Coding Assignment 2

Kaviyaa Vasudevan
Engineering Science (Data Science) MS
University at Buffalo, Buffalo, NY 14260
kaviyaav@buffalo.edu

Neural Network with Backpropagation

1.1 Definition

A neural network is a group of connected Input/Output units where each connection has a weight associated with its computer programs. It helps us to build predictive models from large databases. This model builds upon the human nervous system. It helps us to conduct image understanding, human learning, computer speech, etc. Backpropagation is the essence of neural network training. It is the method of fine-tuning the weights of a neural network based on the error rate obtained in the previous iteration. Proper tuning of the weights allows us to reduce error rates and make the model reliable by increasing its generalization.

Backpropagation in neural network is a short form for "backward propagation of errors." It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.

1.2 Interpreting Backpropogation

The Back propagation algorithm in neural network computes the gradient of the loss function for a single weight by the chain rule. It efficiently computes one layer at a time, unlike a native direct computation. It computes the gradient, but it does not define how the gradient is used. It generalizes the computation in the delta rule. Inputs X, arrive through the preconnected path. Input is modeled using real weights W. The weights are usually randomly selected. Calculate the output for every neuron from the input layer, to the hidden layers, to the output layer.

Calculate the error in the outputs:

ErrorB= Actual Output – Desired Output

Travel back from the output layer to the hidden layer to adjust the weights such that the error is decreased.

Two Types of Backpropagation Networks are:

- Static back-propagation:

It is one kind of backpropagation network which produces a mapping of a static input for static output. It is useful to solve static classification issues like optical character recognition.

- Recurrent Backpropagation:

Recurrent Back propagation in data mining is fed forward until a fixed value is achieved. After that, the error is computed and propagated backward.

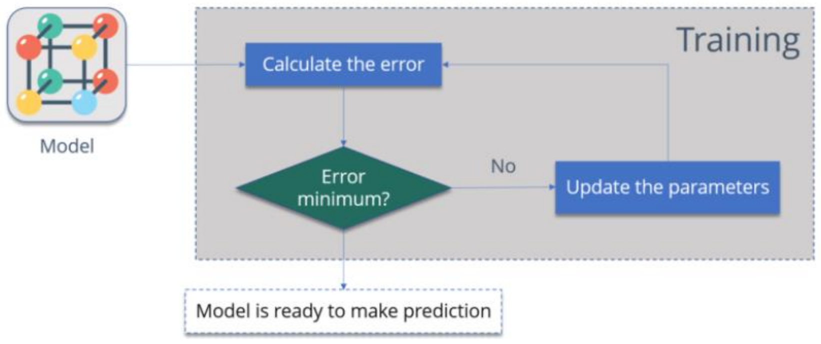


Figure 1.1: Working of Backpropogation

1.3 Back propagation in Neural Networks

The principle behind the back propagation algorithm is to reduce the error values in randomly allocated weights and biases such that it produces the correct output. The system is trained in the supervised learning method, where the error between the system's output and a known expected output is presented to the system and used to modify its internal state. We need to update the weights such that we get the global loss minimum. This is how back propagation in neural networks works.

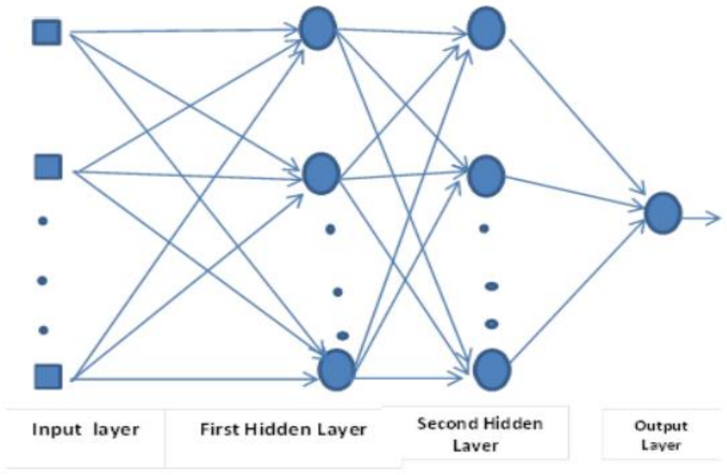
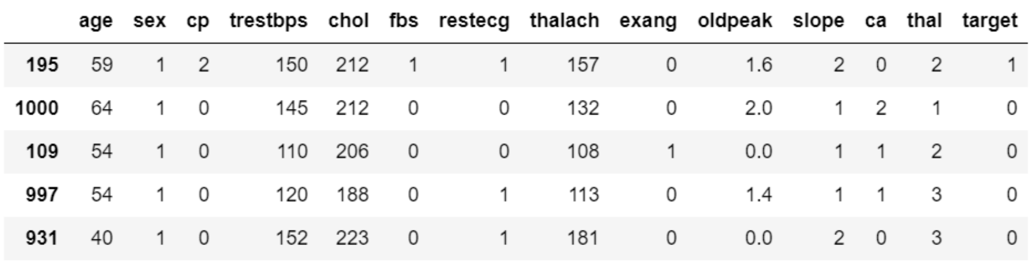


Figure 1.2: Backpropogation Networks Architecture

When the gradient is negative, an increase in weight decreases the error. When the gradient is positive, the decrease in weight decreases the error.

1.3 Data set

This data set dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V. It contains 76 attributes, including the predicted attribute, but all published experiments refer to using a subset of 14 of them. The "target" field refers to the presence of heart disease in the patient. It is integer valued 0 = no disease and 1 = disease. The attributes include age, sex, chest pain type (4 values), resting blood pressure, serum cholestoral in mg/dl, fasting blood sugar > 120 mg/dl, resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise induced angina, oldpeak = ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy, thal: 0 = normal; 1 = fixed defect; 2 = reversable defect. The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
195	59	1	2	150	212	1	1	157	0	1.6	2	0	2	1
1000	64	1	0	145	212	0	0	132	0	2.0	1	2	1	0
109	54	1	0	110	206	0	0	108	1	0.0	1	1	2	0
997	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0
931	40	1	0	152	223	0	1	181	0	0.0	2	0	3	0

Figure 1.3: Dataset

2 Data Preprocessing

The task of data preprocessing is to organize the original business data with the new "business model" , clear those attributes irrelevant to the aim of data mining, supply clean, accurate, simplified data so as to improve the quality and efficiency of excavation under the guidance of domain knowledge. The data preprocessing mainly concludes data cleaning, integration, transformation and reduction. In this way, the dirty, incomplete and inconsistent data in real world can be corrected. The foremost process of data preprocessing is to import the data set by which we develop models. This will be performed using the pandas library. Prior to importing the dataset we are required to import the necessary libraries. The elimination of noise instances is one of the most difficult problems in inductive ML. Usually the removed instances have excessively deviating instances that have too many null feature values which are also referred to as outliers.

Data cleaning: By filling the null, smoothing noise data, identify and delete the isolated data, and solve inconsistency to attain the goal of clearing data.

Data integration: Save the data belonging to several data sources to a consistent data storage (such as data warehouse), these data sources may include several databases, data cubes or ordinary files.

Data conversion: Convert the data into one form that is suitable for excavation, for instance, zoom the attributive data in proportion, make it falls into a comparatively smaller specified zone.

Data reduction: The compressed data used to acquire the dataset is much smaller than the original data, but it still keeps its integrity. Thus, the data mining will have more effect on the condensed dataset, and produce the same (or almost same) analysis result. Data preprocessing is indispensable to data mining. Statistics suggests that data preprocessing takes up 60 percent of the time in a complete process of data mining.

Missing data handling is another issue often dealt with in the data preparation steps. Moreover, in real-world data, the representation of data often uses too many features, but only a few of them may be related to the target concept. There may be redundancy, where certain features are correlated so that is not necessary to include all of them in modeling; and interdependence, where two or more features between them convey important information that is obscure if any of them is included on its own. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. Splitting the data set becomes one of the significant processes of data preprocessing. The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem.

2.1 Visualization

The next step is to visualize the results for which matplotlib library can be used to make scatter plots of our training set results and test set results. The below plot depicts the distance plot of the target variable in the dataset.

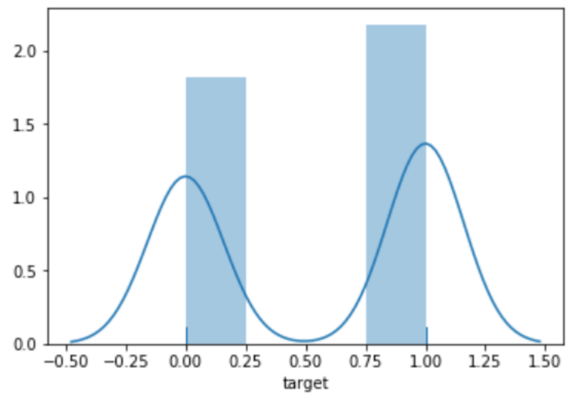


Figure 1.4: Distplot of the target

The below bar plots shows the relationship between the gender and the frequency of the heart disease. It shows that the male are more prone to the disease than the female.

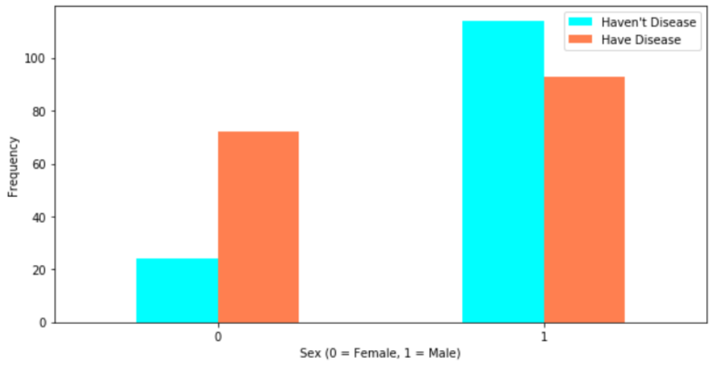


Figure 1.5: Frequency of heart disease with respect to gender

The below barplot shows the relation of the target with respect to the blood pressure of the patients.

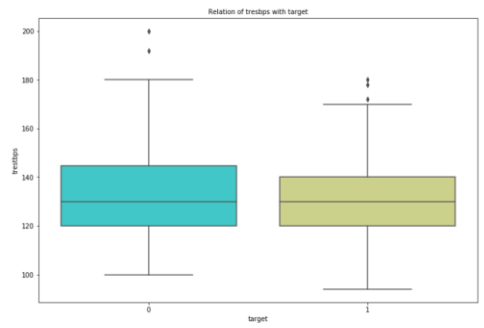


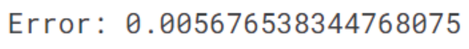
Figure 1.6: Relation of target with blood pressure

2.2 Fitting the Model and Predicting Results

The method computes the depth of the loss function in the input data with respects to all the weights in the network. The gradient techniques are then applied to the optimization methods to adjust the weights to minimize the loss function in the network. Hence, the algorithm requires a known and a desired output for all inputs in order to compute the gradient of loss function. Input the Heart Diseases dataset into MLPBPA, Set the class attribute (num) as target value and pass onto the MLPBPA, Call trainbr, trainlm, traingdx, trainscg with Learngdm, MSE and purelin function to train, adpt train, fine tune performance and to transfer input/output respectively, Set the number of default epochs and goal as 10 and 0 , train the network until the target reached to desired output , If (target! =output) reinitialize the network and train network , Increase the number of neurons , Increase epochs, goal, number of hidden layers, transfer function and training algorithm , else stop the execution.

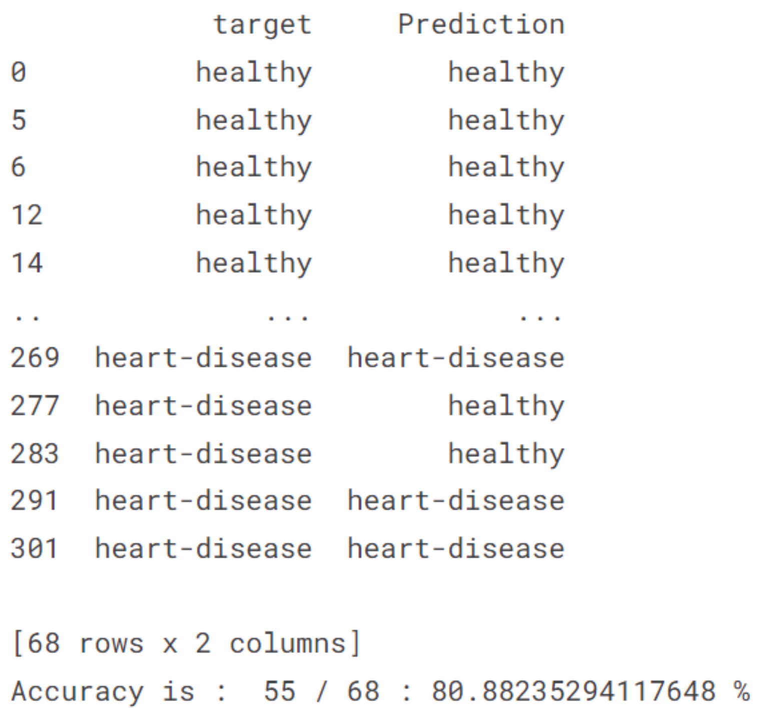
2.3 Error

We connect the hidden layer and input layer and then train the neural network by updating the weights. The error value we get from this is shown below:



```
Error: 0.005676538344768075
```

Figure 1.7: Error



```
            target     Prediction
0          healthy        healthy
5          healthy        healthy
6          healthy        healthy
12         healthy        healthy
14         healthy        healthy
..             ...            ...
269  heart-disease  heart-disease
277  heart-disease        healthy
283  heart-disease        healthy
291  heart-disease  heart-disease
301  heart-disease  heart-disease

[68 rows x 2 columns]
Accuracy is :  55 / 68 : 80.88235294117648 %
```

Figure 1.8: Predicting Results

Coding Assignment 2

Kaviyaa Vasudevan
Engineering Science (Data Science) MS
University at Buffalo, Buffalo, NY 14260
kaviyaav@buffalo.edu

Support Vector Machine

1.1 Definition

Support vector machine is another simple algorithm which is highly preferred by many as it produces significant accuracy with less computation power. Support Vector Machine, abbreviated as SVM can be used for both regression and classification tasks. But, it is widely used in classification objectives. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen.

1.2 Support Vector Machine Analysis

Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

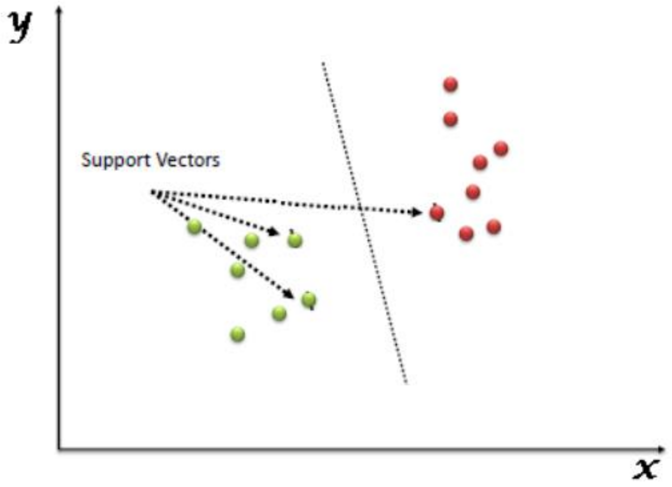


Figure 1.1 Datapoints classification using SVM

1.3 Cost Function and Gradient Updates

In the SVM algorithm, we are looking to maximize the margin between the data points and the hyperplane. The loss function that helps maximize the margin is hinge loss.

$$c(x, y, f(x)) = \begin{cases} 0, & \text{if } y * f(x) \geq 1 \\ 1 - y * f(x), & \text{else} \end{cases} \qquad c(x, y, f(x)) = (1 - y * f(x))_+$$

Figure 1.2: Hinge Loss

Hinge loss function (function on left can be represented as a function on the right). The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below.

$$min_w \lambda \parallel w \parallel^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+$$

Figure 1.3: Loss function for SVM

We have the loss function, we can take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights.

$$\frac{\delta}{\delta w_k} \lambda \parallel w \parallel^2 \; = 2\lambda w_k$$

$$\frac{\delta}{\delta w_k} \big(1 - y_i \langle x_i, w \rangle\big)_+ \; = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases}$$

Figure 1.4: Gradients

When there is no misclassification, i.e our model correctly predicts the class of our data point, we only have to update the gradient from the regularization parameter.

$$w = w - \alpha \cdot (2\lambda w)$$

Figure 1.5 Gradient Update — No misclassification

When there is a misclassification, i.e our model make a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update.

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w)$$

Figure 1.6 Gradient Update — Misclassification

1.4 Interpretation of SVM:

Let’s consider two independent variables x1, x2 and one dependent variable which is either a blue circle or a red circle.

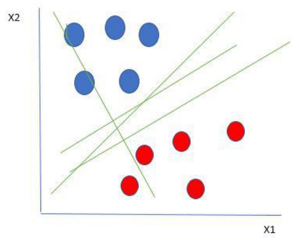


Figure 1.7: Hyperplane

From the figure above its clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregates our data points or does a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points.

1.4.1 Selecting the best hyper-plane:

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.

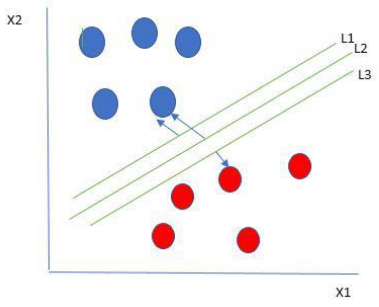


Figure 1.8: Selecting best hyperplane

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the maximum-margin hyperplane/hard margin. So from the above figure, we choose L2.

1.4.2 SVM Kernel

The SVM kernel is a function that takes low dimensional input space and transforms it into higher-dimensional space, ie it converts non separable problem to separable problem. It is mostly useful in non-linear separation problems. Simply put the kernel, it does some extremely complex data transformations then finds out the process to separate the data based on the labels or outputs defined.

1.5 Models with categorical predictor

Sometimes, a three-category variable can be included in a model as one covariate, where the best way is to incorporating three unordered categories is to define two different indicator variable.

The two commonly used techniques to deal with categorical variables:

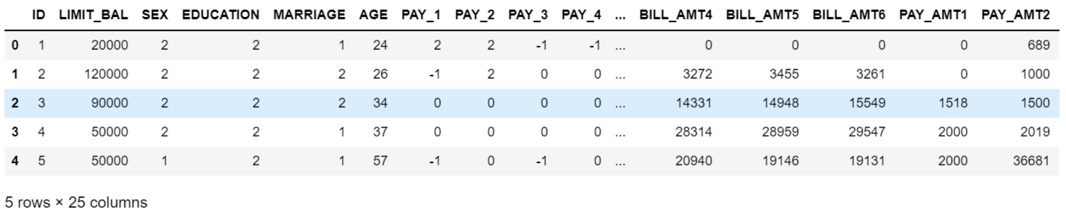
One-Hot Encoding - Convert a variable with N classes into N separate variables with binary labels. Repeat for each of the 3 variables.

Label Encoding - Map categorical variables into integers.

Label Encoding works only if there's some inherent order in the variables. Variables like these are called ordinal. Example would be a variable like days of the week. Monday can be one 1, Tuesday can be 2 and so on. Here the classifier would assume that 2 is greater than 1 in some way, which is fine as there is some order in the variable. One hot encoding is for the case where the variables are not ordinal - like names of places. This also gets used a lot in natural language processing.

1.6 Data set

This dataset contains information on default payments, demographic factors, credit data, history of payment, and bill statements of credit card clients in Taiwan from April 2005 to September 2005. There are 25 variables like ID which gives the ID of each client LIMIT_BAL says about the Amount of given credit in NT dollars (includes individual and family/supplementary credit SEX implies the Gender (1=male, 2=female) EDUCATION is the highest completed education of the customer (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown) MARRIAGE shows the Marital status (1=married, 2=single, 3=others) AGE gives Age of customers in years PAY_0 gives the Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above) PAY_2 gives the Repayment status in August, 2005 (scale same as above) PAY_3 shows the Repayment status in July, 2005 (scale same as above) PAY_4 gives the Repayment status in June, 2005 (scale same as above) PAY_5 implies the Repayment status in May, 2005 (scale same as above) PAY_6 results in the Repayment status in April, 2005 (scale same as above) BILL_AMT1 is the Amount of bill statement in September, 2005 (NT dollar) BILL_AMT2 is the Amount of bill statement in August, 2005 (NT dollar) BILL_AMT3 is the Amount of bill statement in July, 2005 (NT dollar) BILL_AMT4 is the Amount of bill statement in June, 2005 (NT dollar) BILL_AMT5 is the Amount of bill statement in May, 2005 (NT dollar) BILL_AMT6 is the Amount of bill statement in April, 2005 (NT dollar) PAY_AMT1 is the Amount of previous payment in September, 2005 (NT dollar) PAY_AMT2 is the Amount of previous payment in August, 2005 (NT dollar) PAY_AMT3 is the Amount of previous payment in July, 2005 (NT dollar) PAY_AMT4 is the Amount of previous payment in June, 2005 (NT dollar) PAY_AMT5 is the Amount of previous payment in May, 2005 (NT dollar) PAY_AMT6 is the Amount of previous payment in April, 2005 (NT dollar).



	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_1	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689
1	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000
2	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500
3	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019
4	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681

5 rows × 25 columns

Figure 1.9: Dataset

2 Data Preprocessing

The data preprocessing can often have a significant impact on generalization performance of a ML algorithm. The foremost process of data preprocessing is to import the data set by which we develop models. This will be performed using the pandas library. Prior to importing the dataset we are required to import the necessary libraries. The elimination of noise instances is one of the most difficult problems in inductive ML. Usually the removed instances have excessively deviating instances that have too many null feature values which are also referred to as outliers. Missing data handling is another issue often dealt with in the data preparation steps. Moreover, in real-world data, the representation of data often uses too many features, but only a few of them may be related to the target concept. There may be redundancy, where certain features are correlated so that is not necessary to include all of them in modeling; and interdependence, where two or more features between them convey important information that is obscure if any of them is included on its own. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. The Dummy Variable trap is a scenario in which two or more variables are highly correlated; in simple terms, one variable can be predicted from the others. Intuitively, there is a duplicate category: if we dropped the male category, it is inherently defined in the female category (zero female value indicate male, and vice-versa). The solution to the dummy variable trap is to drop one of the categorical variables - if there are m number of categories, use m-1 in the model, the value left out can be thought of as the reference. Splitting the data set becomes one of the significant processes of data preprocessing. The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem.

we can see that all the 25 columns have 22003 count which indicates there is no missing value. we saw that the repayment status is indicated in columns PAY_0, PAY_2 ... with no PAY_1 column, so we rename PAY_0 to PAY_1 for ease of understanding. Next we check the datatype of each variable of dataset. We see that all the columns are int64 type whereas from previous knowledge we know that SEX, EDUCATION, MARRIAGE, PAY_0, PAY_2, PAY_3, PAY_4, PAY_5, PAY_6, default_payment_next_month are categorical features. So we convert these features in categorical. There are no missing value hence no imputation, now we directly move towards visualization of defaulters dataset.

2.1 Visualization

The next step is to visualize the results for which matplotlib library can be used to make scatter plots of our training set results and test set results to see how close our model has predicted the values. We can see that the dataset consists of 77% clients are not expected to default payment whereas 23% clients are expected to default the payment.

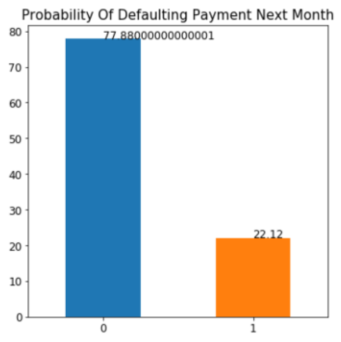


Figure 1.10: Bar plot

By plotting the continuous variables we observe that dataset consists of skewed data of limiting balance and age of clients. We have more number of clients having limiting balance between 0 to 200000 currency. We have more number of clients from age bracket of 20 to 40, i.e., clients from mostly young to mid aged groups. We will observe the effect of variables on target variable below

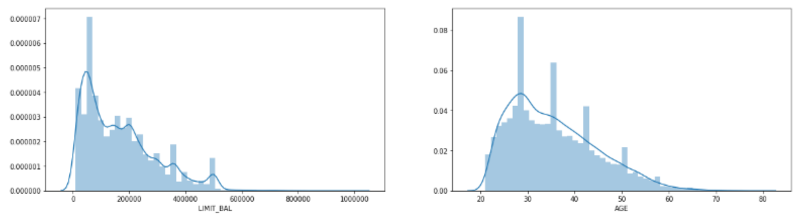


Figure 1.11: Plotting the continuous variables

Below plot indicates that there is higher proportion of clients for whom the bill amount is high but payment done against the same is very low. This we can infer since maximum number of datapoints are closely packed along the Y-axis near to 0 on X-axis

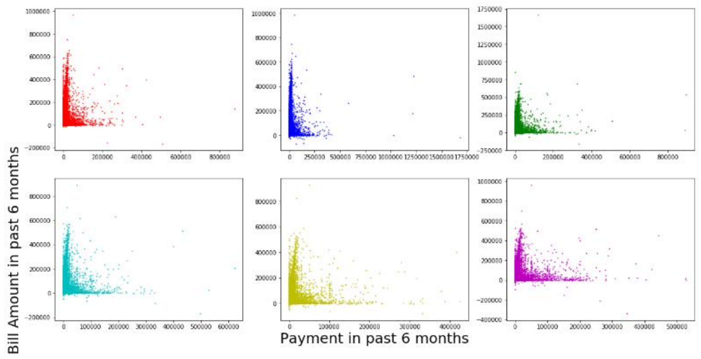


Figure 1.12: Shows the relation between bill amount and payment made

Below plot of Age against limiting balance does not provide any accurate information, as there is mixed variation of clients of all age groups and their current month limiting balance.

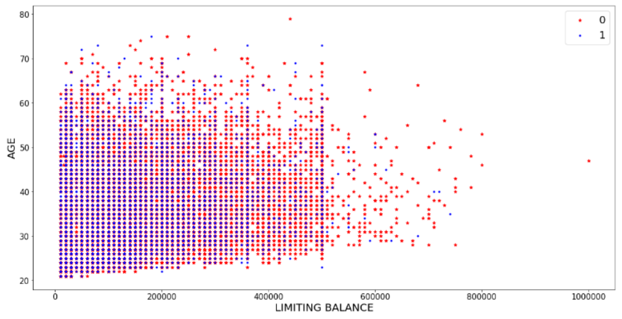
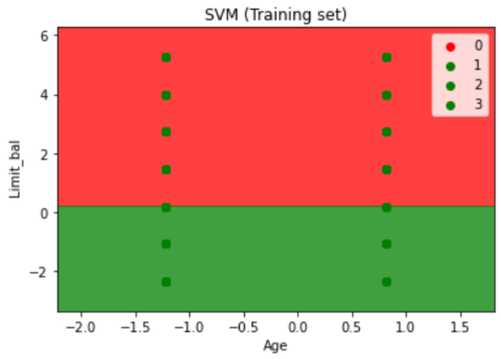


Fig 1.13 Relation between limiting balance and age

2.1 Fitting the Model and Predicting Results

The focus was to look at the performance of the SVM ensemble algorithm as well as to increase the accuracy of the SVM. The data will be partitioned into two sets of 70% training set and30% testing set. The results of this separation contain 9,290 training data and 3,980 test data. Then apply the SVM algorithm. The accuracy of classification using SVM is 89.9%. The sensitivity value is 67.9%, which means that the ability to provide positive results for prospective credit card customers default is 67.9%. Specificity value is 68.1%, which means the ability to give negative results to prospective credit card customers who will not default to 68.1%. Accuracy score of 89%, which means the ability to correctly detect all tested subjects of 89% .



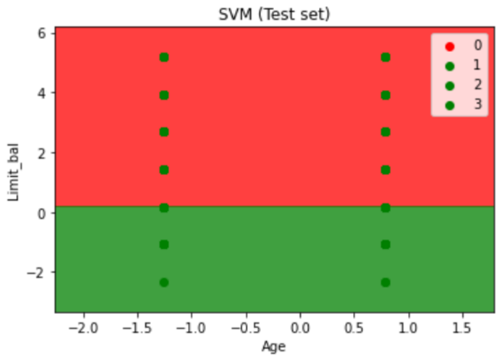


Figure 1.14 Training set results

Figure 1.15 Test set results

Coding Assignment 2

Kaviyaa Vasudevan
Engineering Science (Data Science) MS
University at Buffalo, Buffalo, NY 14260
kaviyaav@buffalo.edu

Naïve Bayes Classification

1.1 Definition

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. The independence assumptions often do not have an impact on reality. Therefore, they are considered as naive. We can derive probability models by using Bayes' theorem. Depending on the nature of the probability model, you can train the Naive Bayes algorithm in a supervised learning setting.

Types of Naïve Bayes Model: There are three types of Naive Bayes Model, which are given below:

Gaussian: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

Multinomial: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.

Bernoulli: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

1.2 Interpreting Naïve Bayes Algorithm

A Naive Bayes model consists of a large cube that includes the following dimensions:

Input field name:

Input field value for discrete fields, or input field value range for continuous fields. Continuous fields are divided into discrete bins by the Naive Bayes algorithm

Target field value:

This means that a Naive Bayes model records how often a target field value appears together with a value of an input field.

The Naive Bayes classification algorithm includes the probability-threshold parameter ZeroProba. The value of the probability-threshold parameter is used if one of the mentioned dimensions of the cube is empty. A dimension is empty, if a training-data record with the combination of input-field value and target value does not exist. The default value of the probability-threshold parameter is 0.001. Optionally, you can modify the probability threshold.

1.3 Bayes' Theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 1.1 Bayes' theorem

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

1.4 Math behind Naive Bays Algorithm:

Given a features vector X=(x1,x2,…,xn) and a class variable y, Bayes Theorem states that:

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

We're interested in calculating the posterior probability P(y | X) from the likelihood P(X | y) and prior probabilities P(y),P(X). Using the chain rule, the likelihood P(X ∣ y) can be decomposed a

$$P(X|y) = P(x_1, x_2, \dots, x_n|y)$$

$$= P(x_1|x_2, \dots, x_n, y) * P(x_2|x_3, \dots, x_n, y) \dots P(x_n|y)$$

but because of the Naive's conditional independence assumption, the conditional probabilities are independent of each other.

$$P(X|y) = P(x_1|y) * P(x_2|y) \dots P(x_n|y)$$

Thus, by conditional independence, we have:

$$P(y|X) = \frac{P(x_1|y) * P(x_2|y) \dots P(x_n|y) * P(y)}{P(x_1) * P(x_2) \dots P(x_n)}$$

And as denominator remains constant for all values, the posterior probability can then be:

$$P(y|x_1, x_2, \dots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

The Naive Bayes classifier combines this model with a decision rule. One common rule is to pick the hypothesis that's most probable; this is known as the maximum a posteriori or MAP decision rule.

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

1.5 Models with categorical predictor

The two commonly used techniques to deal with categorical variables:

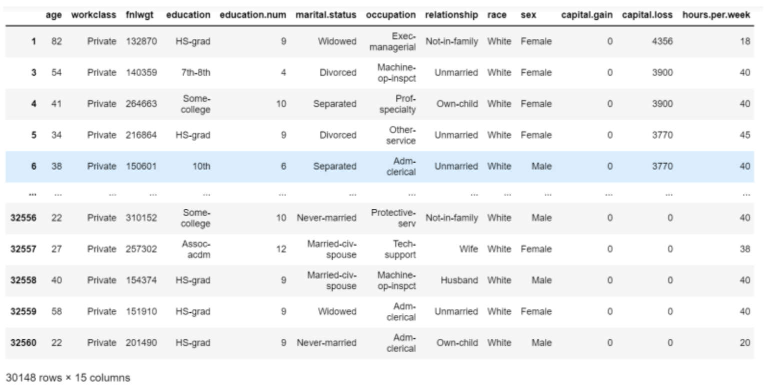
One-Hot Encoding - Convert a variable with N classes into N separate variables with binary labels. Repeat for each of the 3 variables.
Label Encoding - Map categorical variables into integers.

Label Encoding works only if there's some inherent order in the variables. Variables like these are called ordinal. Example would be a variable like days of the week. Monday can be one 1, Tuesday can be 2 and so on. Here the classifier would assume that 2 is greater than 1 in some way, which is fine as there is some order in the variable. One hot encoding is for the case where the variables are not ordinal - like names of places. This also gets used a lot in natural language processing.

1.6 Data set

The Adult UCI data set contains the questionnaire data of the Adult database (originally called the Census Income Database) formatted as a data frame. The Adult data set contains the data already prepared and coerced to transactions for use with a rules. Adult is an object of class transactions with 48842 transactions and 115 items. The Adult UCI data set contains a data frame with 48842 observations on the following 15 variables. Age a numeric vector, Workclass a factor with levels Federal-gov, Local-gov, Never-worked, Private, Self-emp-inc, Self-emp-not-inc, State-gov, and Without-pay. Education an ordered factor with levels Preschool < 1st-4th < 5th-6th < 7th-8th < 9th < 10th < 11th < 12th < HS-grad < Prof-school < Assoc-acdm < Assoc-voc < Some-college < Bachelors

< Masters < Doctorate. Education-num a numeric vector. Marital-status a factor with levels Divorced, Married-AF-spouse, Married-civ-spouse, Married-spouse-absent, Never-married, Separated, and Widowed. Occupation a factor with levels Adm-clerical, Armed-Forces, Craft-repair, Exec-managerial, Farming-fishing, Handlers-cleaners, Machine-op-inspct, Other-service, Priv-house-serv, Prof-specialty, Protective-serv, Sales, Tech-support, and Transport-moving. Relationship a factor with levels Husband, Not-in-family, Other-relative, Own-child, Unmarried, and Wife. Race a factor with levels Amer-Indian-Eskimo, Asian-Pac-Islander, Black, Other, and White. Sex a factor with levels Female and Male. Capital-gain a numeric vector. Capital-loss a numeric vector. Fnlwgt a numeric vector. Hours-per-week a numeric vector. Native-country a factor with levels Cambodia, Canada, China, Columbia, Cuba, Dominican-Republic, Ecuador, El-Salvador, England, France, Germany, Greece, Guatemala, Haiti, Holand-Netherlands, Honduras, Hong, Hungary, India, Iran, Ireland, Italy, Jamaica, Japan, Laos, Mexico, Nicaragua, Outlying-US(Guam-USVI-etc), Peru, Philippines, Poland, Portugal, Puerto-Rico, Scotland, South, Taiwan, Thailand, Trinadad&Tobago, United-States, Vietnam, and Yugoslavia. Income an ordered factor with levels small < large.



	age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race	sex	capital.gain	capital.loss	hours.per.week
1	82	Private	132870	HS-grad	9	Widowed	Exec-managerial	Not-in-family	White	Female	0	4356	18
3	54	Private	140359	7th-8th	4	Divorced	Machine-op-inspct	Unmarried	White	Female	0	3900	40
4	41	Private	264663	Some-college	10	Separated	Prof-specialty	Own-child	White	Female	0	3900	40
5	34	Private	216864	HS-grad	9	Divorced	Other-service	Unmarried	White	Female	0	3770	45
6	38	Private	150601	10th	6	Separated	Adm-clerical	Unmarried	White	Male	0	3770	40
...	...	...	...	...	...	...	...	...	...	...	...	...	...
32556	22	Private	310152	Some-college	10	Never-married	Protective-serv	Not-in-family	White	Male	0	0	40
32557	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38
32558	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40
32559	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40
32560	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20

30148 rows × 15 columns

Figure 1.2 Dataset

2 Data Preprocessing

The data preprocessing can often have a significant impact on generalization performance of a ML algorithm. The foremost process of data preprocessing is to import the data set by which we develop models. This will be performed using the pandas library. Prior to importing the dataset we are required to import the necessary libraries. The elimination of noise instances is one of the most difficult problems in inductive ML. Usually the removed instances have excessively deviating instances that have too many null feature values which are also referred to as outliers. Missing data handling is another issue often dealt with in the data preparation steps. Moreover, in real-world data, the representation of data often uses too many features, but only a few of them may be related to the target concept. There may be redundancy, where certain features are correlated so that is not necessary to include all of them in modeling; and interdependence, where two or more features between them convey important information that is obscure if any of them is included on its own. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. Splitting the data set becomes one of the significant processes of data preprocessing. The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem.

The dataset contains missing values that are marked with a question mark character (?). There are a total of 48,842 rows of data, and 3,620 with missing values, leaving 45,222 complete rows. There are two class values '>50K' and '<=50K', meaning it is a binary classification task. The classes are imbalanced, with a skew toward the '<=50K' class label.

'>50K': majority class, approximately 25%.

'<=50K': minority class, approximately 75%.

The Special characters that are found in the dataset is removed in this step. Our dataset contains the special characters such as "?", " ." . The rows containing the special characters are removed as a part of the cleaning process.

2.2 Visualization

The next step is to visualize the results for which matplotlib library can be used to make scatter plots of our training set results and test set results to see how close our model has predicted the values. The Seaborn Pairplot allows us to plot pairwise relationships between variables within a dataset. This creates a nice visualisation and helps us understand the data by summarising a large amount of data in a single figure. Pair plot is used to understand the best set of features to explain a relationship between two variables or to form the most separated clusters.

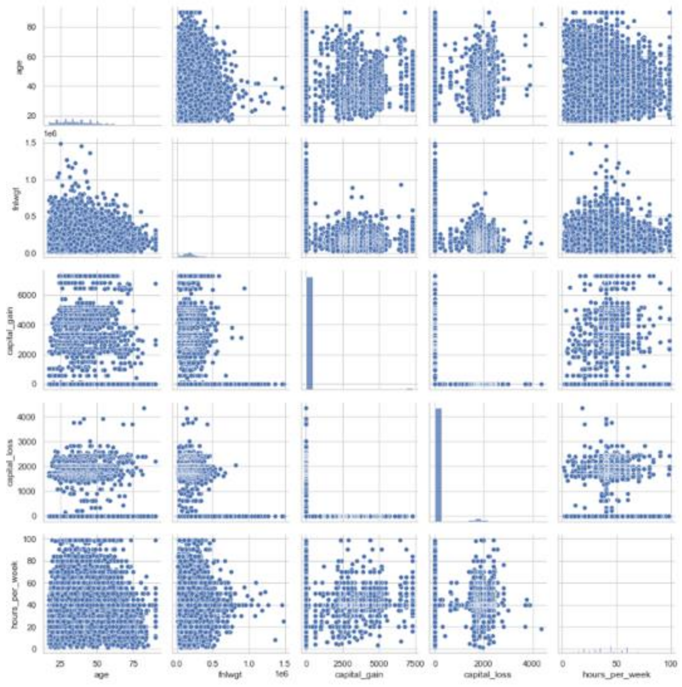


Figure 1.3 The above image shows the pairplot for all the columns.

The visualization results below for the income column specifies that there is a huge ratio ofpeople earning below 50k compared to above 50k.

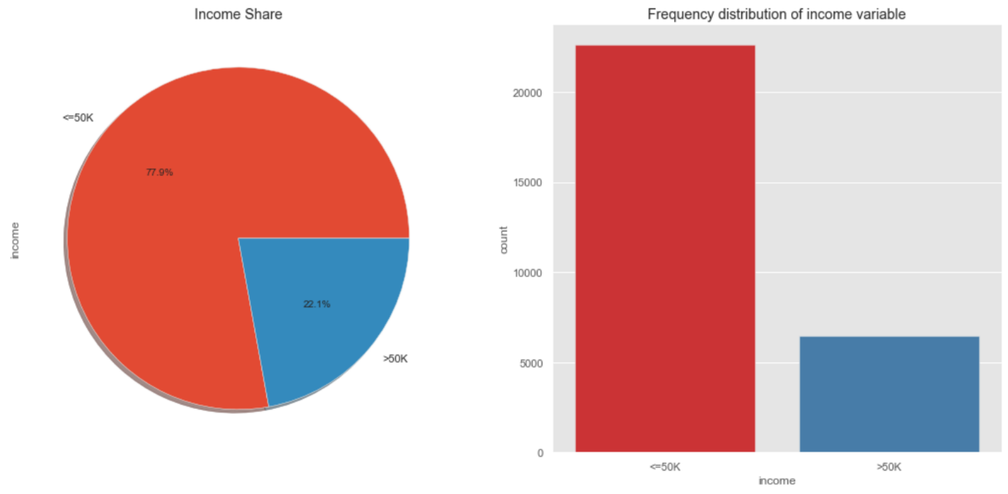


Figure 1.4 The above figure displays the visualization of income

2.2 Fitting the Model and Predicting Results

The entire dataset is split into training and test data set with the ratio of 20 and 80 respectively. The training dataset is trained with the gaussian naïve bayes algorithm and the testing is carries on the test set. The values are predicted after training the data set. The predicted results looked like below:

```
array(['<=50K', '<=50K', '>50K', ..., '>50K', '<=50K', '<=
50K'],
      dtype='<U5')
```

Figure 1.5 Predicting Results

The overall model accuracy is predicted to be around:

```
Model accuracy score: 0.8083
```

Figure 1.6: Accuracy

After comparing the training and test set accuracy, the training-set accuracy score is 0.8067 while the test-set accuracy to be 0.8083. These two values are quite comparable. So, there is no sign of overfitting. the results are as below:

```
Training set score: 0.8067
Test set score: 0.8083
```

Figure 1.7: Comparing Accuracies

So, the model accuracy is 0.8083. But, we cannot say that our model is very good based on the above accuracy. We must compare it with the null accuracy. Null accuracy is the accuracy that could be achieved by always predicting the most frequent class.

```
Null accuracy score: 0.7582
```

Figure 1.8 Null Accuracy

We can see that our model accuracy score is 0.8083 but null accuracy score is 0.7582. So, we can conclude that our Gaussian Naive Bayes Classification model is doing a very good job in predicting the class labels. Now, based on the above analysis we can conclude that our classification model accuracy is very good. Our model is doing a very good job in terms of predicting the class labels.

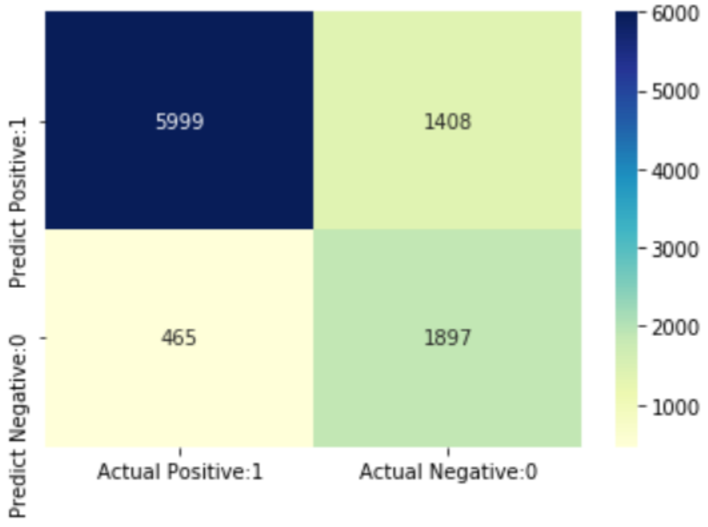


Figure 1.9 Visualizing Confusion Matrix

2.3 ROC Curve

Another tool to measure the classification model performance visually is ROC Curve. ROC Curve stands for Receiver Operating Characteristic Curve. An ROC Curve is a plot which shows the performance of a classification model at various classification threshold levels. The ROC Curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold levels. True Positive Rate (TPR) is also called Recall. It is defined as the ratio of TP to (TP + FN). False Positive Rate (FPR) is defined as the ratio of FP to (FP + TN). In the ROC Curve, we will focus on the TPR (True Positive Rate) and FPR (False Positive Rate) of a single point. This will give us the general performance of the ROC curve which consists of the TPR and FPR at various threshold levels. So, an ROC Curve plots TPR vs FPR at different classification threshold levels. If we lower the threshold levels, it may result in more items being classified as positve. It will increase both True Positives (TP) and False Positives (FP).

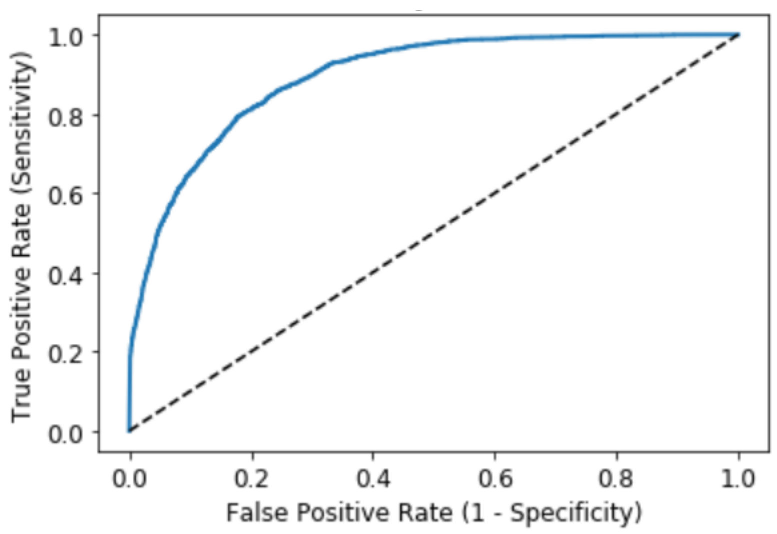


Figure 1.10 ROC Curve

Coding Assignment 2

Kaviyaa Vasudevan
Engineering Science (Data Science) MS
University at Buffalo, Buffalo, NY 14260
kaviyaav@buffalo.edu

Gaussian Mixture Model

1.1 Definition

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

1.2 Gaussian Mixture Analysis

The Gaussian Mixture object implements the expectation-maximization (EM) algorithm for fitting mixture-of-Gaussian models. It can also draw confidence ellipsoids for multivariate models, and compute the Bayesian Information Criterion to assess the number of clusters in the data. A Gaussian Mixture fit method is provided that learns a Gaussian Mixture Model from train data. Given test data, it can assign to each sample the Gaussian it mostly probably belongs to using the Gaussian Mixture predict method. The Gaussian Mixture comes with different options to constrain the covariance of the difference classes estimated: spherical, diagonal, tied or full covariance.

In real life, many datasets can be modeled by Gaussian Distribution (Univariate or Multivariate). So it is quite natural and intuitive to assume that the clusters come from different Gaussian Distributions. Or in other words, it is tried to model the dataset as a mixture of several Gaussian Distributions. This is the core idea of this model.

In one dimension the probability density function of a Gaussian Distribution is given by

$$f(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where myu and sigma square are respectively mean and variance of the distribution.

For Multivariate (let us say d-variate) Gaussian Distribution, the probability density function is given by

$$f(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi|\Sigma|}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^t \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})\right]$$

.

Here myu is a d dimensional vector denoting the mean of the distribution and epsilon is the d X d covariance matrix

1.3 The Gaussian Distribution

It has a bell-shaped curve, with the data points symmetrically distributed around the mean value. The below image has a few Gaussian distributions with a difference in mean (μ) and variance (σ2). Remember that the higher the σ value more would be the spread:

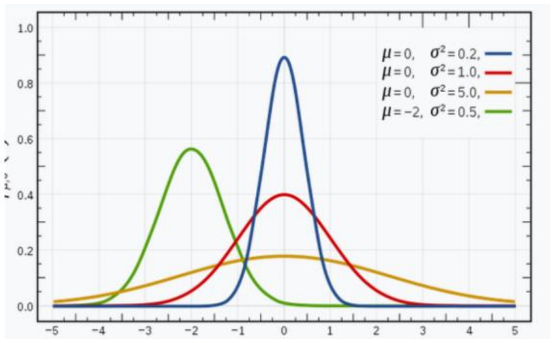


Figure 1.1 Gaussian Distribution

1.4 Expectation-Maximization (EM) Algorithm

The Expectation-Maximization (EM) algorithm is an iterative way to find maximum-likelihood estimates for model parameters when the data is incomplete or has some missing data points or has some hidden variables. EM chooses some random values for the missing data points and estimates a new set of data. These new values are then recursively used to estimate a better first date, by filling up missing points, until the values get fixed.

These are the two basic steps of the EM algorithm, namely E Step or Expectation Step or Estimation Step and M Step or Maximization Step.

E-step: In this step, the available data is used to estimate (guess) the values of the missing variables

M-step: Based on the estimated values generated in the E-step, the complete data is used to update the parameters

E-step:

For each point xi, calculate the probability that it belongs to cluster/distribution c1, c2, … ck. This is done using the below formula:

$$r_{ic} = \frac{\text{Probability } X_i \text{ belongs to c}}{\text{Sum of probability } X_i \text{ belongs to } c_1, c_2, .. c_k} = \frac{\pi_c \mathcal{N}(x_i \; ; \; \mu_c, \Sigma_c)}{\sum_{c'} \pi_{c'} \mathcal{N}(x_i \; ; \; \mu_{c'}, \Sigma_{c'})}$$

This value will be high when the point is assigned to the right cluster and lower otherwise.

M-step:

Post the E-step, we go back and update the Π, μ and Σ values. These are updated in the following manner:

The new density is defined by the ratio of the number of points in the cluster and the total number of points:

$$\Pi = \frac{\text{Number of points assigned to cluster}}{\text{Total number of points}}$$

The mean and the covariance matrix are updated based on the values assigned to the distribution, in proportion with the probability values for the data point. Hence, a data point that has a higher probability of being a part of that distribution will contribute a larger portion:

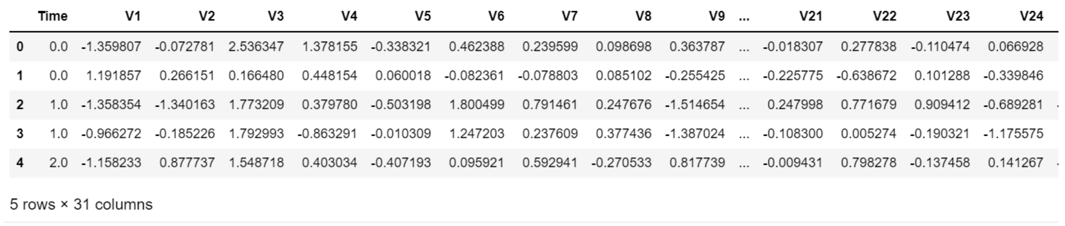
$$\mu = \frac{1}{\text{Number of points assigned to cluster}} \Sigma_i \; r_{ic} x_i$$

$$\Sigma_c = \frac{1}{\text{Number of points assigned to cluster}} \Sigma_i \; r_{ic} \; (x_i - \mu_c)^T \; (x_i - \mu_c)$$

Based on the updated values generated from this step, we calculate the new probabilities for each data point and update the values iteratively. This process is repeated in order to maximize the log-likelihood function.

1.4 Data set

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. The datasets contains transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example dependent cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.



	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267

5 rows × 31 columns

Figure 1.2: Dataset

2 Data Preprocessing

The task of data preprocessing is to organize the original business data with the new “business model” , clear those attributes irrelevant to the aim of data mining, supply clean, accurate, simplified data so as to improve the quality and efficiency of excavation under the guidance of domain knowledge. The data preprocessing mainly concludes data cleaning, integration, transformation and reduction. In this way, the dirty, incomplete and inconsistent data in real world can be corrected. The foremost process of data preprocessing is to import the data set by which we develop models. This will be performed using the pandas library. Prior to importing the dataset we are required to import the necessary libraries. The elimination of noise instances is one of the most difficult problems in inductive ML. Usually the removed instances have excessively deviating instances that have too many null feature values which are also referred to as outliers.

Data cleaning: By filling the null, smoothing noise data, identify and delete the isolated data, and solve inconsistency to attain the goal of clearing data.

Data integration: Save the data belonging to several data sources to a consistent data storage (such as data warehouse), these data sources may include several databases, data cubes or ordinary files.

Data conversion: Convert the data into one form that is suitable for excavation, for instance, zoom the attributive data in proportion, make it falls into a comparatively smaller specified zone.

Data reduction: The compressed data used to acquire the dataset is much smaller than the original data, but it still keeps its integrity. Thus, the data mining will have more effect on the condensed dataset, and produce the same analysis result. Data preprocessing is indispensable to data mining. Statistics suggests that data preprocessing takes up 60 percent of the time in a complete process of data mining.

Missing data handling is another issue often dealt with in the data preparation steps. Moreover, in real-world data, the representation of data often uses too many features, but only a few of them may be related to the target concept. There may be redundancy, where certain features are correlated so that is not necessary to include all of them in modeling; and interdependence, where two or more features between them convey important information that is obscure if any of them is included on its own. Feature subset selection is the process of identifying and removing as much irrelevant and redundant information as possible. This reduces the dimensionality of the data and may allow learning algorithms to operate faster and more effectively. Splitting the data set becomes one of the significant processes of data preprocessing. The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. It is a fast and easy procedure to perform, the results of which allow you to compare the performance of machine learning algorithms for your predictive modeling problem.

2.1 Fitting the Model and Predicting Results

With the gaussian mixture model we saw that for both non-fraudulent and fraudulent classes, features followed Gaussian distributions. For example, for V4, the non-fraudulent class followed a Gaussian distribution with small variance compared to the fraudulent class. Now let's evaluate the scatterplots between two features (i.e. V14 and V17, V11 and V4, etc...). We are trying to find what kind of Gaussian distributions we have (i.e. evaluate the covariances) so that we can estimate a Gaussian Mixture Model (GMM) using the Expectation-Maximization algorithm in order to represent the non-fraudulent class. Finally, detect the fraudulent samples by evaluating their probabilities (i.e. below a threshold).

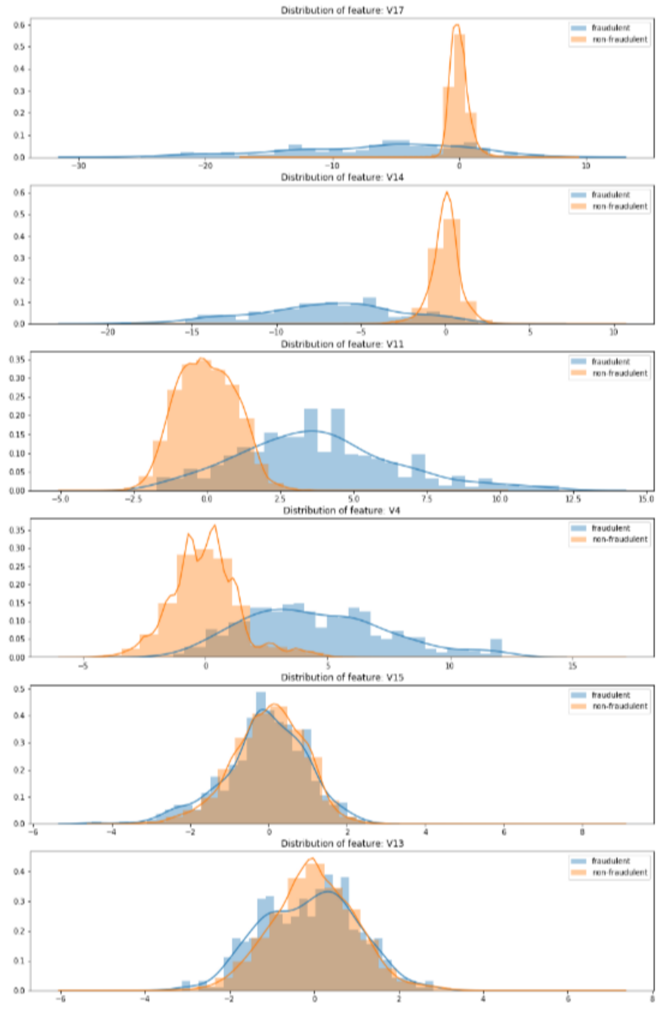


Figure 1.3 Distribution of Features

Now we will train a simple Gaussian mixture model using V14 and V17. We will create a dataset with only non-fraudulent transactions and a dataset with fraudulent ones Then we will plit non-fraudulent data in 90% for training GMM and 10% for cross-validation and testing Then we will split the fraudulent data in 50% for cross-validation (to find the probability threshold) and 50% for testing.

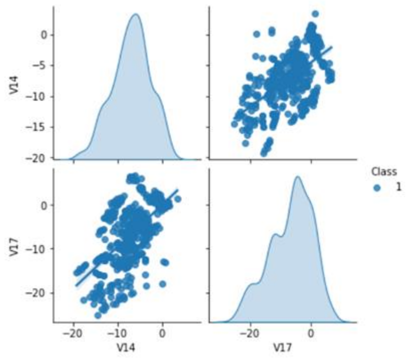


Figure 1.4 Model using 2 features

2.2 Visualization

The GMM is fit, let's find the probabilities of the test set. After we find those probabilities, if the probability is below a threshold we will say it is a fraudulent transaction (that is because our GMM is based on non-fraudulent transactions). Low probability means that is not probable that a given transaction is non-fraudulent.

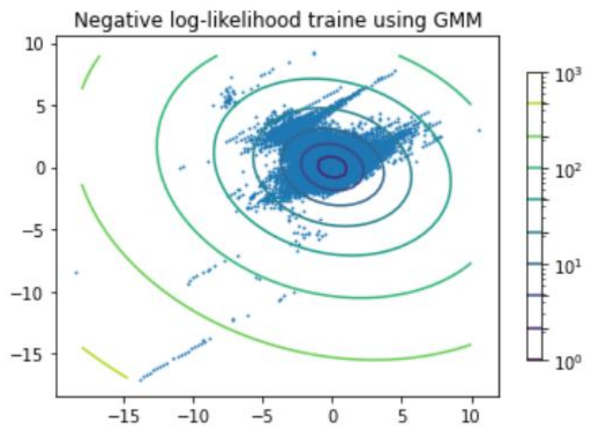


Figure 1.5 GMM Fit

We can see on the plot above that there are some pics of low probability. Those are probably outliers. To make it correctly, we should find the threshold T by doing cross validation. Now, just to have a first impression of how this works compared to the Logistic classifier from previous sections, we will define a fixed threshold and evaluate the performance.

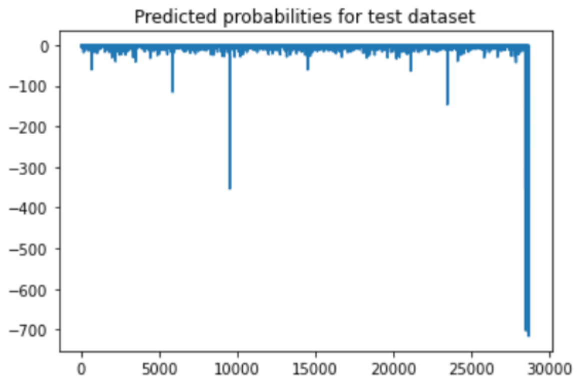


Figure 1.6 Probabilities of test dataset

That we have gone from an AUCPR of 0.47 using Logistic classifier to an AUCPR of 0.67. This will probably be improved more if we optimize the threshold T using cross-validation and we use more features, not just 'V14' and 'V17'.

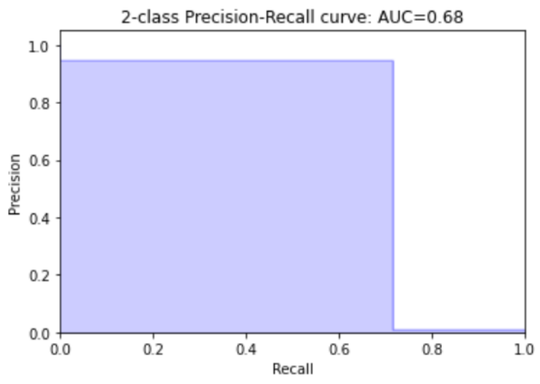


Figure 1.7 2 class precision

References

1. http://digilib.unhas.ac.id/uploaded_files/temporary/DigitalCollection/NDIxODU0YmUyNWEzY2U3M2MyMjY1NmE0MjA4NDYyNjFlMTI2YWU1Mw==.pdf
2. https://www.sciencedirect.com/science/article/pii/S1876610212004833
3. https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html
4. https://link.springer.com/chapter/10.1007/978-3-642-16167-4_41
5. https://ieeexplore.ieee.org/abstract/document/958974