Module 1 Project | Understanding Age Longevity in America

Trang Tran

CPS Analytics, Northeastern University

ALY6020 | Predictive Analytics

Professor Prashant Mittal

Nov 5, 2023

Introduction

This project focuses on developing a K-Nearest Neighbors (KNN) model by leveraging a county-level data set sourced from the County Health Rankings Organization, which includes a wide range of health and demographic indicators, such as diabetes prevalence, income, unemployment rates, % rural, smoking prevalence, and more. This model will be designed to predict whether a county falls into the category of "High" or "Low" life expectancy. In doing so, we aim to better understand the factors that contribute to disparities in life longevity across different counties and find the best hyperparameter for the according model.

Data Description and Preprocessing

The county dataset contains 1941 observations with 22 columns of features. We created a subset of the data set that includes one target variable: 'Life_Expectancy_2Cat' (‘Low’ and ‘High’ categories), and 13 predictors we intended to explore: 'CountyFIPS_Final' (County code), '% below 18 years of age raw value', 'Diabetes prevalence raw value', 'Unemployment raw value', '% rural raw value', 'Adult obesity raw value', 'Adult smoking raw value', 'Median household income raw value', 'Physical inactivity raw value', 'Population raw value', 'Uninsured raw value', 'Cat_pop' (Small, Small-Medium, Large, V Large), 'Cat_Income' (<50k, 50-60k, >60k).

- ***Handling missing data***

  Notably, this subset dataset contains only a very small number of NAs, as shown in Figure 1. To ensure data quality, we have opted to exclude the rows with missing values.

Additionally, we've performed a data type check, presented in Figure 2 below, to confirm the appropriate data types for our variables.

```
CountyFIPS_Final                      0
% below 18 years of age raw value    0
Diabetes prevalence raw value        0
Unemployment raw value               0
% rural raw value                    5
Adult obesity raw value              0
Adult smoking raw value              0
Life_Expectancy_2Cat                 0
Median household income raw value    1
Physical inactivity raw value        0
Population raw value                 0
Uninsured raw value                  0
Cat_pop                              6
Cat_Income                           1
dtype: int64
```

Figure 2: NAs check

```
CountyFIPS_Final                      int64
% below 18 years of age raw value    float64
Diabetes prevalence raw value        float64
Unemployment raw value               float64
% rural raw value                    float64
Adult obesity raw value              float64
Adult smoking raw value              float64
Life_Expectancy_2Cat                  object
Median household income raw value    float64
Physical inactivity raw value        float64
Population raw value                   int64
Uninsured raw value                  float64
Cat_pop                               object
Cat_Income                            object
dtype: object
```

Figure 1: Data types

- ***Creating Dummy Variables***

  Our dataset includes categorical variables, 'Cat_Income' and 'Cat_pop,' which represent categorizations of income and population, respectively. To incorporate these categorical variables into our KNN model effectively, we apply the concept of one-hot encoding. This process results in the creation of binary "dummy" variables that represent each category within the original categorical variables. The new dataset has a total of 19 columns.

- ***Defining Predictors and Response***

  With the dummy variables in place, we can now distinguish between predictors and the response variable. In our analysis, the predictors encompass a wide array of health and demographic factors that consist of 18 predictors, while the response variable is 'Life_Expectancy_2Cat,' which categorizes counties into 'Low' and 'High' life expectancy groups.

- ***Scaling the Predictors***

Standardizing the predictors is a crucial step in this KNN analysis due to the wide range of scales in the factors. This scaling process ensures that all predictors are on a common scale. In this step, we utilize the StandardScaler (z-scores method) to accomplish this standardization.

Implementation of the KNN Classification Model

First, we split the data set into the train, test, and validation sets with a ratio of 60%-20%-20% respectively.

Second, we determine the best optimal k value for the KNN model by using a loop to constantly train the KNN classifier on the train set and evaluate the model's prediction results on the validation set with a range of k values. The best result is k = 7 with the highest accuracy score on the validation test (Figure 3).

```
k = 2, Accuracy on Validation Set: 0.8886010362694301
k = 3, Accuracy on Validation Set: 0.9041450777202072
k = 4, Accuracy on Validation Set: 0.9041450777202072
k = 5, Accuracy on Validation Set: 0.9067357512953368
k = 6, Accuracy on Validation Set: 0.9119170984455959
k = 7, Accuracy on Validation Set: 0.9145077720207254
k = 8, Accuracy on Validation Set: 0.9145077720207254
k = 9, Accuracy on Validation Set: 0.9119170984455959
k = 10, Accuracy on Validation Set: 0.9145077720207254
Best k value is 7 with accuracy 0.9145077720207254
```

Figure 3: Results of finding the optimal k value for KNN model

Third, with the best k value in place, we fit the final KNN model on the train set and run the predicting process on the test set. The overall accuracy on the test set with best k is 93%. Other classification metrics such as Precision, Recall, and F1-score are presented below in Figure 4. A visualization of the confusion matrix is provided in Figure 5.

```
Overall Accuracy on Test Set with best k: 0.9300518134715026
Precision: 0.9255813953488372
Recall: 0.9476190476190476
F1-score: 0.936470588235294
```

Figure 4: KNN model's results

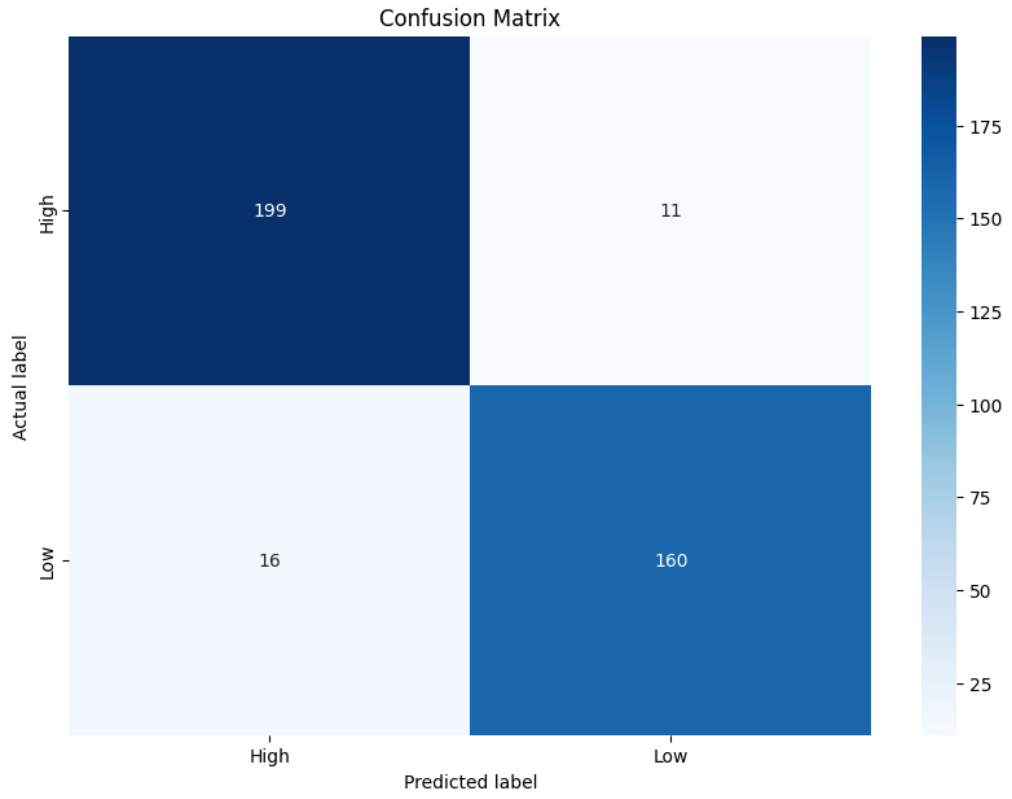

Figure 5: Confusion Matrix

Interpretation of results

The confusion matrix provides a detailed breakdown of the model's performance. In this model, ‘High’ stands for 0 (negative), and ‘Low” stands for 1 (positive). Because scikit-learn has a different arrangement of predicted and actual values in the confusion matrix (Suraj Gurav, 2023). Figure 6 below helps explain the concept of this matrix.

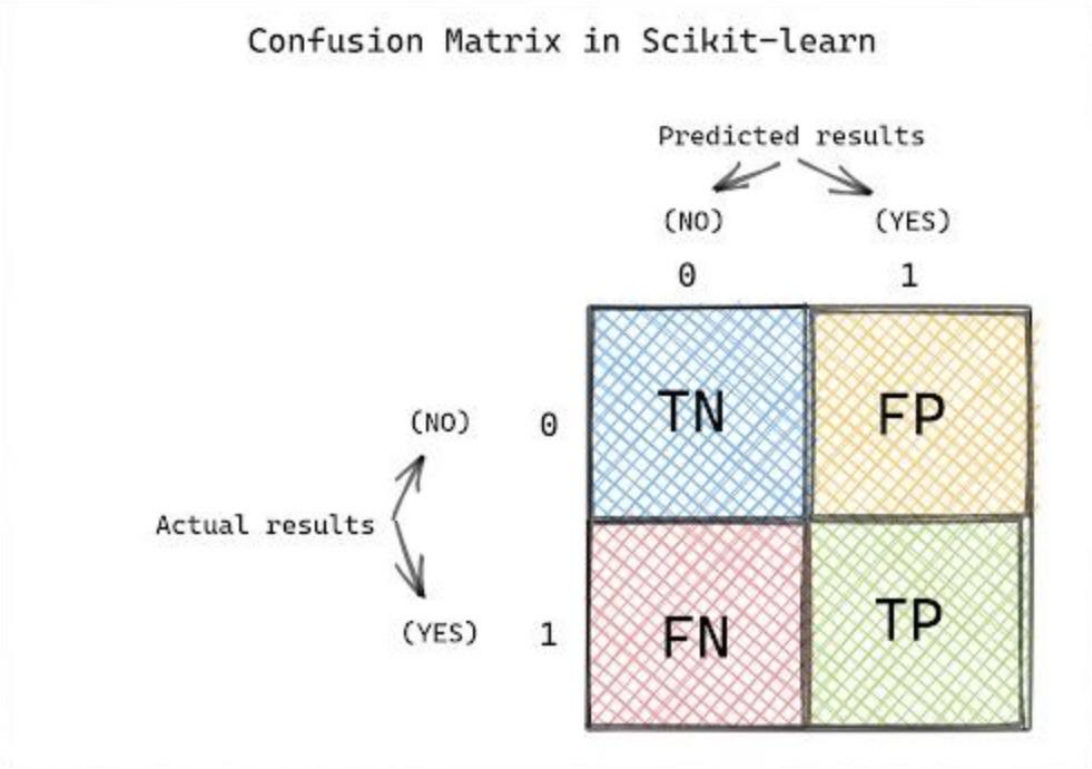

Figure 6: Reference on Sklearn's confusion matrix

- True Positives (TP): 160 - The number of counties correctly predicted as 'Low' life expectancy.
- True Negatives (TN): 199 - The number of counties correctly predicted as 'High' life expectancy.
- False Positives (FP): 11 - The number of counties incorrectly predicted as 'Low' life expectancy when they actually have 'High' life expectancy.
- False Negatives (FN): 16 - The number of counties incorrectly predicted as 'High' life expectancy when they actually have 'Low' life expectancy.

This K-Nearest Neighbors (KNN) model has delivered promising results when evaluated on the test dataset, with an overall accuracy of approximately 93%. This accuracy indicates that our model correctly predicted the life expectancy category for a substantial portion of the counties in the test set.

- Precision (Positive Predictive Value): The precision score of about 92.56% signifies that more than 92% of predicted ‘Low’ life expectancy values are actually ‘Low’ in reality.
- Recall (True Positive Rate or Sensitivity): With a recall score of around 94.76%, our model excels at correctly identifying counties with 'High' or 'Low' life expectancy. It means that nearly 94.76% of all the 'Low' life expectancy counties in the dataset are truly predicted as ‘Low’.
- F1-score (F1-Measure): The F1-score, at approximately 93.65%, is a balanced measure that considers both precision and recall. It highlights the model's ability to strike a balance between minimizing false positives (falso “Low”) and false negatives (false “High”). A high F1 score indicates strong overall model performance.

References

County Health Rankings Organization. County Health Rankings 2023 Data Dictionary. https://www.countyhealthrankings.org/sites/default/files/media/document/2023%20Data%20Dictionary%20%28PDF%29.pdf

Suraj G. (July 17, 2023). An Introduction to the Confusion Matrix in Python. Built In. https://builtin.com/data-science/confusion-matrix-python.

ChatGPT. Fix bugs and check grammar.

ChatGPT. Check the feature importance of variables in KNN models?