**A Comparison of Machine Learning Algorithms to Predict Social Determinants of Health**

## Abstract

The circumstances in the places where individuals are born, live, study, work, play, worship, and age impact a wide range of health, functional, and quality-of-life outcomes and hazards. In this research, we employ data mining, machine learning (ML) techniques, and neural network (NN) methodologies to predict social determinants of health. For this research, we used the CITI Program dataset from the NEIU Machine Learning Repository. The collection includes information on 673 patients, their none unique class features, and **BARRIERS** target attributes. Five alternative machine learning methods were examined to construct prediction models for our categorization utilizing SDH data. The results demonstrated that the ideal machine learning methodology, as measured by predictive precision, recall, and F-measure score, differed depending on the classification algorithm, implying that a method for creating predictive models from SDH data is effective. The best models for each categorization and label obtained prediction accuracies of 30–90%, showing that the methodology has the potential to supplement conventional methods for categorizing Social Determinants of Health. We developed the NN model with a different hidden layer with different epochs with different threshold and discovered that the NN with two hidden layers delivered the highest percent accuracy.

## Keywords

Machine learning, Data Mining, Neural Network, K-fold Cross Validation, Accuracy, Precision, Recall, F-Measure, social determinants of health (SDH)

## 1. Introduction

The non-medical elements that influence health outcomes are known as social determinants of health (SDH). They are the circumstances in which people are born, grow, work, live, and age, as well as the larger set of factors and institutions that shape daily life conditions. Economic policies and systems, development objectives, social norms, social policies, and political systems are examples of these forces and systems.

The SDH have a significant impact on health disparities, which are inequitable and preventable variations in health status found within and between nations. Health and sickness follow a social gradient in nations of all income levels: the lower the socioeconomic position, the poorer the health.

Data mining and machine learning have been emerging, dependable, and supportive technologies in the medical arena. The data mining approach is used to preprocess and pick important characteristics from our data, while the machine learning method aids in the automation of SDH prediction.

Data mining and machine learning algorithms can help identify the hidden pattern of data using the cutting-edge method; hence, a reliable accuracy decision is possible. Data Mining is a process where several techniques are involved, including machine learning, statistics, and database system to discover a pattern from the massive amount of dataset [1]. According to Nvidia: Machine learning uses various algorithms to learn from the parsed data and make predictions [2].

## 2. Methods

### 2.1. Data, feature, and software tool

All of the patients in the CITI program dataset are at least 19 years old. There are 673 patients in the dataset, each with their own set of nine unique features. The properties of this dataset are described in Table 1. Nine factors used to predict SDH are Age Language, Birth, Zip Code, Nearest cancer center, Kilometers to nearest cancer center, Duration to nearest cancer center, and Outcomes. The Seven class properties are treated as independent/feature variables, whereas the 'outcomes' attribute is treated as a dependent or target variable. The 'outcomes' SDH target attribute is made up of 93 labels, each of which is a binary value with a value of 0 indicating no label and 1 indicating label. We employed data mining and machine learning techniques in our study to predict whether or not a patient had a label. For the performance analysis of the SDH dataset, we employed the Python programming language and data mining technique with python. Data preparation, clustering, classification, visualization, and feature selection are all available in python. All including Neural Network is coded in the Python programming language and implemented in the Visual Studio integrated development environment.

Table 1. The attributes of SDH dataset.

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Type |
| PDAGE | Age (years). | NUMERIC |
| PDLANG | Main Language spoken | STRING |
| PDBIRTH | Country of Origin/Heritage | STRING |
| PDZIP | Current zip code of the person | STRING |
| NEAREST CANCER CENTER | Nearest cancer center | STRING |
| KM\_ TO\_NEAREST CANCER CENTER | Distance to Nearest cancer center | NUMERIC |
| DURATION\_ TO\_NEAREST CANCER CENTER | Duration to Nearest cancer center | NUMERIC |

### 2.2. Data preprocessing

Preprocessing aids in the transformation of data so that a more accurate machine learning model may be generated. To enhance data quality, preprocessing performs a variety of activities such as, filling missing values, data normalization/standardization, and feature selection. The collection contains 673 samples

### 2.2.1 Missing value identification

We found the missing values in the datasets using Excel and the Weka tool, as shown in Table 2. The missing value was replaced with the appropriate mean value.

[Table 2]. The number of missing values in SDH dataset.

|  |  |
| --- | --- |
| Attribute | No of Missing values |
| PDAGE | 1 |
| PDLANG | 1 |
| PDBIRTH | 2 |
| PDZIP | 6 |
| NEAREST CANCER CENTER | 6 |
| KM\_ TO\_NEAREST CANCER CENTER | 6 |
| DURATION\_ TO\_NEAREST CANCER CENTER | 6 |

### 2.2.2 Feature Selection

Pearson's correlation approach is a well-known method for determining the most important attributes/features. This approach calculates the correlation coefficient, which is related to the output and input qualities. The value of the coefficient remains between 1 and 1. A significant correlation is shown by a value above or below 0.5, whereas no correlation is indicated by a value of zero. [Table 3] shows the results of using the correlation filter to find the correlation coefficient. For relevant qualities, we chose a cut-off of 0.2. As a result, the five most important input attributes are AGE, PDLANG, BIRTH, ZIP, Nearest hospital, Nearest cancer center, Km to nearest cancer center, Duration to nearest cancer center and Nearest cancer center zip.

Table 3. The correlation between input and output attributes.

|  |  |
| --- | --- |
| Attribute | Correlation coefficient |
| PDAGE | 0.394993 |
| PDLANG | 0.111318 |
| PDBIRTH | 0.090272 |
| PDZIP | 0.081835 |
| NEAREST CANCER CENTER | 0.059098 |
| KM\_ TO\_NEAREST CANCER CENTER | 0.054201 |
| DURATION\_ TO\_NEAREST CANCER CENTER | 0.059211 |

### 2.2.3 Normalization

We did feature scaling by Normalization the data to have a data from 0 and 1 range, which increased the calculation performance of the method. We have 666 samples/instances after preprocessing and handling missing and none values.

2.3. Dataset train and test method/ Evaluation Strategy

The dataset is ready to train and test after it has been cleaned and preprocessed. To test the performance of the multiple machine learning models, we utilized K-fold cross-validation and the 80 percent /20 percent train/test splitting approach individually. The train/split approach divides the dataset into training and testing sets at random. The data is separated into K folds in the K cross-validation approach. Validation/testing is done with one fold, and training is done with the remaining K-1 folds. The technique will be repeated until each and every K fold is a test set. The average of all recorded Kth test scores is used to assess performance.

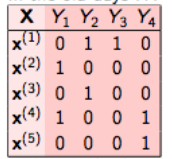
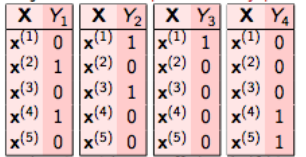
2.4. Design and implementation of classification model

In this study, different ML classification techniques such as BinaryRelevance, ClassifierChain, OneVsRestClassifier, and neural network are used to investigate the SDH (CNN). For the CNN algorithm, we utilized Kth value = 10. Figure 4 depicts the proposed model diagram.

### 2.4.1 BinaryRelevance model implementation

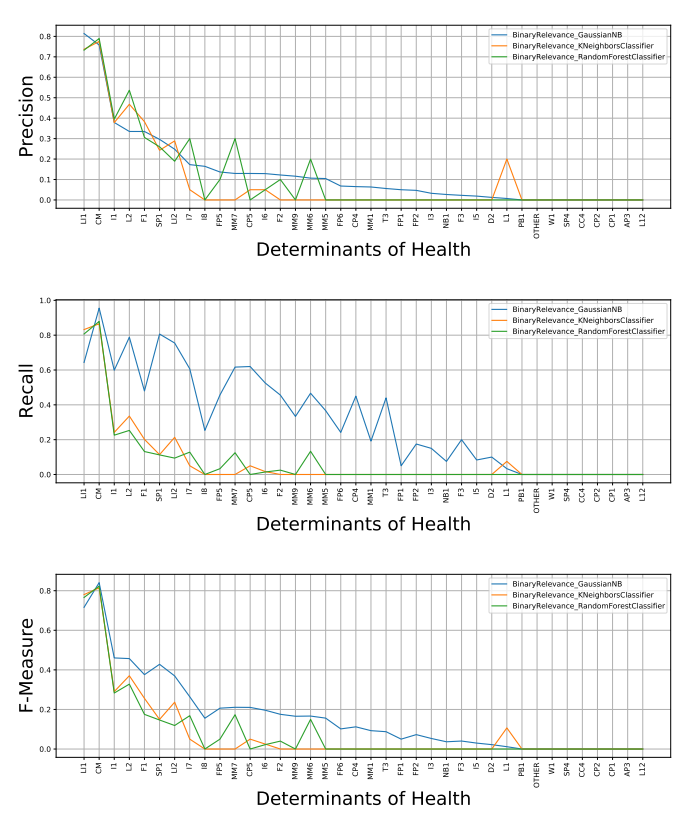
This is the most basic strategy, in which each label is treated as a distinct single-class classification issue. Let's look at an example, as given below. We have the following data set, where X is the independent characteristic and Y is the target variable.

This problem is divided into four separate single class classification problems in binary relevance.

We deploy three binary relevance model with three different classifications. We implemented the BinaryRelevance with GaussianNB, KNeighborsClassifier, and RandomForest. and the results are compared. [Fig 1]

Fig 1 Binary Relevance Comaprisons



Download : Download PDF with more info (177KB)

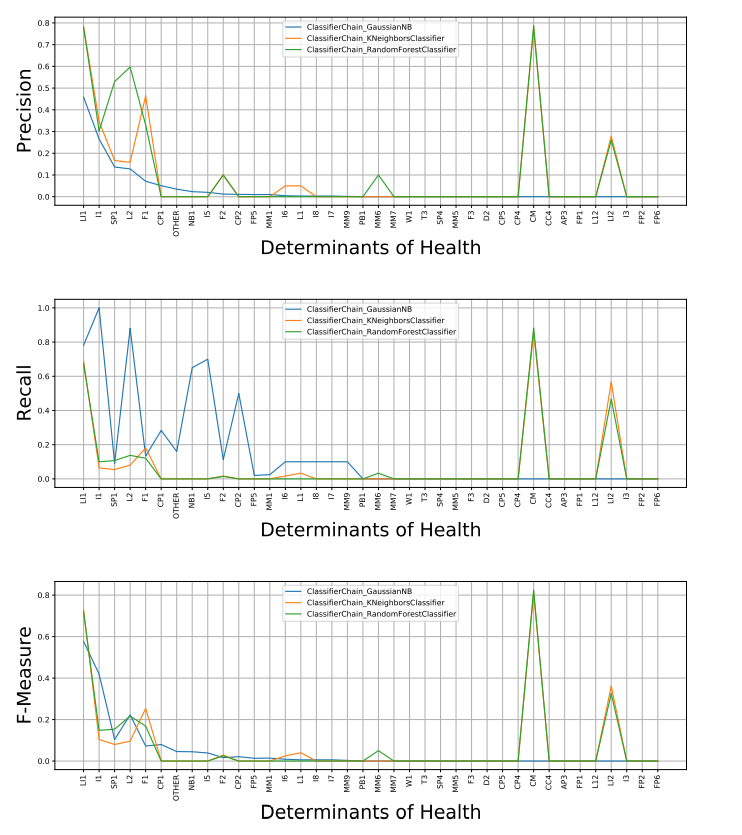
2.4.2 ClassifierChain model implementation

The initial classifier is trained only on the input data, and each subsequent classifier is trained on the input space as well as all previous classifiers in the chain.



We deploy three binary relevance model with three different classifications. We implemented the ClassifierChain with GaussianNB, KNeighborsClassifier, and RandomForest. and the results are compared. [Fig2]

Fig 2 ClassifierChain Comaprisons



Download : Download PDF with more info (177KB)

2.4.3 OneVsRestClassifier model implementation

This technique, often known as one-vs-all, consists of fitting one classifier per class. The class is fitted against all the other classes for each classifier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

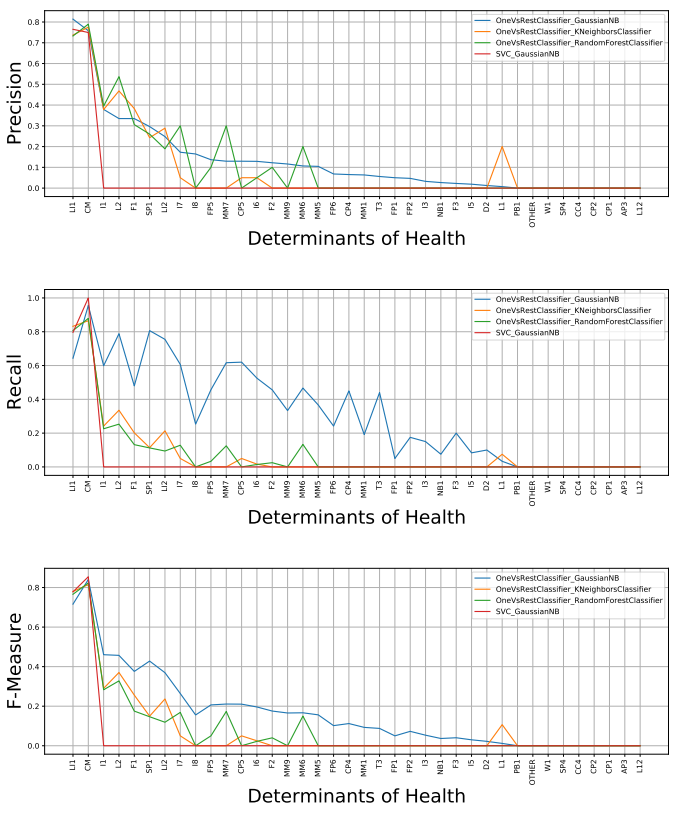
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y1 | Y2 | Y3 | Y4 |
| x1 | 0 | 1 | 1 | 1 |
| x2 | 0 | 0 | 1 | 1 |
| x3 | 1 | 0 | 1 | 0 |

We deploy three binary relevance model with three different classifications. We implemented the OneVsRestClassifier with GaussianNB, KNeighborsClassifier, SVC, and RandomForest. and the results are compared. [Fig3]

Fig 3 OneVsRestClassifier Comaprisons



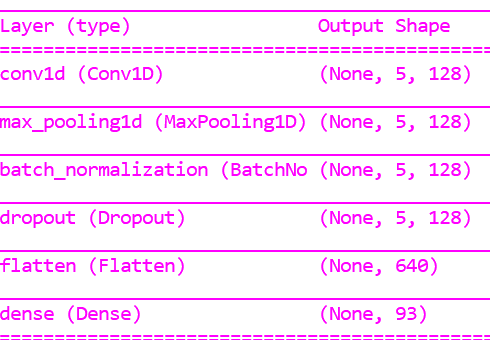
Download : Download PDF with more info (177KB)

### 2.4.4 Neural network model implementation

We created two separate neural network models with different hidden layer depths. The neural network was implemented with hidden layers 1 and 2, 5 different thresholds and 512 epochs, and the results were compared. The activation function in the hidden layer of CNN processes the weighted sum of input. In our research, we used sigmoid and RELU activation functions. The neural network models were created using the Keras and TensorFlow libraries. A Sequential class from the Keras library was utilized. The 'Outcomes' attribute contains the target variables. During the backpropagation procedure of CNN, the optimizer is necessary to reduce the output error. As an optimizer, we used rmsprop (Root Mean Square Propagation). The learning rate is a parameter in an optimization algorithm that controls the weight adjustment with respect to loss gradient. We used different learning rates to find an effective one. From the scikit-learn library, we used the train\_test\_split function to perform the train/test splitting task. We also used the K-Fold cross\_val\_score function from the scikit\_learn library for the K-fold cross-validation task.

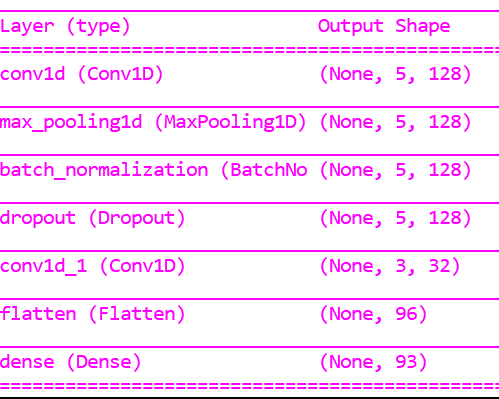
### 2.4.4.1 Developing a NN model with one hidden layer

We began by creating a neural network that included one hidden layer in addition to the input and output layers. As there are seven features, we designated the input layer as having seven neurons. Max pooling, batch normalization, a dropout layer, and a RELU activation function are all included in the hidden layer. There are 93 neurons in the output layer, with a sigmoid activation function. In Fig. 5, the model summary of NN with one hidden layer is shown.



### 2.4.4.2 Developing a NN model with two hidden layers

We've created a NN model with one hidden layer that has the same input shape, neurons, and activation function as NN. The second layer has a hidden layer with 26 neurons and an output layer with 93 neurons with a sigmoid activation function, similar to the NN with one layer. In Fig. 6, the model summary of NN with two hidden layers is shown.

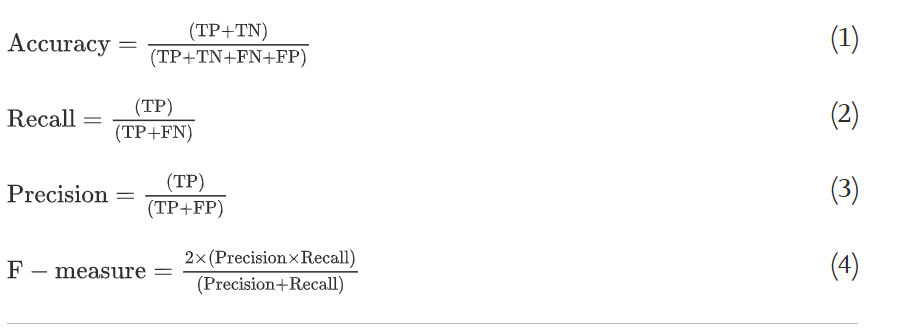


## 3. Result and discussion

### 3.1. Results for ML method Binary Relevance, ClassifierChain, OneVsRest

The confusion matrix can be used to calculate the machine learning algorithm's accuracy. The confusion matrix is shown here in its abstract form.

| **Predicted No (0)** | **Predicted Yes (1)** |
| --- | --- |
| **Actual No (0)** | TN | FP |
| **Actual Yes (1)** | FN | TP |



Here, FP False Positive, FN False Negative, TN True Negative, and TP True Positive. Eqs. (1), (2), (3), (4) are used to calculate the performance measurement of the classification method.

Table 6. The performance measure of all classification methods​ for K-fold cross-validation and Train/Test splitting method.

| **Classification** | **Precision** | **Recall** | **F-measure** | **Accuracy** |
| --- | --- | --- | --- | --- |
| BR(GaussianNB)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| BR((KNeighborsClassifier)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| BR(RandomForestClassifier(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| BR(GaussianNB) (Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| BR(KNeighborsClassifier)(Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| BR(RandomForestClassifier)(Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| CC(GaussianNB)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| CC((KNeighborsClassifier)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| CC(RandomForestClassifier(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| CC(GaussianNB) (Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| CC(KNeighborsClassifier)(Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| CC(RandomForestClassifier)(Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR(GaussianNB)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR((KNeighborsClassifier)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR(RandomForestClassifier(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR (SVC) (K-Flod) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR(GaussianNB)(Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR(KNeighborsClassifier)(Splitting) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR(RandomForestClassifier)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
| OVR((SVC)(K-fold) | 0.000 | 0.000 | 0.000 | 0.000 |
|  |  |  |  |  |

### 3.2. Results for neural network

| **Hidden layer** | **Threshold** | **Precision** | **Recall** | **F-measure** | **Accuracy** |
| --- | --- | --- | --- | --- | --- |
| 1 | 0.3 | 0.000 | 0.000 | 0.000 | 0.000 |
|  | 0.4 | 0.000 | 0.000 | 0.000 | 0.000 |
|  | 0.5 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.3 | 0.000 | 0.000 | 0.000 | 0.000 |
|  | 0.4 | 0.000 | 0.000 | 0.000 | 0.000 |
|  | 0.5 | 0.000 | 0.000 | 0.000 | 0.000 |

## Conclusion

According to research, social determinants of health can have a greater impact on health than health treatment or lifestyle choices. Numerous research imply that SDH is responsible for 30-55 percent of health outcomes. Furthermore, estimations reveal that industries other than health contribute more to population health outcomes than the health sector.

Addressing SDH appropriately is fundamental for improving health and reducing longstanding inequities in health, which requires action by all sectors and civil society.

Utilizing Python, we preprocessed the data. In the CITI Program dataset, we employed seven input features (PDBIRTH, PDLANG, PDZIP, Km to nearest cancer center, Duration to nearest cancer center, Nearest cancer center zip, and Nearest cancer center zip) and 93 output features (outcomes). On the CITY Program dataset, we employed multiple machine learning methods to predict SDH, including BinaryRelevance(GaussianNB), BinaryRelevance((KNeighborsClassifier), BinaryRelevance((RandomForestClassifier), ClassifierChain(GaussianNB), ClassifierChain(KNeighborsClassifier), ClassifierChain(RandomForestClassifier), OneVsRestClassifier(GaussianNB), OneVsRestClassifier(KNeighborsClassifier), OneVsRestClassifier(RandomForestClassifier), and OneVsRestClassifier(SVC), and assessed the performance on various measures. or some metrics, such as accuracy, precision, recall, and F-measure, all models produce positive outcomes. We also used the NN model to predict SDH in the CITY Program dataset. With varied thresholds and 512 epochs, we employed the 1, 2 hidden layers in the neural network model, which has the highest accuracy among our constructed models for SDH. The NN with two hidden layers is the most efficient and promising of all the presented models for evaluating SDH, with an accuracy rate of around 86 percent for all varied thresholds (0.3, 0.4, 0.5).

The current work shows that using CITI Program profiles and machine learning algorithms, prediction models for social determinants of health like housing, insurance, and disability may be created.

## References

Techniques for Solving a Multi-Label Classification Problem

Problem Transformation Method

Adapted Algorithm Method

Ensemble Approaches