

Ryan Shintaku
Ishaan Karvir
Bryce Cordill
Sam Wathen

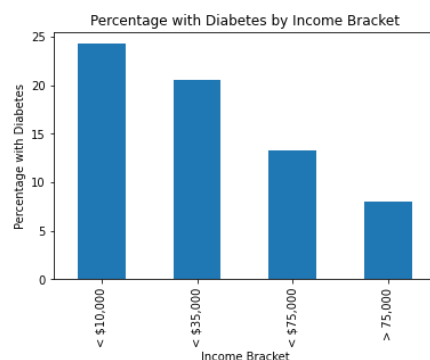
Predicting Diabetes from Health, Lifestyle, and Socioeconomic Factors in the US

Introduction

Diabetes is one of the most widespread chronic diseases in the US that affects over 37 million people and consists of two main types. Type 1 diabetes is caused by an autoimmune reaction that stops the body from making insulin which affects about 5-10% of diabetics and cannot be prevented. Type 2 diabetes is caused by the body's inability to use insulin well causing blood sugar levels to be irregular and affects about 90-95% of those with diabetes. However, Type 2 diabetes can be prevented or delayed with healthy lifestyle choices. The dataset used in this project focuses on Type 2 diabetes and the potential health, lifestyle, and socioeconomic related factors that can affect if a person has Type 2 diabetes. This dataset contains over two hundred thousand samples across twenty one feature variables¹ and the target variable (if they have Type 2 diabetes or not). Any categorical variables including the target variable have been encoded to numeric values. For example, the target variable has been encoded to 0 if the person does not have diabetes and 1 if they do have diabetes. This project uses the K Nearest Neighbors algorithm on various feature variables to try to predict if a person has Type 2 diabetes.

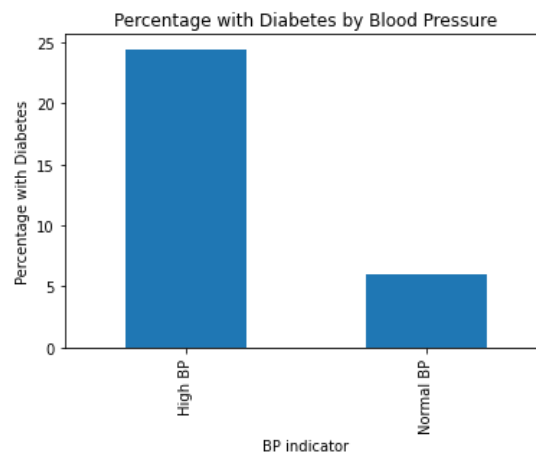
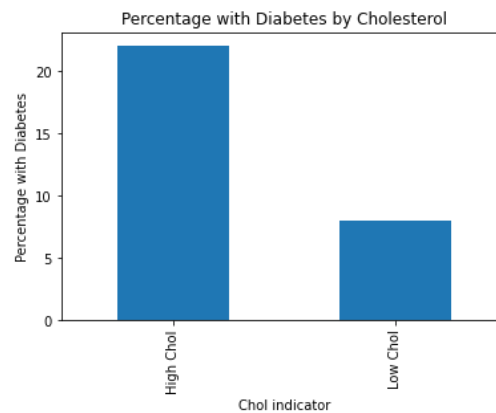
Preliminary Analysis

Before using the K Nearest Neighbors model, some preliminary analysis was performed to check if certain feature variables were possibly more correlated to a person having Type 2 diabetes. For each of the categorical variables, bar plots were created to compare the different groups within the variable to determine if there is a significant difference in percentage of those who have diabetes across the groups. One of the feature variables that had a significant difference in percentage with diabetes across different groups was the income variable. As seen below, nearly twenty five percent of people who make less than ten thousand a year have Type 2 diabetes whereas those who make more than seventy five thousand a year less than ten percent have diabetes.



¹ Complete list of feature variables: high blood pressure, high cholesterol, cholesterol check, body mass index, smoker, heart disease, stroke, physical activity, eats fruits, eats vegetables, heavy alcohol consumption, healthcare, visited doctor recently, general health levels, mental health, physical health, difficulty walking, sex, age, education, income

A couple other significant features that had a significant difference in percentage of those with diabetes were people with high cholesterol and/or high blood pressure.



As seen above, over twenty percent of people with high cholesterol have diabetes whereas less than ten percent of those with low cholesterol have diabetes. Similarly to the cholesterol levels, nearly twenty five percent of people with high blood pressure have diabetes compared to only about five percent of people with normal blood pressure have diabetes. Other feature variables that were deemed significant in the preliminary analysis include body mass index, whether the person eats at least one vegetable a day, whether they are heavy smokers, whether they are physically active, and whether they have had a stroke previously. These significant features were then used to predict whether a person has diabetes or not.

Methods/Results

To predict if a person has Type 2 diabetes, the K Nearest Neighbors (KNN) algorithm was implemented with $K = 11$. K is an odd number so that there are no ties when classifying an input row. Before running the data through the KNN algorithm, the data needed to be mapped to an RDD of tuples where the first element of the tuple is the index, so later on the accuracy of the model can be calculated. The second element of the tuple was another tuple where the first

element of this tuple was either 0 if the person is not diabetic or 1 if they are diabetic. The second element of the tuple was a list of the feature variables for the given person. An example of the mapped data can be seen below.

```
(0,(0.0,List(8.0, 0.0, 1.0, 26.0, 0.0, 0.0, 1.0, 1.0)))
(1,(0.0,List(8.0, 1.0, 1.0, 26.0, 1.0, 1.0, 0.0, 0.0)))
(2,(0.0,List(8.0, 0.0, 1.0, 26.0, 0.0, 0.0, 1.0, 1.0)))
(3,(0.0,List(8.0, 1.0, 1.0, 28.0, 1.0, 0.0, 1.0, 1.0)))
(4,(0.0,List(8.0, 0.0, 1.0, 29.0, 1.0, 0.0, 1.0, 1.0)))
```

After mapping the raw data to an RDD of tuples, the data was then split into 80% training and 20% testing to then be passed to the getNeighbors function. The getNeighbors function takes as input the training and testing data where the cartesian product is created between the testing set and training set. This allows every row in the testing set to be compared to every row in the training set. The result of the cartesian product is shown below.

```
((1,(0.0,List(8.0, 1.0, 1.0, 26.0, 1.0, 1.0, 0.0, 0.0))), (3,(0.0,List(8.0, 1.0, 1.0, 28.0, 1.0, 0.0, 1.0, 1.0))))
((1,(0.0,List(8.0, 1.0, 1.0, 26.0, 1.0, 1.0, 0.0, 0.0))), (4,(0.0,List(8.0, 0.0, 1.0, 29.0, 1.0, 0.0, 1.0, 1.0))))
((1,(0.0,List(8.0, 1.0, 1.0, 26.0, 1.0, 1.0, 0.0, 0.0))), (5,(0.0,List(7.0, 0.0, 1.0, 18.0, 0.0, 0.0, 1.0, 1.0))))
```

As seen above, the cartesian product creates an RDD of tuples where the first element of the tuple is a row from the testing set, and the second element of the tuple is a row from the training set. The euclidean distance is then calculated between the list of feature variables in the first element of the tuple and the list of feature variables in the second element of the tuple. The result is then mapped to a tuple where the first element is the index of the testing row, and the second element is a tuple of the index of the training row, calculated euclidean distance, and classification of the training row (if training row was diabetic or not).

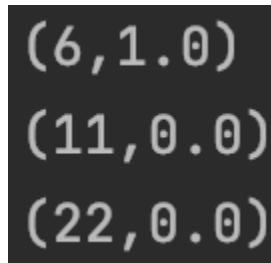
```
(96,(192,4.358898943540674,0.0))
(96,(193,3.4641016151377544,0.0))
(96,(194,6.557438524302,0.0))
```

The example tuple above was then grouped by the key so the values could then be sorted by the distance for each key in ascending order. After sorting the values by distance, the tuples with the smallest 11 distances were taken for each key since $K = 11$. This key-value pair was then remapped to a tuple where the first element was the index of the testing row, and the second element is a list of the diabetes classification variable corresponding to the 11 smallest distances for the key.

```
(1,List(1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0))
```

The example above shows the index of the testing row along with a list of 11 elements. These elements represent how the 11 closest neighbors to row 1 of the testing set are classified (0 is

nondiabetic, 1 is diabetic). From here, the highest frequency was found for the target variable of the K nearest neighbors which was then used to classify row 1 of the testing set. In the example above, 0 appears six times whereas 1 appears five times, so row 1 of the testing set would be classified as not diabetic. The output of getNeighbors is an RDD of tuples where the first element is the index and the second element is how the row is classified. Example output of the getNeighbors function and complete getNeighbors function can be seen below.



(6, 1.0)
(11, 0.0)
(22, 0.0)

```
def computeDistance(row1: List[Double], row2: List[Double]): Double = {  
  val distance = row1.zip(row2).map({case(x, y) => math.pow(x - y, 2)})  
  return math.sqrt(distance.sum)  
}  
  
def getNeighbors(train: RDD[(Long, (Double, List[Double]))], test: RDD[(Long, (Double, List[Double]))]) : RDD[(Long, Double)] = {  
  val data = test.cartesian(train).map({case((i1, (tar1, l1)), (i2, (tar2, l2))) =>  
    (i1, i2, computeDistance(l1, l2), tar2)})  
  }.groupByKey().map({case(key, vals) =>  
    (key, vals.toList)  
  }).sortByKey().mapValues(x =>  
    x.sortBy(y => (y._2)).take(N)).mapValues(x => x.map({case(i2, dist, tar2) =>  
    tar2})).mapValues(x =>  
    x.groupBy(identity).mapValues(_.size).maxBy(_._2)._1)  
  data  
}
```

After running the data through the getNeighbors function, the accuracy of the model was found by joining the result of getNeighbors to the key-value pair of the index of the testing row and actual target variable. Using the feature variables mentioned before, the model was correct about 62% of the time in determining if someone has Type 2 diabetes or not.

After trying the K Nearest Neighbors algorithm on the significant features from the preliminary analysis, other combinations of features were tested to compare if another set of feature variables would provide better accuracy. Using all feature variables from the dataset gave an accuracy of about 72%. The next set of features used were strictly health related features such as high blood pressure, high cholesterol, body mass index, stroke and heart disease. The accuracy when using only health related features was about 54%. This subset of features did not perform as well, so the next set of features tested were non-health related factors such as sex, age, income, access to healthcare, and education. This subset of features had an accuracy of 64% and performed slightly better than when using the significant features determined from preliminary analysis. The final set of features tested were the combination of health related factors and non health related factors previously mentioned. When using this combination of factors, the accuracy was about 74% and performed slightly better than using all features.

Conclusion

Overall, the K Nearest Neighbors algorithm on a combination of both health related factors and non related health factors performed the best when determining if a person has Type 2 diabetes. The K Nearest Neighbors algorithm and this dataset can help those who do not currently have diabetes determine if they are at risk of diabetes by running KNN on a person's response to the different feature variables. If the algorithm predicts that they have diabetes but they do not currently have diabetes, this person may be prediabetic. Early detection of being prediabetic can alert people that they need to make healthy lifestyle changes to prevent further risk of becoming diabetic.

Link to our dataset

<https://www.kaggle.com/alexteboul/diabetes-health-indicators-dataset>