

Objective:

In this session, you will learn to build SVM for classification problems

Key takeaways:

- Build SVM model for classification using different kernels
- Perform Grid Search for appropriate parameters for these kernels

Activity:**DatasetDetails**

Attribute	Description
ID	CustomerID
Age	Customer's age in completed years
Experience	#years of professional experience
Income	Annual income of the customer (\$000)
ZIPCode	Home Address ZIP code.
Family	Family size of the customer
CCAvg	Avg. spending on credit cards per month (\$000)
Education	Education Level. 1: Undergrad; 2: Graduate; 3: Advanced/Professional
Mortgage	Value of house mortgage if any. (\$000)
PersonalLoan	Did this customer accept the personal loan offered in the last campaign? (Target attribute)
SecuritiesAccount	Does the customer have a securities account with the bank?
CDAccount	Does the customer have a certificate of deposit (CD) account with the bank?
Online	Does the customer use internet banking facilities?
CreditCard	Does the customer use a credit card issued by Universal Bank?

1. Load Data into R:**2. Data preparation**

- to remove the columns ID & ZIP
- Convert categorical attribute "Education" to numeric
- Standardization of Data
- Split the data into train and test datasets

3. Model Building**#Classification using SVM**

```
install.packages("e1071")
```

```
library(e1071)
```

#Building the model on train data

```
x = subset(train_bankdata, select = -Personal.Loan) # remove response variable
```

```
y = as.factor(train_bankdata$Personal.Loan)
```

```
model = svm(x,y, method="C-classification", kernel="linear", cost =10, gamma=0.1)
summary(model) #Interpretation of summary
```

4. Applying the model on train data & test data and predict whether personal loan is taken or not

5. Build the confusion matrix

6. Compute the error metrics
perform a gridsearch

```
tuneResult<- tune(svm, train.x= x, train.y =y, ranges=list(gamma=10^(-6:-1), cost =2^(2:3)))
print(tuneResult)
tunedModel<-
tuneResult$best.model
tunedModelY<-
predict(tunedModel, as.matrix(x))
Conf<-
table(y, tunedModelY)
#you can now compute the metrics.
```

References:

<http://eeecon.uibk.ac.at/~zeileis/papers/Ensemble-2005.pdf>

https://escience.rpi.edu/data/DA/svmbasic_notes.pdf