Report for the Support Vector Machine Assignment

Dataset : GermanData

Motive: To deduce the best kernel applicable for a given code in R for a given dataset and also replicate the same in Orange

Observation :

R-studio

The assignment contains a given R file, which has a program as below. The program has been explained line wise as we proceed with the program in R below :

```
######### Please fill the ??? with proper description (atleast 130 charaters for each)
######### for SVM function try different values to achieve better results
# loading neccessary packages and dataset
install.packages("caret")
install.packages("e1701")
library(caret)
library(e1071)
data(GermanCredit)
dataset = GermanCredit
write.csv(dataset, file="Germancredit.csv")
# ???
```

We load the dataset using with the code above. I have added additional code (highlighted) to save the data in a csv to be used for prediction in Orange.

```
str(dataset)
dataset[,1:7] = as.data.frame(lapply(dataset[,1:7], scale))
str(dataset)
dataset
write.csv(dataset, file="dataset.csv")
# ???
```

From the dataset, we scale the first 7 rows which are Duration, Amount, Installment Rate Percentage, Residence Duration, Age, Number Existing Credits, Number People Maintenance using the "**dataset[,1:7] = as.data.frame(lapply(dataset[,1:7], scale))**" code. These 7 rows act as our independent variables which provide us data essential for prediction.

```
sample_index = sample(1000, 200)

test_dateset = dataset[sample_index,]

train_dateset = dataset[-sample_index,]

# ???
```

The sample command in R describes how the data should be sampled for every 1000 entries→ 200 is the sample. This means the test data constitutes to 20% of the total data, while training is 80%.

Model SVM and Tuning :

We are required to find the best kernel and also the corresponding value for cost and gamma. We follow the below steps to achieve this –

- ➔ We create a generic model (without dependency of choosing the type of kernel and its factors).

```
x<-subset(train_dateset, select = -Class)
y<- train_dateset$Class
model<-svm(x,y, scale=F)
```

- ➔ In order to get an unbiased output for each kernel, we take a large range of values for cost and gamma and tune to find their best values as below :

Radial

```
obj<-tune(svm, Class~.,kernel ="radial", data= train_dateset,
ranges=list(gamma=2^(-15:3), cost=2^(-5:15)))
summary(obj)
```

Linear

```
obj<-tune(svm, Class~.,kernel ="linear", data= train_dateset,
ranges=list(gamma=2^(-15:3), cost=2^(-5:15)))
summary(obj)
```

Polynomial

```
#obj<-tune(svm, Class~.,kernel ="polynomial", data= train_dateset,
ranges=list(gamma=2^(-15:3), cost=2^(-5:15)))
#summary(obj)
```

Sigmoid

```
#obj<-tune(svm, Class~.,kernel ="sigmoid", data= train_dateset,
ranges=list(gamma=2^(-15:3), cost=2^(-5:15)))
```

#summary(obj)

We find the best performance factor(Summary) for each of the kernels for our further analysis.

We see the results as below :

Radial :

Parameter tuning of 'svm':

sampling method: 10-fold cross validation

best parameters:

gamma	cost
0.0625	2

best performance Radial : 0.23125

Linear :

Parameter tuning of 'svm':

sampling method: 10-fold cross validation

best parameters:

cost
0.125

best performance Linear : 0.25375

Polynomial :

Parameter tuning of 'svm':

sampling method: 10-fold cross validation

best parameters:

gamma	cost
0.125	0.125

best performance Polynomial : 0.24625

Sigmoid :

Parameter tuning of 'svm':

sampling method: 10-fold cross validation

best parameters:

gamma cost
0.03125 2

best performance Sigmoid : 0.2425

Best Performance Factor and Modelling :

- ➔ The summary of each tuning will give us a **Best Performance** factor. The lower the Best Performance Factor implies higher precision. Hence we choose the kernel which has the lowest Best Performance value.
- ➔ We use the kernel type, gamma and cost values for this kernel and input it in the svm model creation code given in our assignment to predict and build a confusion matrix.

From our example, we understand that the Best Performance factor was lowest for kernel= radial with values for cost = and gamma=. We use this for modelling our SVM as below.

```
#model = svm(Class ~ ., kernel = ???, cost = ???, gamma = ???, data = trainm_dateset, scale = F)

# ???

model = svm(Class ~ ., kernel = "radial", cost = 2,gamma=0.0625, data = train_dateset, scale = F)
```

Prediction and Confusion Matrix:

```
predictions <-  predict(model, test_dateset[-10])

# ???

table(test_dateset[,10], predictions)
```

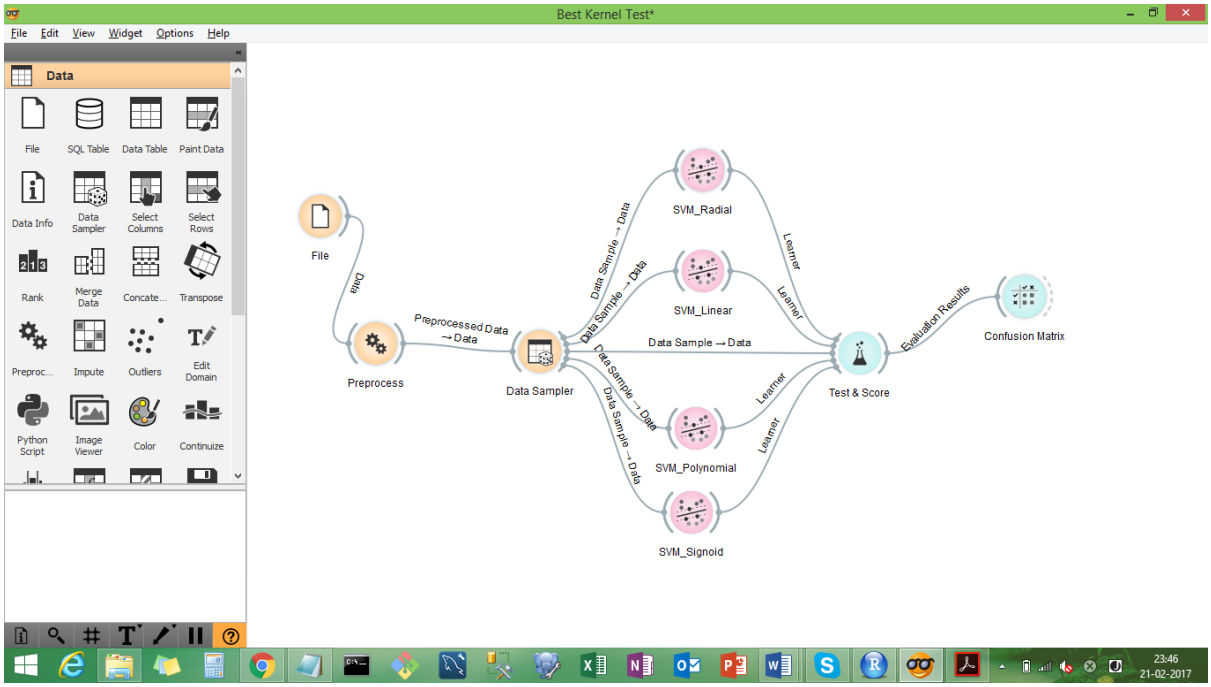
We predict the CONFUSION MATRIX using the above code and get the answer as below :

```
      predictions
       Bad Good
  Bad   32   30
  Good   7  131
```

Orange

We upload the raw dataset in Orange and proceed as below to deduce the best kernel applicable for our code :

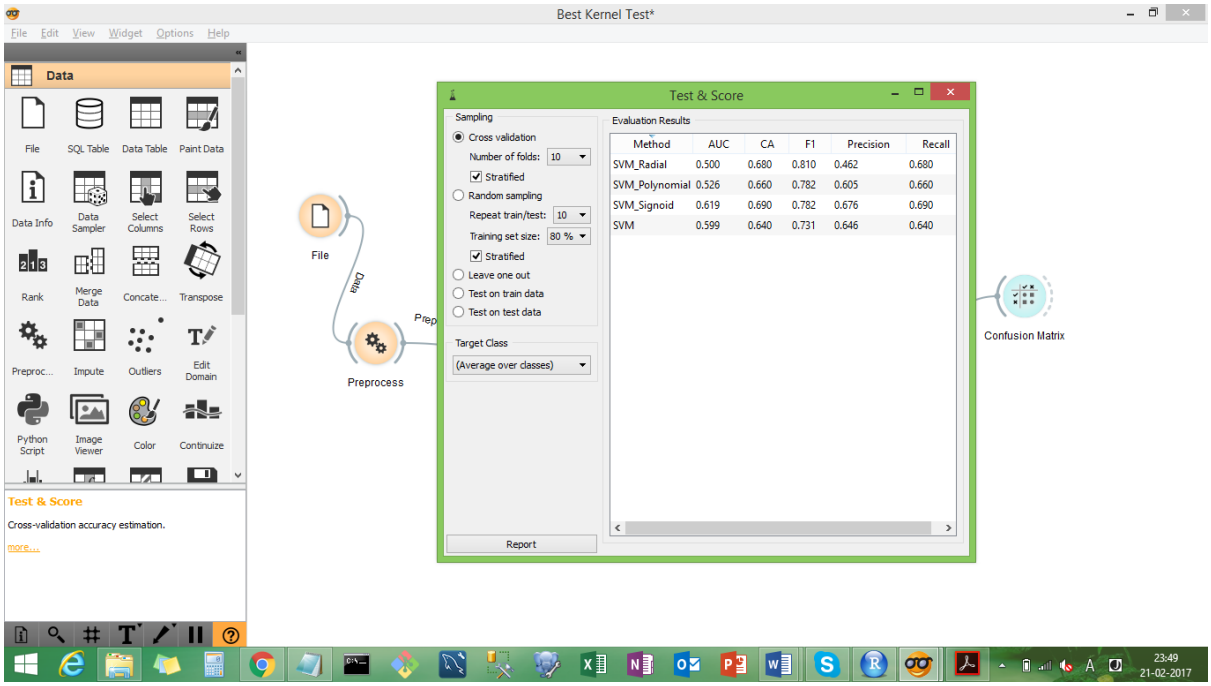
Deduce Best Kernel



We scale data from raw file to normalize the first 7 columns as we did in R using the Preprocess widget.

We sample data to 20% as mentioned in the code using Data Sampler widget.

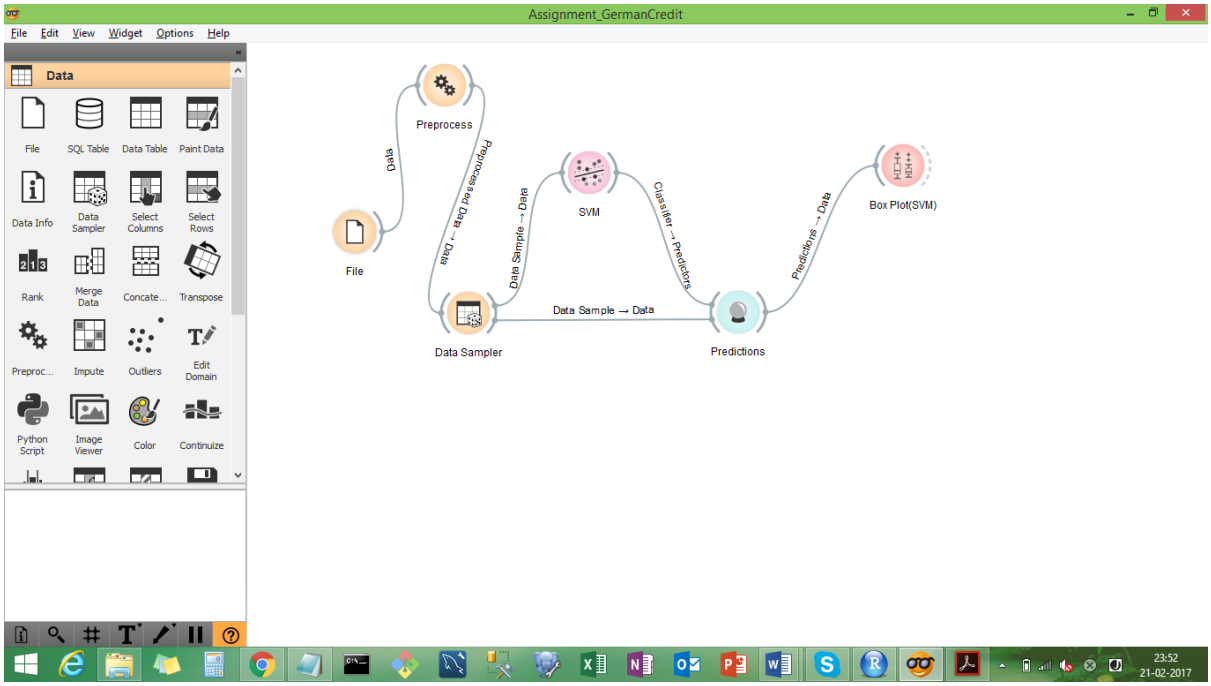
We connect the Data Sampler to the Test & Score widget directly and via SVMs.
I have input values for kernel and its respective parameters (as deduced in R) and seen the below in Test & Score.



The F1 value for Radial is maximum, which means the best kernel to be chosen is Radial from the rest of the F1 values for remaining kernels.

We can also view the data in confusion matrix for the kernel.

Predict for the Best Kernel – Radial :



We input value for radial and its corresponding factors in the SVM and are also able to see the confusion matrix output on the Box Plot when linked to Predictions.

Conclusion :

The Best kernel can be chosen by tuning and then its corresponding values provide us the better prediction for a given range of Gamma and Cost. Orange provides us visually appealing data and R is more concise.