CSE 4088

INTRODUCTION TO MACHINE LEARNING

FINAL REPORT

Comparison of Consistency-based & Correlation-based Feature Selection Algorithms

1. Project Members:

Zafer Emre OCAK *150113075*
Ozan GÜLHAN *150114013*

2. Abstract:

In machine learning algorithms, one of the biggest problems is scarcity of time and requirement of huge datasets. Feature selection provides optimization for these matters. Reducing time and amount of data required can change the fitness of the model significantly. By aiming this, feature selection algorithms are very important.
In order to see those effects, we consider implementing the algorithms which are “consistency-based” and “ correlation-based” feature selection. In addition to that, we can also compare the required amount of time and data to reach the same fitness levels for the model.

To pick up a model, we have had a lot of different options but among them, we have decided on Bank Marketing Dataset. One of the most desired features of our project was finding an appropriate data. We have searched on the internet and it has two essential properties that first, high number of features and second, a lot of instances available.
In the model, the classification goal is to predict if the client will subscribe (yes/no) a term deposit.

Overall, we would like to reduce number of attributes and to decrease least amount of time to train the model for the same fitness levels.

3. Overview of the Project

- Finding dataset and preparing it for the feature selectors.
- Constructing the ANN model.
- Training model with raw dataset which means we didn’t modify dataset at all.
- Running the first selector and obtaining the best subset among all the candidate subsets.
- Running the second selector and obtaining the best subset among all the candidate subsets.
- Training the model with new subsets given by feature selectors.
- Analyzing the results.

Ozan's Part	***Emre's Part***
Feature selectors were available as an open source packages in the Weka library. I created a new class which calls selectors and get the results. I gave the dataset location and run the selectors. Generated subsets were given to ANN.	I searched on the internet and construct an artificial neural network in Keras library in Python. I got familiar with the concept of layers, epochs and dataset features. In order to utilize feature selection algorithms, I had to have knowledge on data dimensionality. We analyzed data together.

4. Accomplishment:

i. Finding dataset

At first, we were struggling to find a suitable dataset for our model. Because, in some cases available datasets may not be convenient for the model to be used. Thus, we have found a dataset which contains 10000 people's bank information. This dataset was an Excel file has a format of comma seperated values(.csv). So, at the last column, the dataset has labels of each customer's credit approval. All the labels have binary output indicating 1s and 0s.

ii. Building ANN model

To begin with, we have made a comprehensive research on the internet to figure out the Keras library and its functions. As constructing the model we put 1 input-layer, 2 hidden-layer which have activation function of ReLu and 1 output-layer for generating the classification output. Our input layer receives 11 features as inputs because out of 13 features, one of them is for the labelling data and one of them was unnecessary feature - *row number* - which doesn't any effect on the output result.

iii. Model training on raw data

Training with raw data, the model had 11 dimensions as inputs and 1 binary output. Before modification of the dataset via feature selection, these were the total number of features in the model. We tracked the total duration via putting start and end temporary variables and in addition to that, we have analyzed the accuracy rate after the last epoch the model has passed.

iv. Running the Selectors

a. Correlation-based Feature Selection

- **‘**Good feature subsets contain features highly correlated with the classification , yet uncorrelated to each other.’ [2]
- This means that, the more a feature is uncorrelated with other features in the subset, it has a good effect on the merit of subset, also the more it’s correlated with the class , it also has a good effect.

$$\text{Merit}_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

- *The following equation gives the merit of a feature subset S consisting of k features.*
- *Rcf is the average value of all feature-classification correlations.*
- *Rff is the average value of all feature-feature correlations.*

b. Consistency-based Feature Selection

- When this algorithm takes the candidate subset, it looks for all the instances where their attribute values match but their class labels are different. This kind of patterns are inconsistent. Count of these patterns is the inconsistency count.
- Inconsistency rate of a subset S is the sum of all the inconsistency counts over all patterns of the feature subset that appearsn in the data divided by total number of instances.
- The less this number for a subset, the better for that subset.

v. Training the model with new subsets

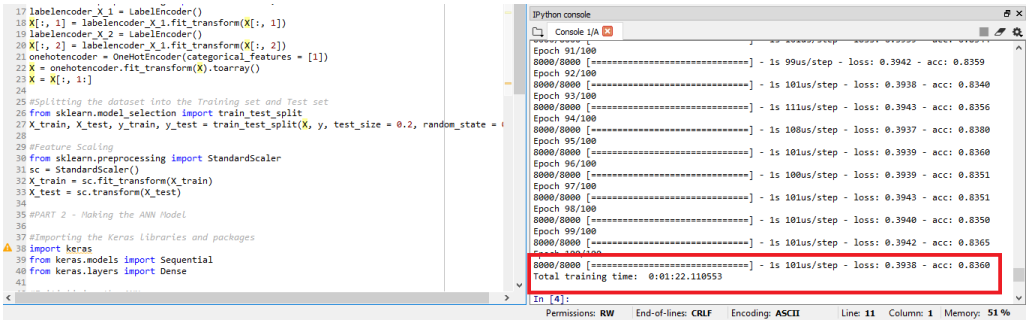
This part is the main concept of our project. We have given the substituted features from the selector algorithms and gave the model to observe the differences.

So we have a graph below that clearly explaining what is happening in different datasets with different selection methods.

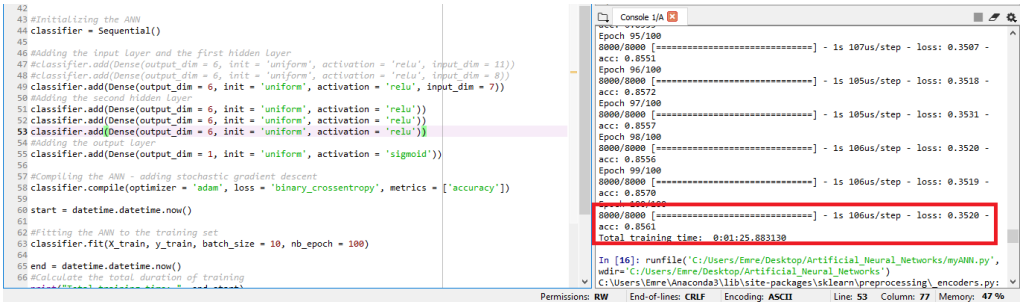
	Total Features Used	Accuracy Rate	Total Duration
Raw Model	11	%83.60	83 seconds
Correlation-based	8	%85.61	85 seconds
Consistency-based	7	%83.41	80 seconds

vi. Analyzing the results and experiments carried out

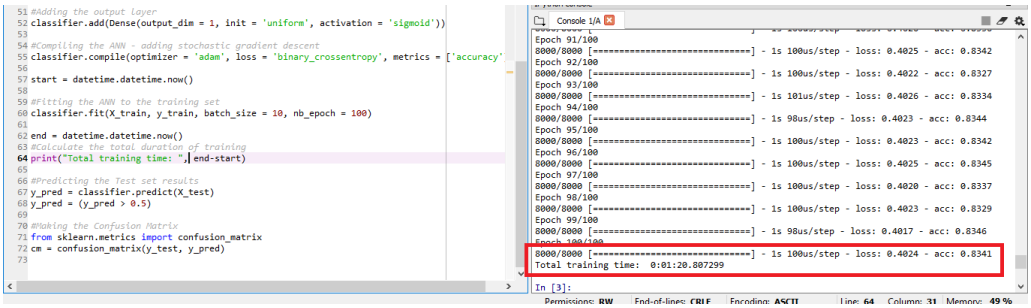
At this part, we illustrate the experiment results as below:



These experiment shows the model accuracy and total training time with *raw dataset*.



These experiment shows the model accuracy and total training time with *the dataset correlation-based selector has been applied*.



These experiment shows the model accuracy and total training time with *the dataset consistency-based selector has been applied*.

a. Installing essential libraries

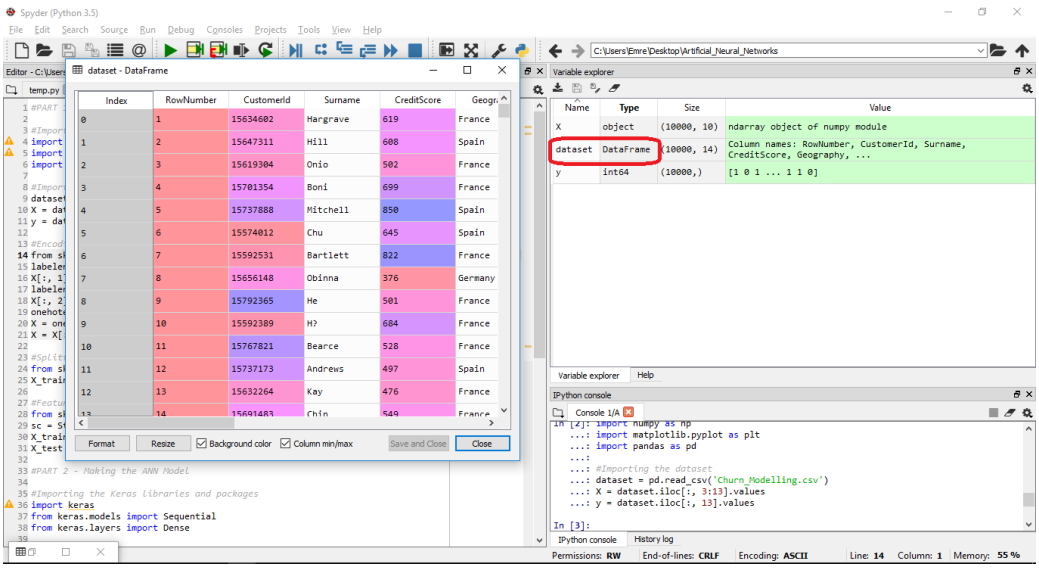
In this very first step of the project, we determined we need to certify the essential libraries to be used in our model.

Imported libraries is going to be explained briefly. The libraries were:

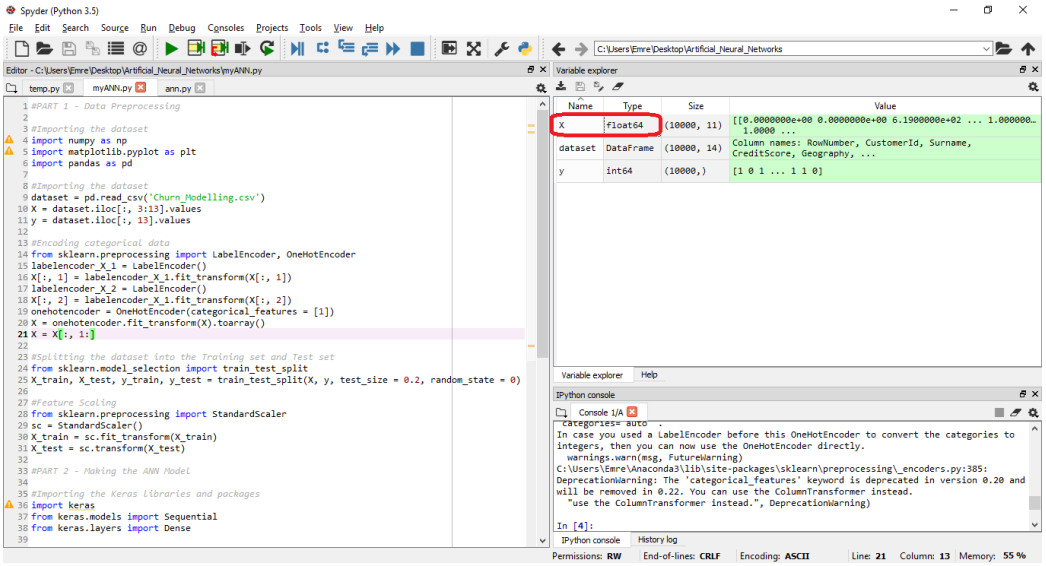
- *Numpy Lib of Python*
  Basically, this library serves as a mathematical optimizer in Python. We had an idea that in machine learning algorithms, capability of arithmetic operations is a must. Thus, since we already implement Artificial Neural Networks(ANN) in our model, we knew we would need that library beforehand.

- *Pandas Lib of Python*
  Our both instance and complete dataset is in .csv format so we need to use this library to easily read the excel file.

- *LabelEncoder & OneHotEncoder Libs of Python*
  These libraries serves as fit the model dataset suitable for training. Via importing them, we can change the dataset format to array format. Prior to, as soon as we read our dataset from the .csv file, it was in Object format.



Notice that, now we have floating data format so later, the model can manipulate the data.



- *Keras Lib of Python*
  Keras is the most vital and important library in our program. Without it, we would not able to create a suitable model and train it easily.

b. Getting a relatively small dataset for training and testing

In the former subtask, the adaption of the training and testing dataset have already been mentioned. In addition to that, array of X, stands for our input data and array of Y stands for our given output to adjust the weights in our model. We have 10000 data of accounts and 8000 of them have been used for training part and the remaining 2000 data of accounts have been used for testing the model.

c. Constructing the neural network

As Keras library has been already mentioned, it gave us the oppurtunity to easily construct layers and splitting up the training and testing dataset.

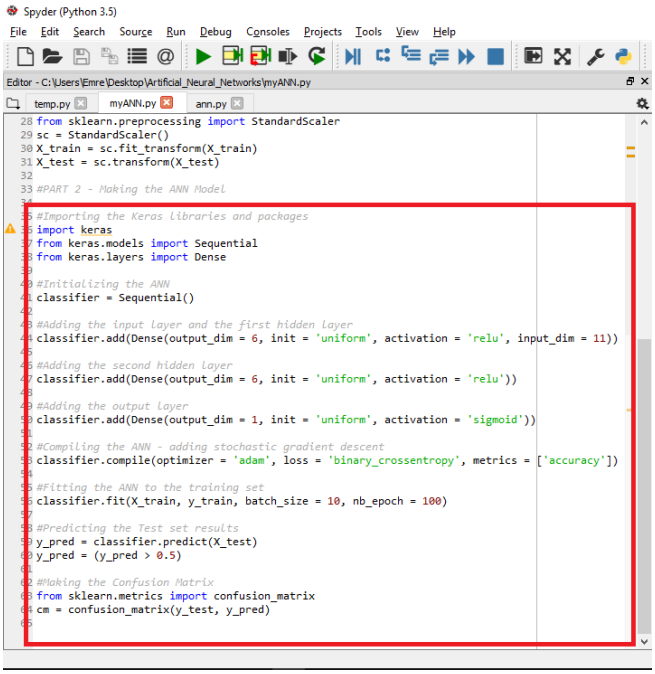
```
#Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)

#Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

```

In first two lines, %20 of the sample dataset is dedicated to testing part and other %80 of the sample dataset has been given to training part.

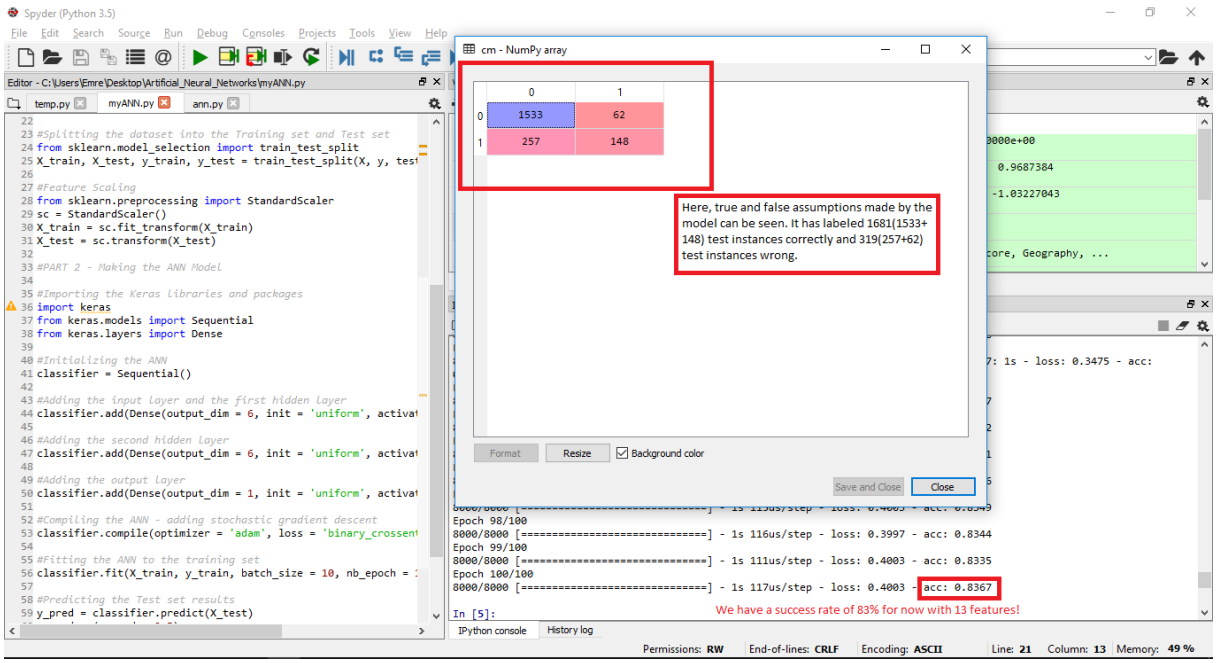
Under the label of feature scaling, X_train array has been used to altering weights and getting model to become ready for testing. X_test is obviously assigned to testing process.



This part of our program code, priorly, the corresponding import process has been done and afterwards, we determined the layers and the particular weight adjustion methods have been determined which are relu and sigmoid functions. In this model, we have 3 layers and 13 total of neurons. The weight's initial values are set according to uniform distribution.

With a batch size of 10 which means the data is going to be grouped by 10 for each iteration and will be given. Also, number of epoch is 100 so this means that our model will be trained by that dataset for 100 times.

d. Observing the initial performance of the model



We have 2 different possible outcomes of predictions, 1 or 0... The total correct and false answers can be shown with an explanation. Our current success rate is 83% with 13 different features.

5. Summary:

To briefly talking about the findings in this project, we have got a general understanding of feature selection algorithms and their corresponding reports. Moreover, we got familiar with the existing machine learning technologies and use them in order to develop a model. There were actually quite different concepts in that area. For instance, if a data scientist would like to process an image dataset then he/she probably need to handle convolutional neural networks.

Furthermore, we had significant amount of experience about the dataset compatibility for the ANN models. For example, we have dealt with data types for using consistency-based algorithm because the dataset compatibility with selector is essential.

We also altered the number of layers and the number of epochs in our model to observe the improvement in the model.

All these actions greatly contributed our knowledge of machine learning concept. There were ideas and undiscovered methods we could use in this project. To simply mention, we consider utilizing cross-validation and picking different model selection methods in collaboration with different model structures and with different classification problems.

6. References:

[1] Manoranjan Dash, Huan Liu, Consistency-based search in feature selection, Artificial Intelligence 151 (2003) 155–176

[2] Mark A. Hall, Correlation-based Feature Selection for Machine Learning, 1999

[3] https://keras.io/getting-started/sequential-model-guide/