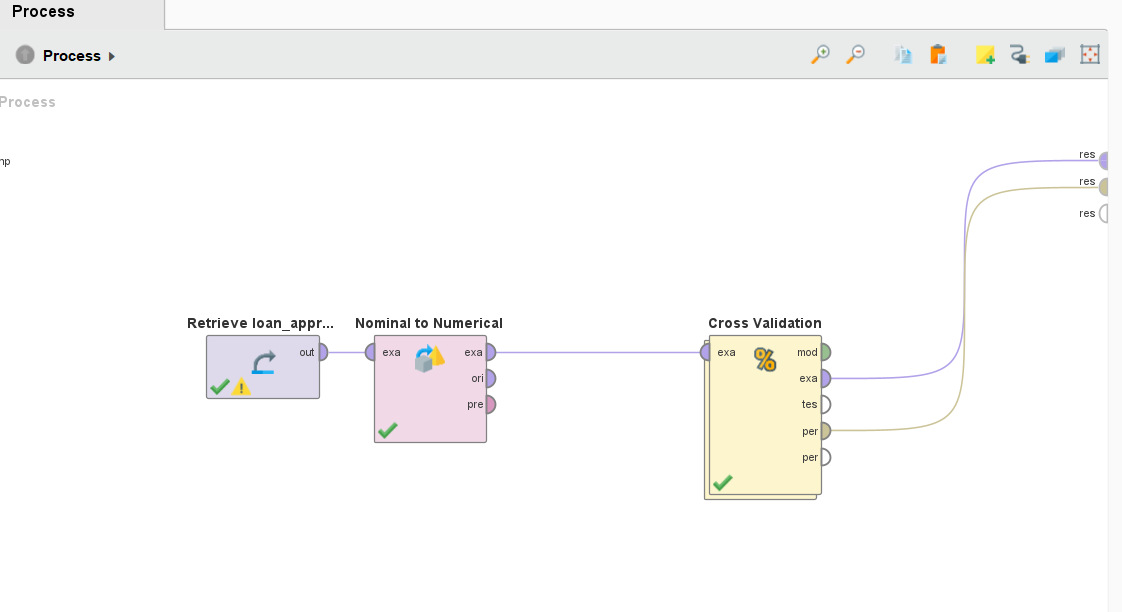
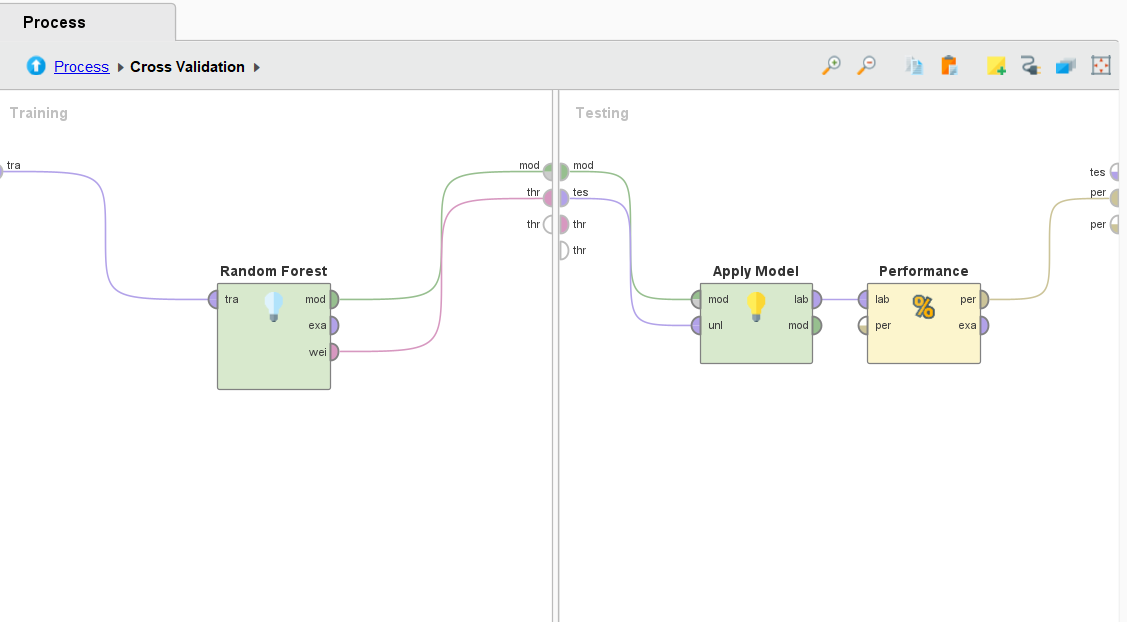
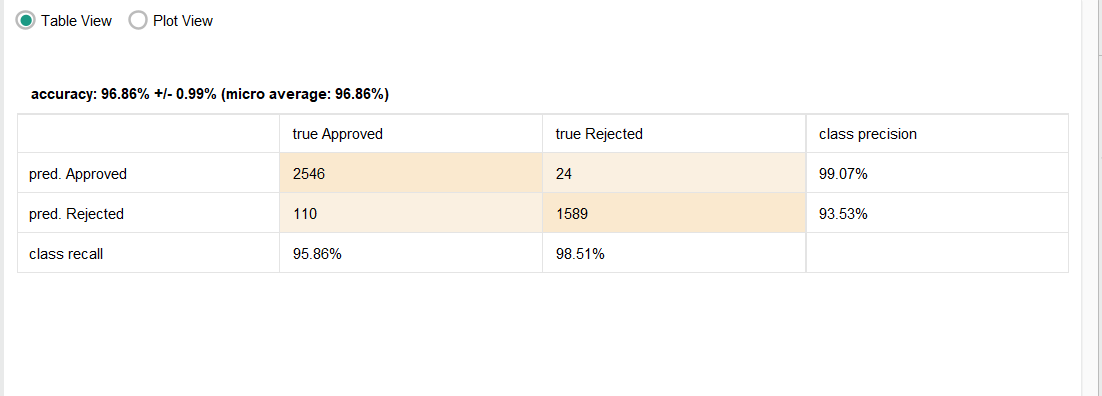
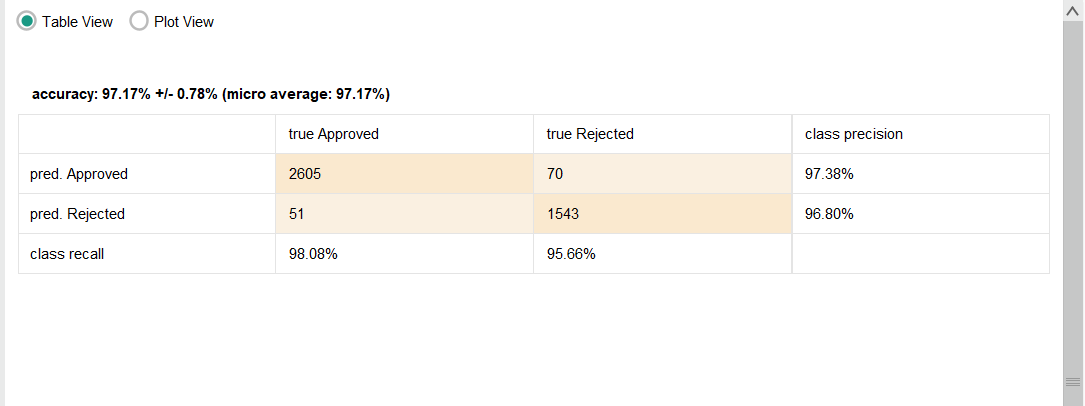
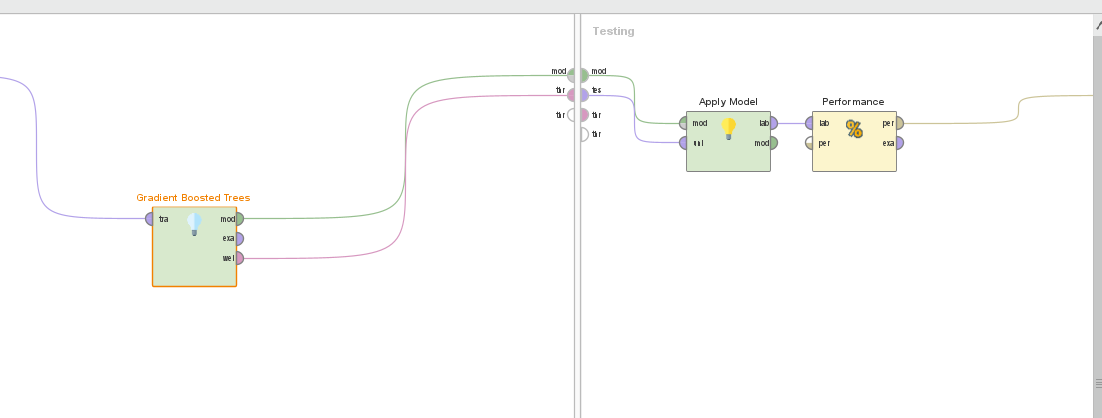
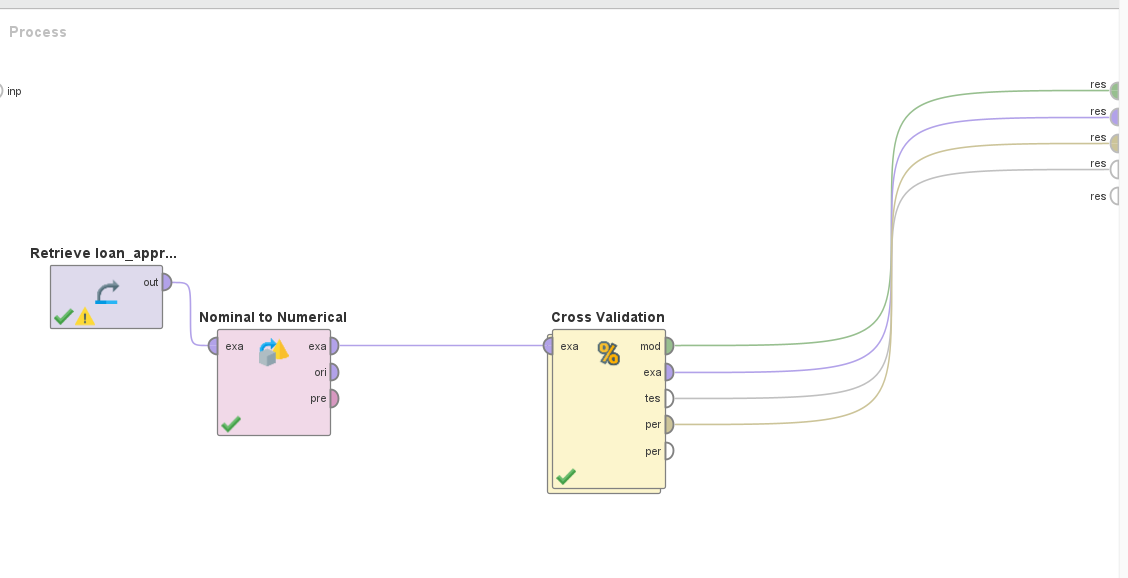
HW-3 ( KARTIK KANOTRA)

1. I am trying to find whether we can predict the Approval or Rejection of a loan based on the details of the person requesting it.
2. Data Source : I have taken a database listed by Archit Sharma in github used to solve a similar problem. It has 11 features and over 4000 records which covers most of the cases.
3. The dataset had over 4000 records and features like loan amount , luxury assets value had a lot of variance so to better represent it for the training algorithm , I normalized the dataset .  
   One more preprocessing step which was needed was converting binomial values to binary bits. This was necessary since many of the training algorithms don’t accept binomial and polynomial values.

D , E )   
**Random Forest**  


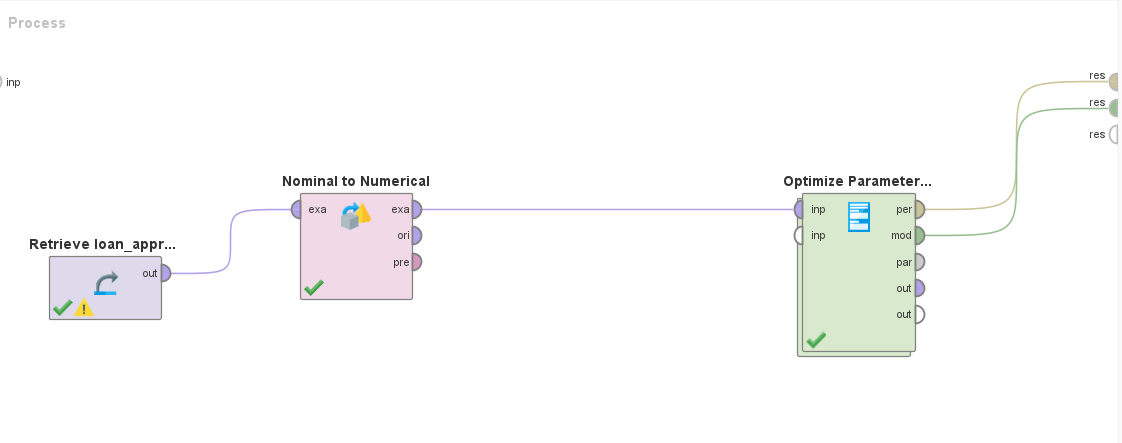


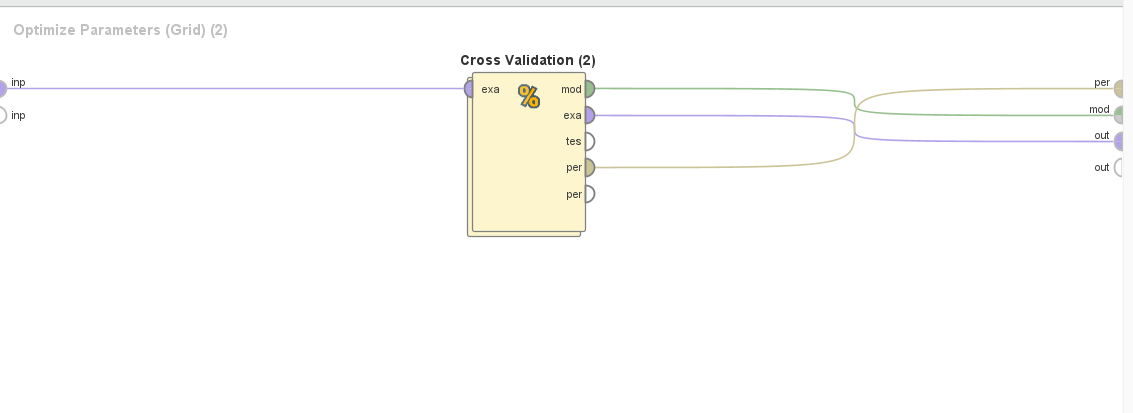
**Pros of Random Forest**: High Accuracy , Robust to Overfitting  
**Cons of Random Forest**: High Memory Usage , Interpretibility  
**Hyperparameters:** Number of Trees , Maximum Depth ,Class weight , criterion

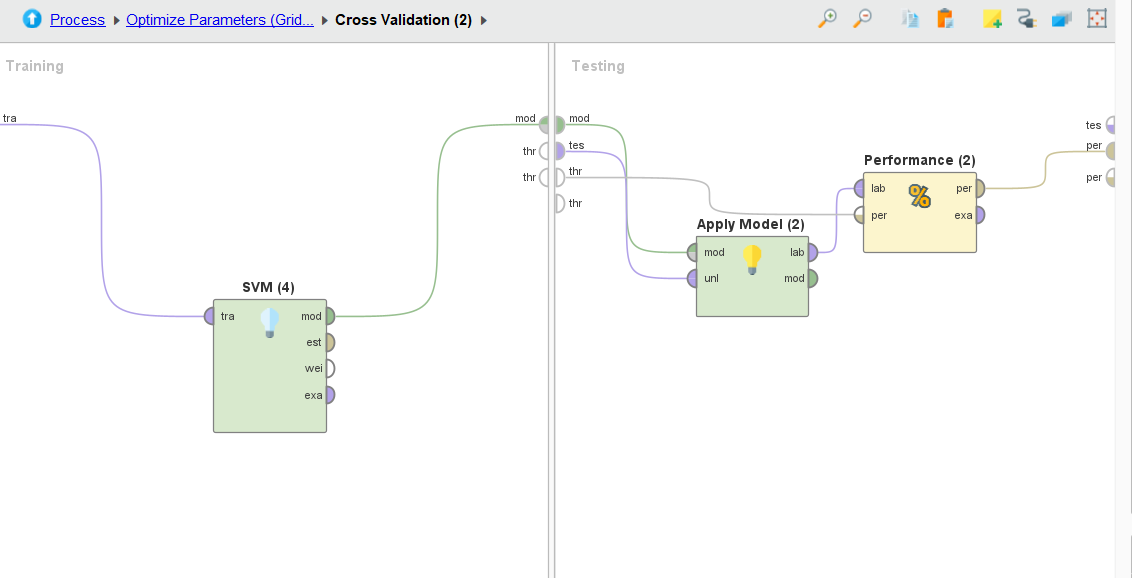
**Gradiant Boosted :**  
  


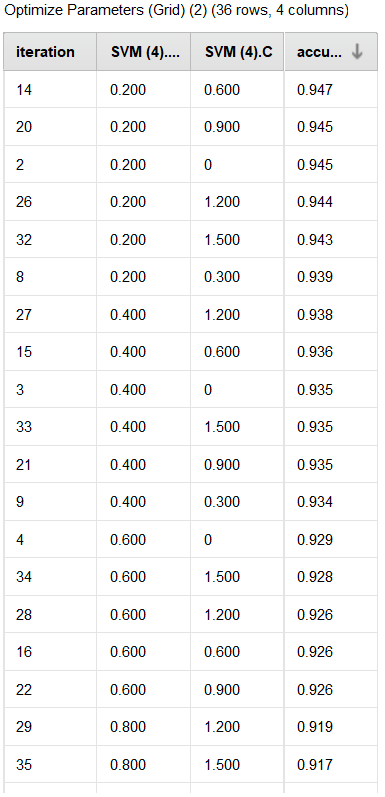
**Pros of Gradient Boosting**: High performance , Flexibility, Robust to Outliers  
**Cons of Gradient Boosting**: Training time , Overfitting , Memory Usage  
**Hyperparameters**: Learning Rate , Max Depth , loss , random State

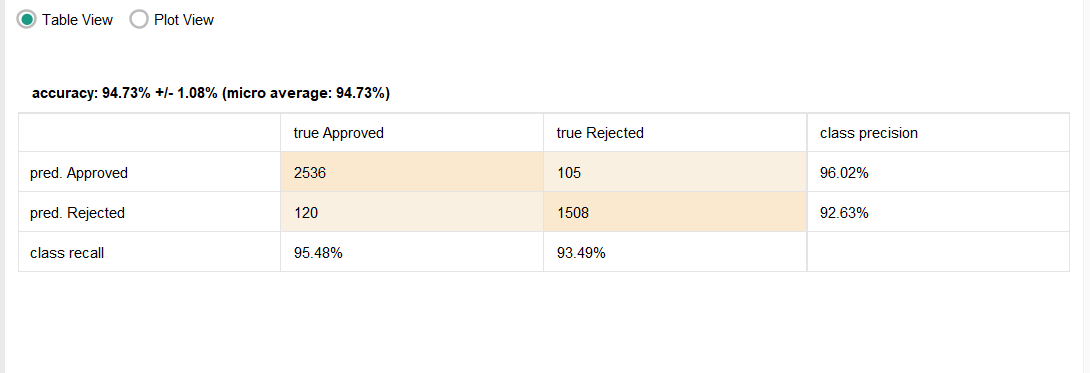
**SVM** ( With Hyperparameter Tuning)

****







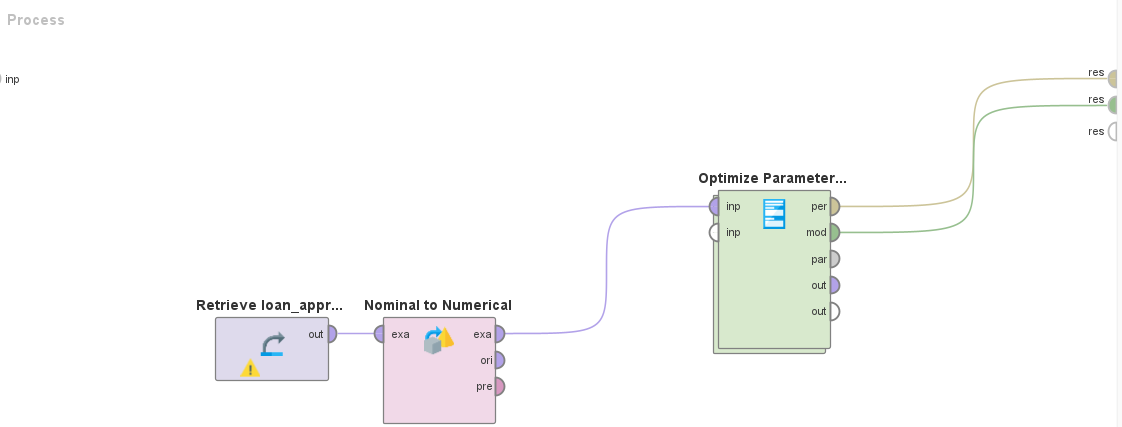


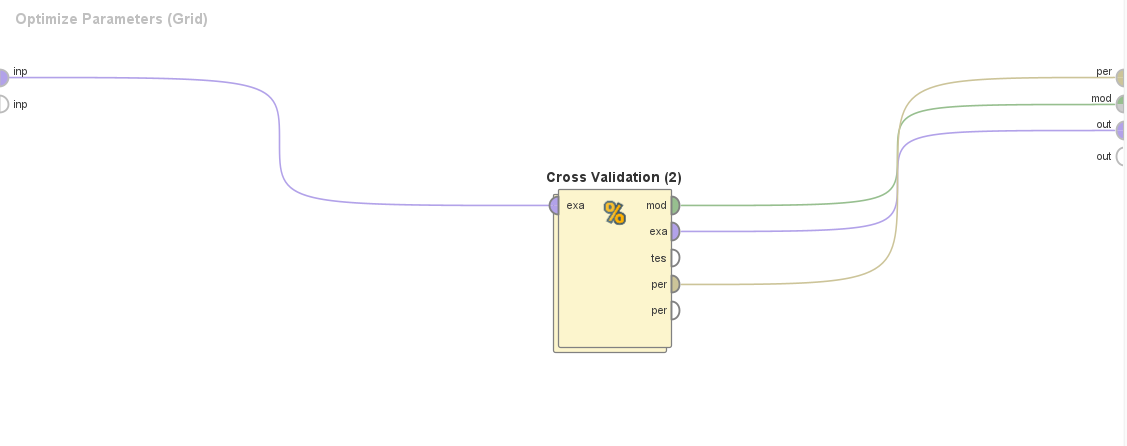
**Hyperparameters considered for tuning** : Kernal Gamma ,C

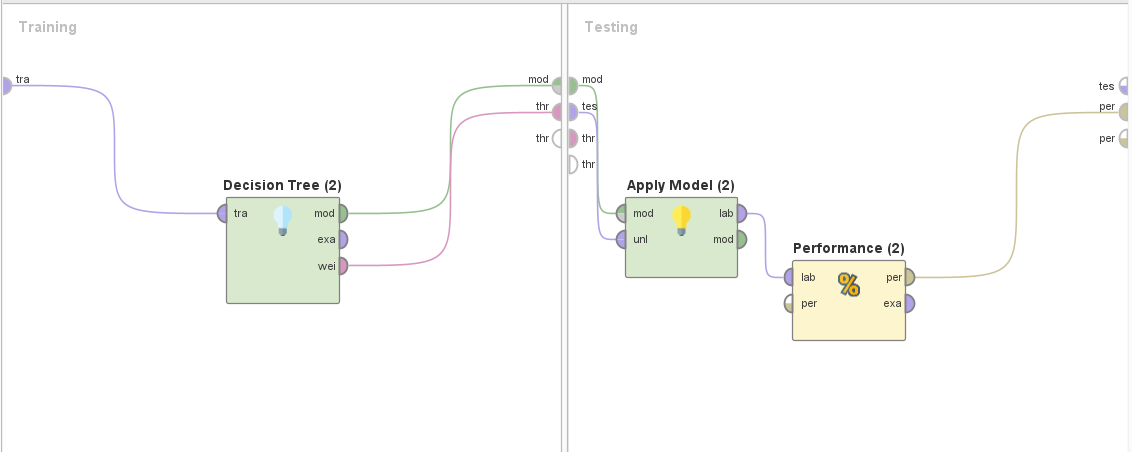
Used grid search in a specific ranges for both these parameters ,got maximum accuracy for gamma=0.2 and C=0.6.

**Pros for SVM**: Effectiveness in Higher Dimentions ,Robustness , Memory Efficiency  
**Cons For SVM** : Kernal Choice , Training Time , Parameter Selection

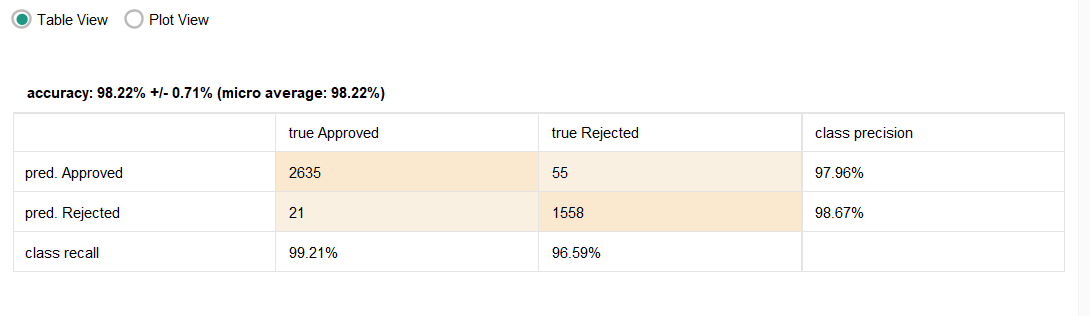
**Decision Tree** ( With Hyperparameter Optimization)











**Hyperparameters considered for tuning:**  Maximal Depth , Minimal Size for split. Used grid search in a specific ranges for both these parameters ,got maximum accuracy for Maximal depth=0 and minimal size for split=12.  
  
**Pros for Decision Tree**: interpretability , feature selection , simple to understand and implement.  
  
**Cons for Decision Tree**: Overfitting , Instability , Optimization

**KNN:**

**Code:**

<https://github.com/kartik0649/PA_HW>  
  
**Pros of KNN**: Simplicity , Versatility , Adaptability   
**Cons of KNN**: High Memory Requirement , Scaling of data , Sensitive to Noisy data  
**Hyperparameters:** Number of Neighbours , Weights , Leaf size  
  
  
  
  
  
F.) **Results:   
I** got the best result with the accuracy of **98.2 %**  using **Decision tree**

and which was able to predict with great confidence whether the loan would be approved or not using details of the person requesting the loan.