Proceedings of Cloud based International Conference ,
"Computational Systems for Health and Sustainability(CSFHS)"
5th June, 2020 Organized by sbytetechnologies.com

Prediction Of Heart Disease And Diabetes Using Machine Learning

AKHILA MARAM
Final Year Student, B.Tech (CSE), BNMIT, Bangalore

MAHALAKSHMI N
Final Year Student, B.Tech (CSE), BNMIT, Bangalore

NIRIKSHA N
Final Year Student, B.Tech (CSE), BNMIT, Bangalore

Abstract: **Healthcare technology is changing. The use of algorithms for increasingly important tasks is spreading across the healthcare sector. A new generation of machine learning algorithms that promise to inform diagnosis and assist in treatment are emerging. Machine Learning can play an essential role in predicting presence/absence of chronic disorders, heart diseases and more. Such information, if predicted well in advance, can provide important insights to doctors who can then adapt their diagnosis and treatment per patient basis. This paper deals with designing, developing and implementing prediction systems for diabetes and heart disease using machine learning algorithms.**

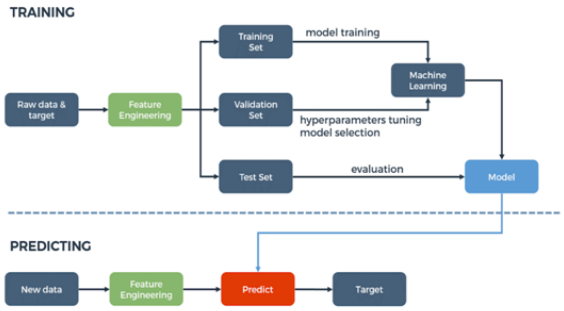
Key words : **Healthcare Technology; Algorithms; Machine Learning; Chronic Disorders;**

INTRODUCTION

The importance of ML in every sector is growing every day, hence more people are investing time to learn it. Machine learning has applications in all types of industries, including manufacturing, retail, healthcare and life sciences, travel and hospitality, financial services, energy, feedstock, and utilities. Amongst these applications, one of the most important and significant fields of concern is the healthcare sector. Machine learning has transformed healthcare. It's being used to diagnose lung cancer, pneumonia, heart disease, diabetes, hypertension, and other diseases. Machine learning is more accurate and faster at diagnosis than real doctors. Machine learning builds a model that makes predictions based on evidence in the presence of uncertainty. A machine learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data.

Methodologies

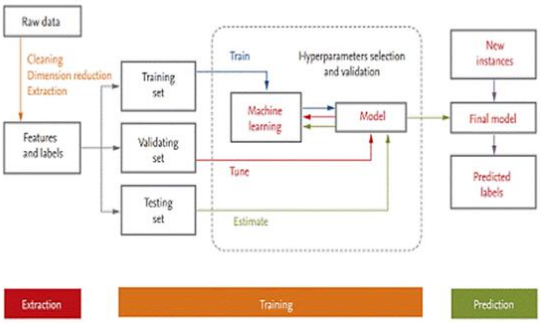
There is a need for combinational and more complex models to increase the accuracy of predicting the early onset of heart disease and diabetes. With the more amount of data being fed into the database the system will be very intelligent. The proposed system is intended to design an efficient prediction model to predict the onset of diabetes and heart disease in humans with complex models that deal with algorithms which produce more accurate results. The proposed system takes the datasets of diabetes and heart disease as input in the form of csv files. The dataset is then subjected to data pre-processing, a crucial step in the process. This includes data selection, data cleaning and feature engineering. The pre-processed data obtained is then used for statistical analysis using various visualization tools to understand the features and relations between these features in a comprehensive manner. The dataset is then split into the training and testing set as per the requirements at that stage in the process. The training data is used in the learning phase where several iterations are required. The prediction model is then developed based on the learning phase. The performance of the model is obtained using the test data. Once the system is put in use the operation of the system will be judged. System architecture contains the details regarding the plan for implementing the non-functional requirements.In machine learning, a common task is the study and construction of algorithms that can learn from and make predictions on data. Such algorithms work by making data-driven predictions or decisions, through building a mathematical model from input data. The model is initially fit on a training dataset that is a set of examples used to fit the parameters of the model. Successively, the fitted model is used to predict the responses for the observations in a second dataset called the validation dataset. The validation dataset provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyperparameters. Finally, the test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset.



Block diagram of the system model

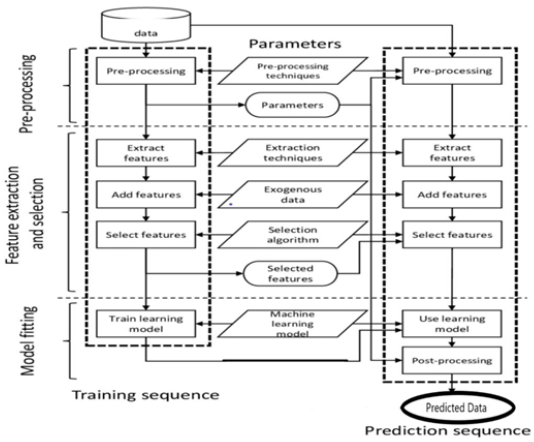
The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. Through iterative optimization of an objective function, machine learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Therefore, in our work we have aimed at establishing a technique which takes in the selected symptoms as input and based on the evaluations, it predicts whether the patient has diabetes and heart disease. It is implemented by constructing a Fully connected neural network model over diabetes and heart disease dataset machine learning repository. We implemented the Fully connected model by generating training and testing samples and have also included several optimization strategies to improve the accuracy of the model.

Proposed System



Diabetes and Heart disease prediction system

The diabetes and heart disease dataset is used to train the model and classify every data record into either of these features: disease detected or disease not detected (positive or negative). Upon sufficient training, the model would be able to classify the testing set of data records into the specific divisions. The output will be displayed after analysing the entire data set provided. A GUI is provided that will take user input and then perform analysis against the training data and the test data. This system with the above said features would help the user of the system or service to understand the symptoms of the disease at a faster rate and take suitable actions through the trained model. The system is designed to perform the following tasks: To analyse the diabetes and heart disease dataset using machine learning techniques, Obtain the data inputs from the user for detection of the disease, Test the user data against the training and test data and Estimate the accuracy of the prediction analysis. The process of training the ML model includes the task of providing the learning algorithm with the training data from which it can learn. The data set having the data records of the diseases is provided for this purpose. The training data must contain the correct answer which is nothing but target or target attribute. The learning algorithm finds matches in the training data that maps the input data attributes to the target value. The field of Machine Learning provides many algorithms for prediction. Multi-layer perceptron is one such algorithm which is used in this system for predicting the results of disease detection. Though this algorithm looks simple, it outperforms many other algorithms used in prediction analysis. After using this system people tend to reduce their manual efforts of data classification analysis. This system proves to be very useful to the health sector and the clinical experts of the medical field. The application mainly strives hard to give the best to the medical data users. The automated classification of the pretrained model will provide the fastest analysis of the diabetes and heart disease data records. The classifier can be trained to identify the presence of the disease. Since the whole process is automated the output can be obtained in a very short time.



Dataflow Diagram

The core concept used in implementing the prediction system of diabetes and heart disease is the Multi-layer perceptron neural network. The

implementation of this concept requires the following steps: Classification of the dataset is done with the input split ratio, i.e, the data corpus is divided into train data and test data. The train data is to train the ML model or in other words, get it accustomed to the type of data that it has to classify. The test data is the part of data that the ML model has to classify from its learning from train data. When the train data is fed to the ML model, feature extraction takes place. In other words, the ML model tries to figure out which features of the dataset are leading to the corresponding target output, also called the learning process. The better the learning process, the more varied test data it can classify. Now, when the test data is input to the ML model, feature extraction takes place. Each data record is reviewed for different features. The probability of detection of diabetes and heart disease is determined. The probability with the highest value is displayed as the output of the prediction analysis. This process is carried out for each data record in the test data corpus. The accuracy of the output obtained is known by comparing the actual output with the classifier output and the percentage accuracy is displayed. From the number of positive and negative target values present in the output, the prediction is made successfully on the dataset of diabetes and heart disease. Tensor Flow computations are expressed as stateful dataflow graphs. It helps in executing Machine Learning algorithms in a seamless way.

System Design

Datasets

Diabetes-

The data set has 6795 observations and 20 variables. It has been obtained from the Department of Biostatistics, Vanderbilt University.

Name	**Labels**	**Units**	**Levels**
Seqn	Respondent sequence number		
Sex	Sex		2
Age	Age	years	
Re	Race/Ethnicity		5
Income	Family Income		14
Tx	On Insulin or Diabetes Meds		
Dx	Diagnosed with DM or Pre-DM		
Wt	Weight	Kg	
Ht	Standing Height	Cm	
Bmi	Body Mass Index	kg/m^2	
Leg	Upper Leg Length	Cm	
Arml	Upper Arm Length	Cm	
Armc	Arm Circumference	Cm	
Waist	Waist Circumference	Cm	
Tri	Triceps Skinfold	Mm	
Sub	Subscapular Skinfold	Mm	
Gh	Glycohemoglobin	%	
albumin	Albumin	g/dL	
Bun	Blood urea nitrogen	mg/dL	
SCr	Creatinine	mg/dL	

Diabetes Dataset

Heart Disease-

The data set has 1300 observations and 14 variables. It has been obtained from UCI data repository.

Name	**Labels**	**Units**	**Levels**
Age	Age	years	
Sex	Sex		2
Cp			
trestbps	Trest pain		4
Chol	Serum Cholesterol	mg/dl	
Fbs	Fasting Blood Sugar	mg/dl	
Restecg	Resting ECG		
Thalach	Maximum Heart Rate Achieved		
Exang	Exercise Induced Angina		
oldpeak	ST Induced by exercise		
Slope	Peak exercise ST		
Ca	Cardiac Arrest		
Thal	Thalassemia		
Target	Target		

Heart Disease Dataset

Algorithms Used:

Multi-Layer Perceptron

A multilayer perceptron (MLP) is a feed forward artificial neural network that generates a set of outputs from a set of inputs. An MLP is characterized by several layers of input nodes connected as a directed graph between the input and output layers. MLP is a deep learning method which utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron.

Layers

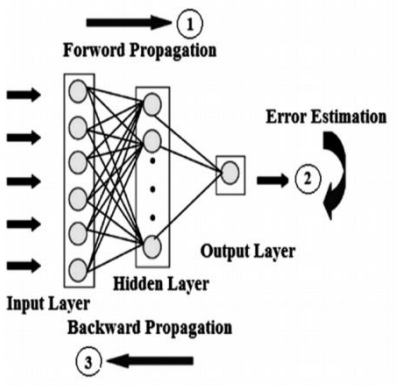
The MLP consists of three or more layers of nonlinearly-activating nodes as follows: Input Layer (one), Output layer (one) and Hidden layers (one or more). ssMLPs are fully connected with each node in one layer connecting a certain weight to every node in the following layer.

Working

Multilayer perceptron's train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Backpropagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE). Feedforward networks such as MLPs are just like ping-pong: they are mainly involved in two motions, a constant back and forth (forward and backward passes):

- In the forward pass, the signal flow moves from the input layer through the hidden layers to the output layer, and the decision of the output layer is measured against the ground truth labels;
- In the backward pass, using backpropagation and the chain rule of calculus, partial derivatives of the error function regarding the various weights and biases are back-propagated through the MLP. That act of differentiation gives us a gradient, or a landscape of error, along which the parameters may be adjusted as they move the MLP one step closer to the error minimum. (this can be done with any gradient-based optimization algorithm such as stochastic gradient descent).

The network keeps playing that game of ping-pong until the error can go no lower. This state is known as convergence.



Working of MLP Neural network

Learning

Learning occurs in the perceptron by changing connection weights after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and is carried out through backpropagation, a generalization of the least mean squares algorithm in the linear perceptron.

We can represent the degree of error in an output node j in the nth data point (training example) by $e_j(n) = d_j(n) - y_j(n)$ where d is the target value and y is the value produced by the perceptron. The node weights can then be adjusted based on corrections that minimize the error in the entire output, given by

$$\mathcal{E}(n) = \frac{1}{2}\sum_j e_j^2(n)$$

Using gradient descent, the change in each weight is

$$\Delta w_{ji}(n) = -\eta \frac{\partial \mathcal{E}(n)}{\partial v_j(n)} y_i(n)$$

Where y_i the output of the previous neuron and η is the learning rate, which is selected to ensure that the weights quickly converge to a response, without oscillations.

The derivative is calculated based on the induced local field v_j, which itself varies. It is easy to prove that for an output node this derivative can be simplified to

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = e_j(n)\phi'(v_j(n))$$

Where ϕ' is the derivative of the activation function described above, which itself

does not vary. The analysis is more difficult for the change in weights to a hidden node, but it can be shown that the relevant derivative is

$$-\frac{\partial \mathcal{E}(n)}{\partial v_j(n)} = \phi'(v_j(n)) \sum_k -\frac{\partial \mathcal{E}(n)}{\partial v_k(n)} w_{kj}(n)$$

This depends on the change in weights of output layer. So to change the hidden layer weights, the output layer weights change according to the derivative of the activation function, and so this algorithm represents a backpropagation of the activation function. The two historically common activation functions are both sigmoids, and are described by,

$$y(v_i) = \tanh(v_i) \text{ and } y(v_i) = (1 + e^{-v_i})^{-1}$$

In recent developments of deep learning the rectifier linear unit (ReLU) is more frequently used as one of the possible ways to overcome the numerical problems related to the sigmoids. The first is a hyperbolic tangent that ranges from -1 to 1, while the other is the logistic function, which is similar in shape but ranges from 0 to 1. Here is the output of the *i*th node (neuron) and v_i, is the weighted sum of the input connections. Alternative activation functions have been proposed, including the rectifier and soft plus functions.

Backpropagation algorithm

```
1: procedure TRAIN
2:     X ← Training Data Set of size mxn
3:     y ← Labels for records in X
4:     w ← The weights for respective layers
5:     l ← The number of layers in the neural network, 1...L
6:     D_ij^(l) ← The error for all l,i,j
7:     t_ij^(l) ← 0. For all l,i,j
8:     For  i = 1 to m
9:         a^l ← feedforward(x^(i), w)
10:        d^l ← a(L) − y(i)
11:        t_ij^(l) ← t_ij^(l) + a_j^(l) · t_i^(l+1)
12:    if j ≠ 0 then
13:        D_ij^(l) ← (1/m) t_ij^(l) + λ w_ij^(l)
14:    else
15:        D_ij^(l) ← (1/m) t_ij^(l)
16:        where ∂/∂w_ij^(l) J(w) = D_ij^(l)
```

Gradient Descent algorithm

Input:
Training sample $S = \{(\vec{x}, y)\}^m$, where $(\vec{x}, y) \in X \times \{r_q \succ r_{q-1} \succ \cdots \succ r_1\}$
Learning rate $\eta > 0$, cost parameters $\{\tau_{k(i)}\}$ and $\{\mu_{q(i)}\}$, penalty weight λ

Make $S' = \{((\vec{x}_i^{(1)}, \vec{x}_i^{(2)}), z_i)\}_{i=1}^{\ell}$ from S;
$\vec{w} = \vec{0}$;
while (stop_condition isn't met){
  $\Delta\vec{w} = \vec{0}$;
  for($i = 0$; $i < \ell$; $i++$){
    if ($z_i \langle \vec{w}, \vec{x}_i^{(1)} - \vec{x}_i^{(2)} \rangle < 1$) $\Delta\vec{w} = \Delta\vec{w} + z_i \tau_{k(i)} \mu_{q(i)} (\vec{x}_i^{(1)} - \vec{x}_i^{(2)})$;
  }
  $\Delta\vec{w} = \Delta\vec{w} - 2\lambda w$;
  $\vec{w} = \vec{w} + \eta\Delta\vec{w}$;
}
return $\vec{w}$;

Working of the system

Heart Disease

1) **Pre-Processing:** The heart disease raw data of patients from four hospitals was obtained from UCI database. The obtained data was reformatted into CSV to read it as data frame. The dataset from four hospitals was combined into one dataset and it was shuffled to improve the generalization. Some of missing values was filled by finding the relationship between different features like:

- A person was assumed as smoker if the data is present for no of cigarettes smoked in a day or the number of years since he has been smoking.
- The pain location was assumed to be substernal if the patient has a heart disease and the pain location was assumed to be not substernal when the patient is diabetic even if the patient has a heart disease.
- The patient with resting blood pressure more than 120 was assumed to have hypertension condition.
- The y_i exercise angina was assumed to be present if the patient has pain in substernal area. The remaining missing values of continues data was filled with mean and categorical data with -1.
- Pre-processing of Data: Some of the features was pre-processed to make it suitable for training like:
- The continues cholesterol data was converted into categorical by assigning 0 if the patient has less than or equal to 200 mg/dL (desirable), 1 if it is between 200 mg/dL and 239mg/dL (borderline high) and 2 if it is above 239mg/dL (high)
- The resting ECG was simplified from 0, 1, 2 (normal, ST abnormality, left ventricular hypertrophy) to 0, 1 (normal, abnormal).
- The slope produced by ST depression was changed 1, 2, 3 representing 1 as high slope, 2 has flat or normal and 3 as down slope to 1, 0, 2 respectively.
- The heart wall damage values 1 to 7 was converted into 0, 1, 2. The values less than 3 being no damages was converted into 0, the values between 3 and 6 being irreparable damage to 2 and the value 7 being reparable damage to 1.
- The output of some of the hospitals had values from 0 to 4 representing the severity of the disease, which was simplified into 0, 1 (no disease, disease) to make it consistent. All the categorical values except the output was converted into one hot encoder.

2) **Preparing data for training:** The pre-processed data was normalized into values between 0 and 1 using Min Max Scaler to bring all the data into same scale. Then the data was split into training and testing data, 30% of the data was used for testing the accuracy of the model. Of the 70% data selected for training, 10% was used for validation of the model.

3) **Training of the Model:**

Building the model: Sequential model using fully connected layers was used for prediction of heart disease. The fully connected layer consists of fully connected neurons, meaning each neuron is dependent on all the output of the previous layer. The choice of fully connected neural network was obvious as the patient data is independent of one another and it is not an image dataset.

Tuning of the model: Initially due to small amount of data the model was overfitting, so we used generalization techniques to improve the model. We used L1 regularizers and dropout layers to increase the generalization of the model. The L1 regularizers or lasso regularizers work by adding a squared magnitude penalty term to the loss function, which helps to avoid the overfitting of the model and it is resistant to outliers. The dropout layers help in improving the generalization by dropping a fixed percentage of neurons in every epoch, which creates an effect of training models with different architecture. The number of neurons was also reduced to reduce the capacity of the model, which in turn increases the generalization of the model. Even after extensive parameter tuning the maximum accuracy achieved was around 91.2%.

Diabetes

1) Pre-Processing: The diabetes raw data of patients from hospitals was obtained from the Biostatistics Department, Vanderbilt University. The obtained data was reformatted into CSV to read it as data frame. The number of attributes were reduced to 9 from 21 after Exploratory data analysis. The data set had around 6795 records.

2) Preparing data for training: The pre-processed data was normalized into values between 0 and 1 using Min Max Scaler to bring all the data into the same scale. Then the data was split into training and testing data, 35% of the data was used for testing the accuracy of the model. Of the 65% data selected for training, 10% was used for validation of the model.

3) Training of the Model:

Building the model: Sequential model using fully connected layers was used for prediction of Diabetes. The fully connected layer consists of fully connected neurons. The model consisted of 4 layers, one input layer, two hidden layers, and one output layer with 64, 32, 32 and 1 neurons respectively.

Tuning of the model: Initially due to the small amount of data the model was overfitting, so we used generalization techniques to improve the model. We used L1 regularizers and dropout layers to increase the generalization of the model. Draws samples from a uniform distribution within [-limit, limit] where limit is sqrt (6 / fan_in) where fan_in is the number of input units in the weight tensor. After extensive parameter tuning the maximum accuracy achieved was around 90.2%.

RESULT ANALYSIS

Diabetes

Algorithms used	**Accuracy**
Decision Tree	84%
Naïve Bayes	87.9%
Logistic Regression	88%
Multilayer perceptron neural network	90.2%

Heart Disease

Algorithms used	**Accuracy**
Naïve Bayes	75.8%
Logistic Regression	82.9%
Decision Tree	85%
Random Forest	86.1%
Support Vector Machines	86.1%
Multilayer perceptron neural network	91.27%

CONCLUSION

The system is aimed at predicting the most common chronic diseases in the Indian subcontinent Diabetes and Heart disease. The system obtains the user data through the user-friendly GUI for each disease based on the blood tests values and the other tests for determining various parameters. The values are used by Multi-Layer Perceptron to build a suitable predicting model. The MLP helps overcome the accuracy deficiency of various other models in existence such as Naïve Bayes, Logistic Regression,

etc. The fine tuning of parameters and variation of kernel initializers and optimizers has resulted in improvement of accuracy supported with increase in number of neurons per Layer of the MLP. The system fulfils the aim of accuracy enhancement from the existing system, while analysing a wide range of parameters for each disease. The system helps combat the unavailability of specialized health care personnel and provides an accurate cost-effective time efficient system to be used by the population. The system is designed to consider the prediction analysis on the diabetes and heart disease dataset to detect the presence of diabetes and heart disease for the user input. The machine learning algorithms have been used to implement the above stated. The multi-layer perceptron neural network implementation involves dividing the input dataset into training dataset and testing dataset. The trained model classifies the test data and displays it as the test dataset, the result of the presence of the disease is displayed for better understanding of the classified results. The accuracy of the results is also calculated and displayed. The Multi-layer Perceptron Neural Networks implementation takes input data from the user and displays the prediction analysis results. The categories are: Disease detected or Disease not detected. The accuracy of the results is determined by the user as there is a predefined dataset (test dataset) for calculation of accuracy.

ACKNOWLEDGEMENT

We would like to thank Mr. Prashanth J, Assistant Professor, Department of Computer Science, BNMIT and Dr. Sahana D Gowda, Head of Department, Department of Computer Science, BNMIT for their constant guidance and assistance in the project work. Our heartfelt thanks to Dr. S., Sridhar, Ex Vice Chancellor currently CEO, Sbytetechnologies.com , for organizing a cloud-based international conference and allowing us to exhibit our understandings of the work.

REFERENCES

[1] Aditi Gavhane, Isha Pandya, GouthamiKokkula, Prof. Kailas Devadkar (PhD), "Prediction of Heart Disease Using Machine Learning" in Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018) IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1-5386-0965-1, pp.1275-1278

[2] Ayman Mir, Sudhir N. Dhage, "Diabetes Disease Prediction using Machine Learning on Big Data of Healthcare in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)

[3] Fahad P K and Pallavi M S, "Prediction of Human Health using Machine Learning and Big Data" in International Conference on Communication and Signal Processing, April 3-5, 2018, India, pp.0195-0199

[4] Jatin N Bagrecha, Chaithra G S, Jeevitha S, "Diabetes Disease Prediction using Neural Network" in International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177, Volume 7 Issue IV, Apr 2019, pp.3888-3893

[5] Senthilkumar Mohan, Chandrasegar Thirumalai, and Gautam Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques" in IEEE special section on smart caching, communications, computing and cybersecurity for information-centric internet of things July 3, 2019, pp.81542-81554

[1]

