COT4501:Autonomous Machine Classification

Team: Dash

Team Members:
Akshita Kandlikar
Naga Sanketh Vysyaraju
Preetham Dasari
Rahul Rudra

Approach:

- In the code that uses this dataset, we are deleting the first row in the data as it has the names of each of the columns of the dataset and called it **X.**

- In the code we are first extracting the column in the dataset, which has the names of the classes and creating the matrix **Y**, which contains records of the type **[0, 0…0,1, 0...n elements].** The value n represents the total number of distinct classes in the dataset and each position i.e. each column represents unique class. If a **Y[i]** record has value "1" in a particular position the in it, then it represents the class of the corresponds record in **X** i.e. **X[i]**, which contains the main data of different attributes.

- Before dividing the data in to training and testing, we are shuffling the data each time the code is executed and then starting the partitioning. For each dataset, the training data is 70% of the total data, testing data is rest of the 30% data and the cross-validation dataset it 20% of the training data we partitioned initially. So, the actual training data is 56% of the total data, actual testing is 30% of the total data and the cross-validation part is 14% of the total data.

- Since we are shuffling the data each time we are executing the code, the training, validation and testing set would not be the same each time. So, the final accuracy (Which is defined in the output section) of the code that we are printing out would not be the same every time. There will be a difference in percentage but it won't be very significant.

We have built a basic linear regression which is just a very basic model and there are many other methods like Support Vector Machine(SVM) and Artificial Neural Networks(ANN) which perform much better.

SVM: An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

ANN: **Artificial neural networks** (**ANN**) or **connectionist systems** are computing systems vaguely inspired by the biological neural networks that constitute animal brains The neural network itself is not an algorithm, but rather a framework for many different Machine Learning algorithms to work together and process complex data inputs. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules.

Along with the SVM algorithm, which is used through "sklearn", we are also comparing our program with ANN algorithm, which is used through "sklearn.neural_network".

Results

Output: We are printing the accuracy of our code on this dataset by cross checking the output generated by the code with the actual output we know as part of the testing dataset. The accuracy is defined as: "**The number of true positives achieved in this testing dataset divided by the total number of records in the testing dataset**".

We are also constructing the **Confusion Matrix** for every code along with the code for SVM and ANN.

In **Confusion Matrix** the rows represent the actual classes and the columns represent the predicted classes. So, all the false positives and true positives are listed in it.

There are two strategies for reducing the problem of multiclass classification to multiple binary classification problems. It can be categorized into One vs All and One vs One. The techniques developed based on reducing the multi-class problem into multiple binary problems can also be called problem transformation techniques.

One vs All:

One vs All. This strategy involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This strategy requires the base classifiers to produce a real-valued confidence score for its decision, rather than just a class label; discrete class labels alone can lead to ambiguities, where multiple classes are predicted for a single sample.

One vs One:

One vs One. This strategy training of (k*(k-1))/2 classifiers for a k-way multi class problem; each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all $K\,(K-1)\,/\,2$ classifiers are applied to an unseen sample and the class that got the highest number of "+1" predictions gets predicted by the combined classifier. Like One vs All, One vs One also suffers from ambiguities in that some regions of the input space may receive the same number of votes.

Datasets and their description:

Diabetes dataset: This Pima Indian Diabetes (PID) Dataset is sourced from UCI Machine Learning Repository. They have 8 feature attributes and 1 decision attribute. There are two classes positive diabetes (label 1) and negative diabetes (label 0). The total count of the records is 768. The 8 features are,

1.) Number of times the patient is pregnant.
2.) Concentration of Glucose and plasma.
3.) Diastolic Blood Pressure.
4.) Triceps Skin flod Thickness
5.) Serum insulin
6.) BMI (Body Mass Index)
7.) Diabetes pedigree function
8.) Age
9.) Label of the Class

The dataset had all the data in floating point numbers only. This dataset can be used to predict whether a patient is diabetes positive or negative based on the different values of the above mentioned parameters.

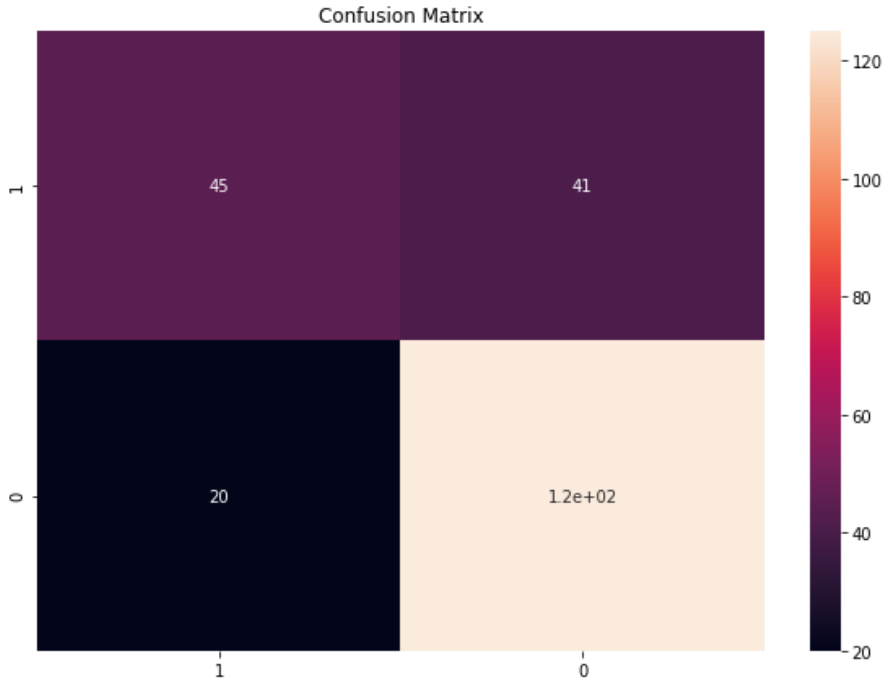
Diabetes is a most common disorder in people, where in the glucose levels consistently stay high or above normal conditions leading to diabetes. Human body uses glucose as main source of energy. The glucose we have in our body comes from the intake food. Insulin, a hormone produced by pancreas helps in regulating the amount of glucose in blood. It allows the glucose in blood to get in to the cells for energy. Irregularities in producing the insulin leads to high levels of glucose in blood resulting in diabetes.

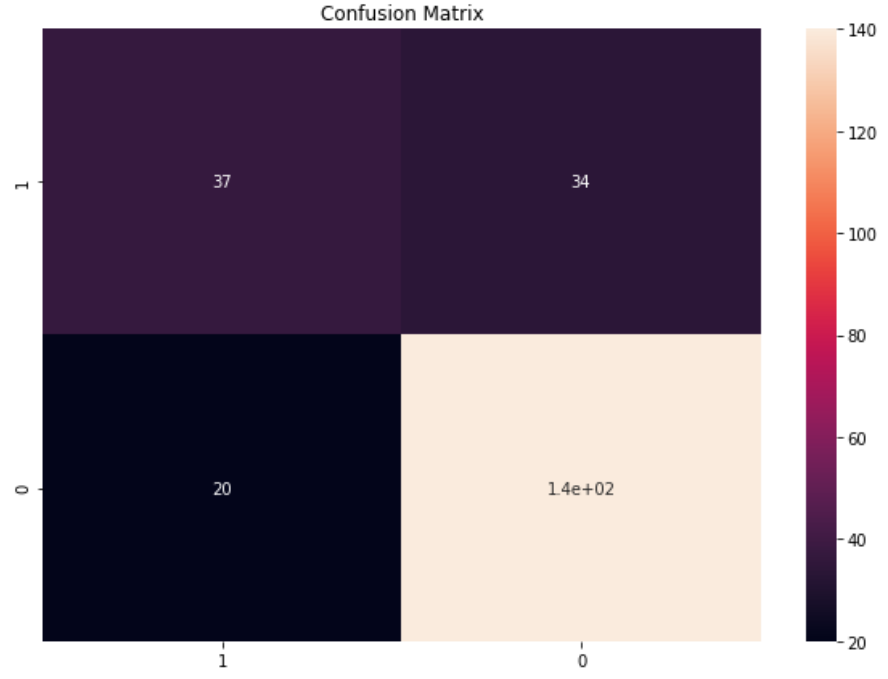
There are two types of diabetes , Type-1 and Type-2. In type-1 diabetes insulin is not produced at all. So the glucose levels are very high. Whereas in type-2 diabetes, the body does not produce enough amounts of insulin, gradually leading to higher amounts of glucose in the body. This is most prevalent in middle age groups even though it can occur to people of all ages. This is difficult to identify until the disorder causes long term damage. So this is an important problem. The PIMA dataset is a type-2 diabetes dataset.

Results for this dataset:

Accuracy of the linear least squares: 0.735930735931
Accuracy for the SVM : 0.770562770563
Accuracy for ANN : 0.766233766234

The confusion matrices for our algorithm and for the ANN





Cars Evaluation

We have used the car evaluation dataset, which can be used to predict the condition of the car based on attributes such as Price and Technical characteristics. And there is a hierarchy in the price and comfort. In the price there are features like buying price and maintenance price, In Technical characteristics there are features like number of doors, number of seats, persons, boot-space, and safety.

The dataset had all the data in text so each text item is mapped to a number so that it can be used to with the linear least square model.

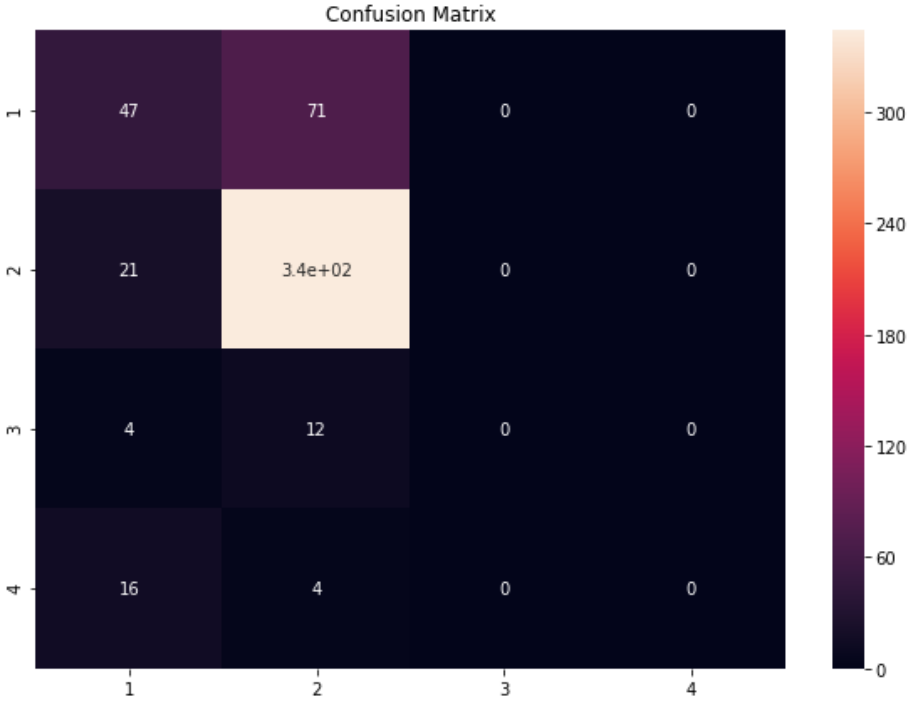
This class consist a total of 1728 records and four different classes, which are differentiated using different proportions of all the attributes listed above. The four classes are,

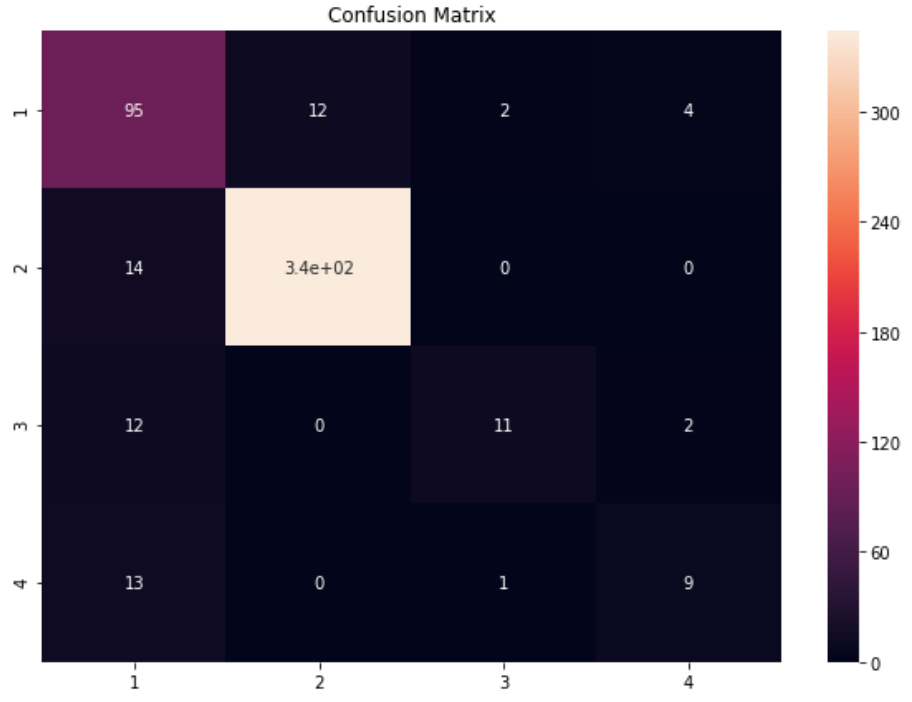
1.) unacc 2.) acc 3.) good 4.) vgood

Results for this dataset:

Accuracy of the linear least squares : 0.753371868979
Accuracy of the SVM : 0.815028901734
Accuracy of the ANN : 0.884393063584

The confusion matrices for our algorithm and for the ANN





Wine

Wine Dataset: These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. So, the total number of attributes on which the three different types of wines (class-1, class-2 and class-3) are distinguished are 13.

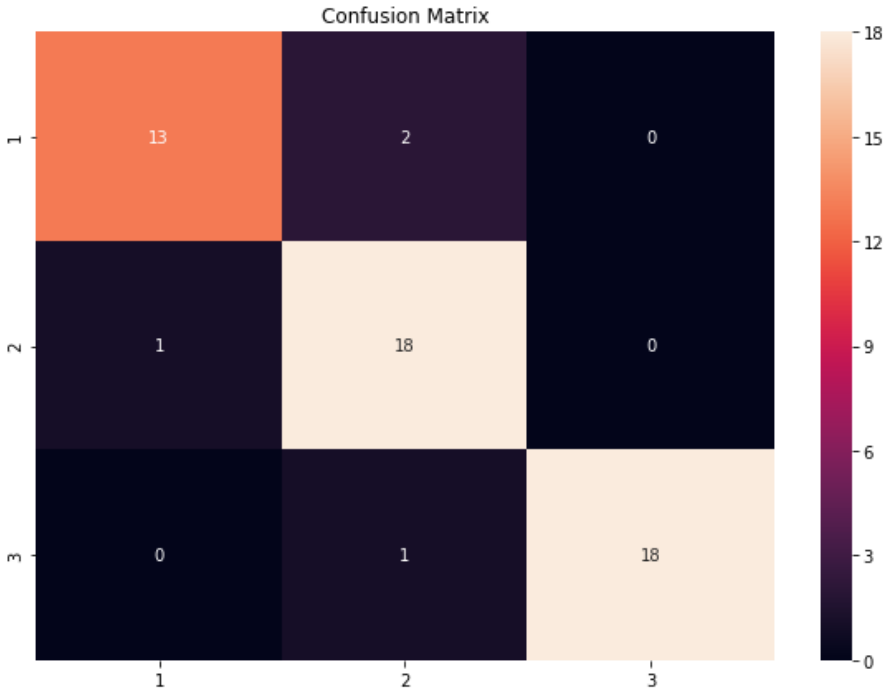
The attributes are,

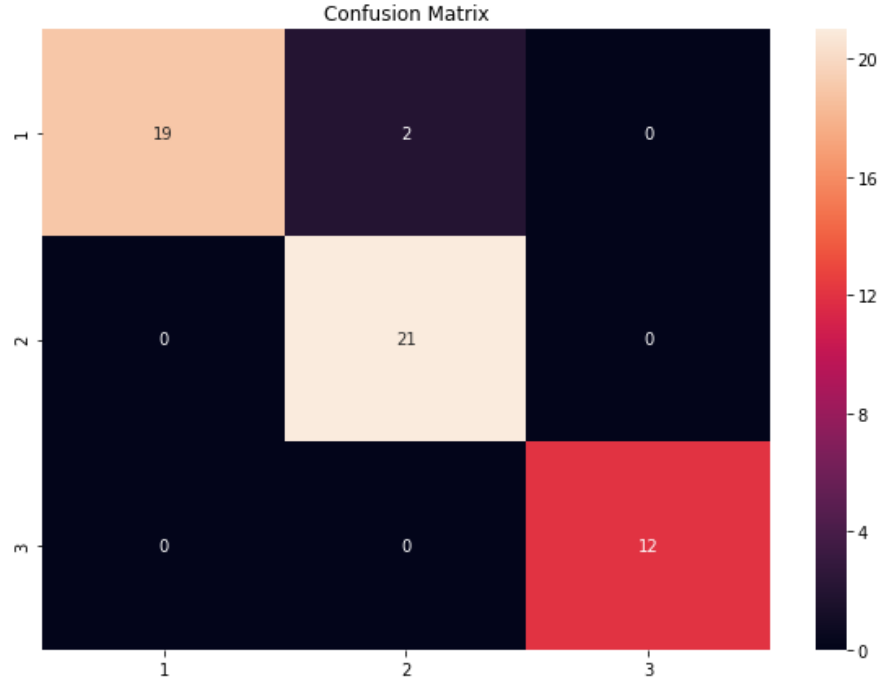
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

Results for this dataset:

Accuracy of the linear least squares:	0.924528301887
Accuracy of the SVM :	0.925925925926
Accuracy of the ANN :	0.962962962963

The confusion matrices for our algorithm and for the ANN





IRIS

Iris Dataset: This is one of the best known datasets to be found in the pattern recognition literature. The data set contains 3 classes and a total of 150 instances, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other. This is a really simple domain.

The total number of attributes are 4 and they are,

1) sepal length in cm
2) sepal width in cm
3) petal length in cm
4) petal width in cm

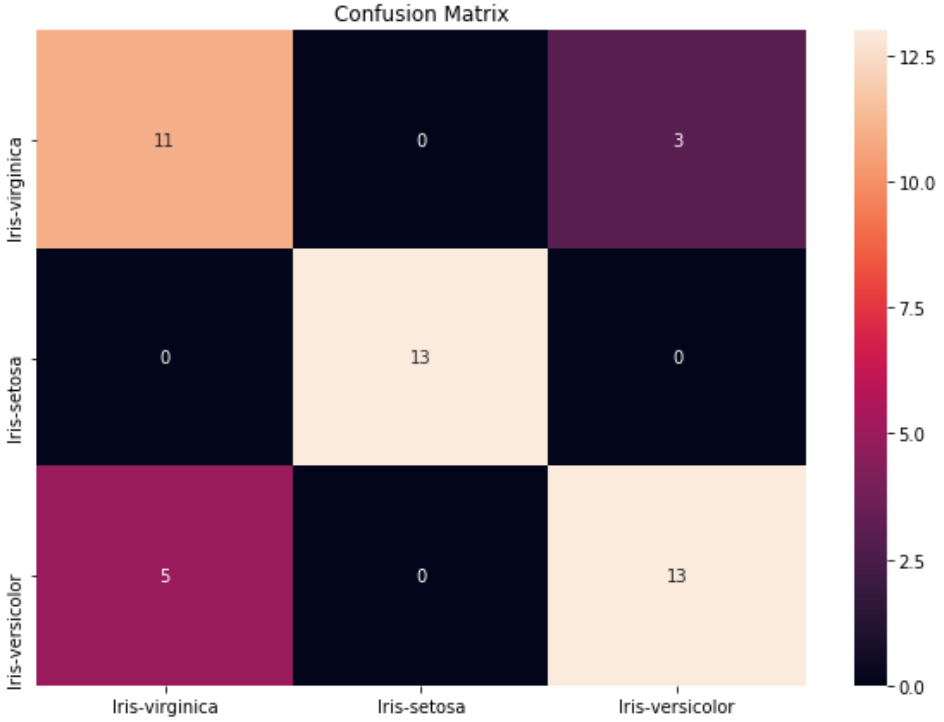
The three different classes are,

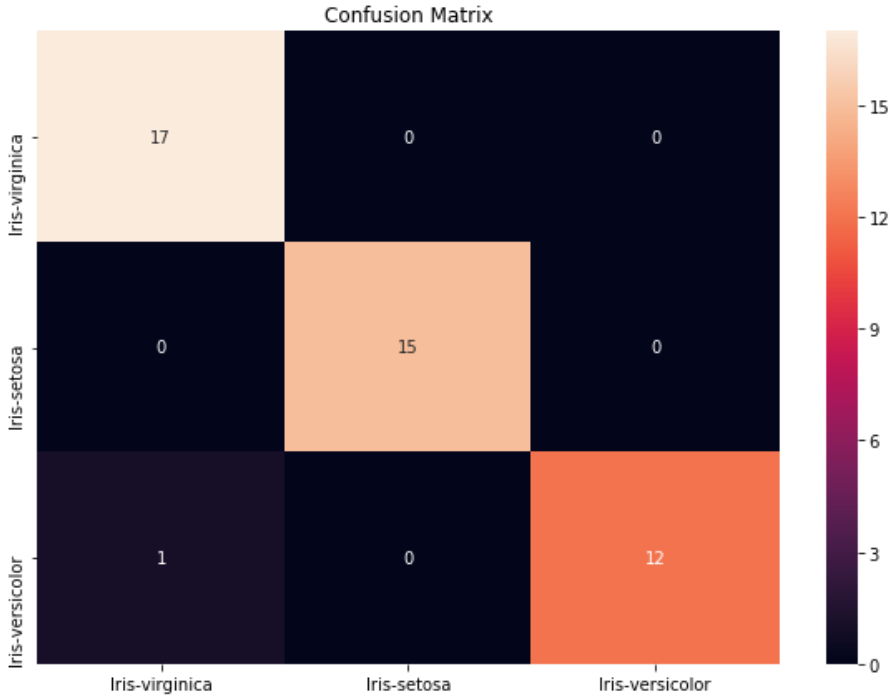
1) Iris-virginica
2) Iris-setosa
3) Iris-versicolor

Results for this dataset:

Accuracy of the linear least squares: 0.822222222222
Accuracy of the SVM : 0.955555555556
Accuracy of the ANN : 0.977777777778

The confusion matrices for our algorithm and for the ANN





Note:

1.) All the programs are written in Python 2.7 and executed in MacBook Pro 2017, Intel i5 2.7Ghz, 8GB Memory.
2.) All the library code is written in tester.py file and the main algorithm in main.py file.
3.) All the results are listed out after executing the program only once.

Reasons for choosing the datasets:

We have chosen the Diabetes data set to test how our algorithm performs on medical data. And the Car Evaluation dataset to build an linear model so that used car companies like AutoTrader.com can quickly evaluate the condition of the car using the recorded attributes.

References:

Datasets:

1.) Car Evaluation: https://archive.ics.uci.edu/ml/datasets/car+evaluation
2.) Iris: https://archive.ics.uci.edu/ml/datasets/Iris
3.) Wine: https://archive.ics.uci.edu/ml/datasets/wine
4.) Diabetes: https://www.kaggle.com/uciml/pima-indians-diabetes-database

Others:

1.) ANN: https://en.wikipedia.org/wiki/Artificial_neural_network, https://scikit-learn.org/stable/modules/neural_networks_supervised.html

2.) SVM: https://en.wikipedia.org/wiki/Support-vector_machine, https://scikit-learn.org/stable/modules/svm.html