thun. Cancer disparities by race/ethnicity and socioeconomic status. *CA: a cancer journal for clinicians*, 54(2):78–93, 2004.

[10] Annemarie Ziegler, Alan S Jonason, David J Leffellt, Jeffrey A Simon, Harsh W Sharma, Jonathan Kimmelman, Lee Remington, Tyler Jacks, and Douglas E Brash. Sunburn and p53 in the onset of skin cancer. *Nature*, 372(6508):773, 1994.