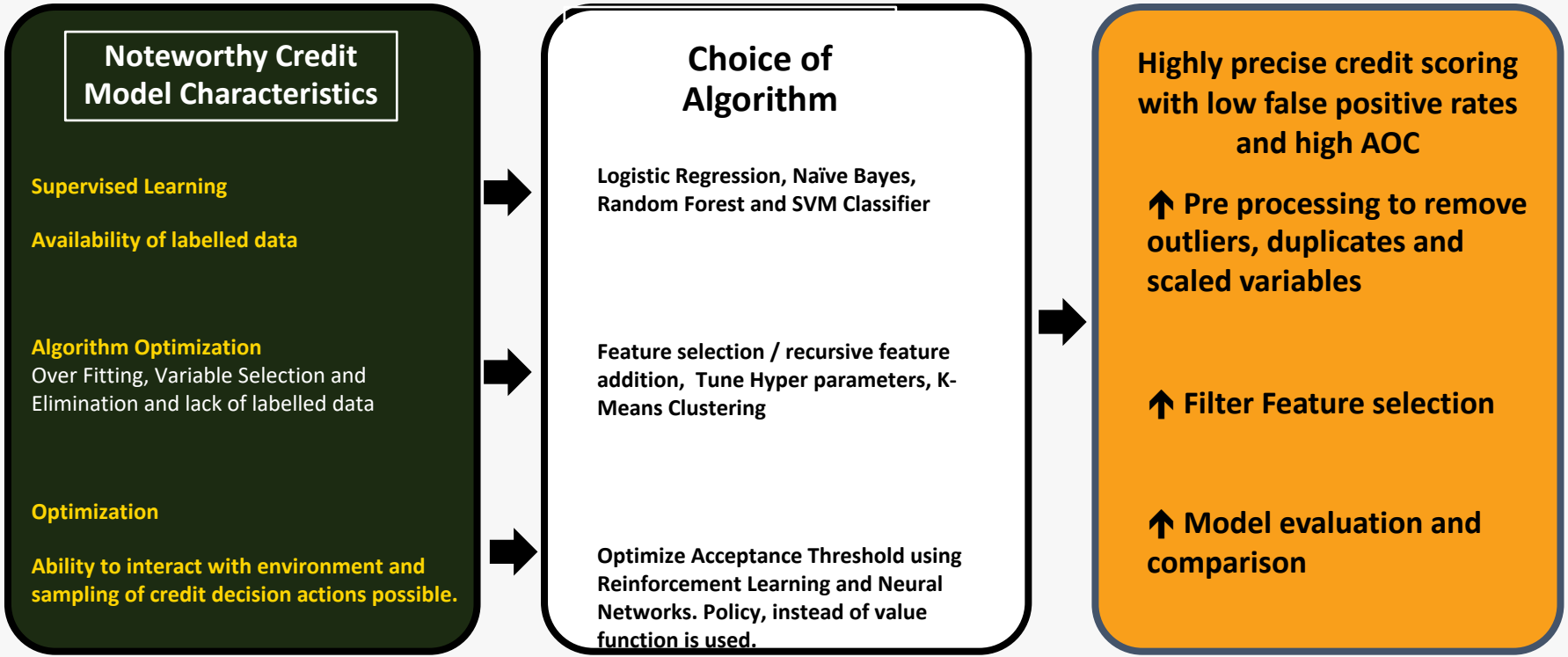
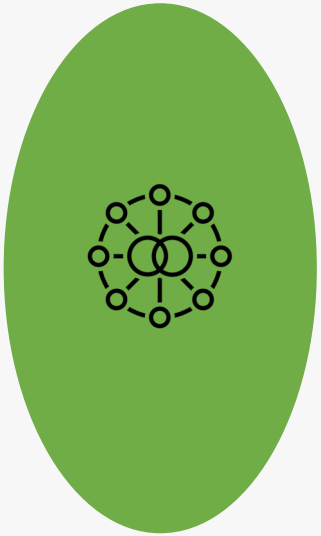
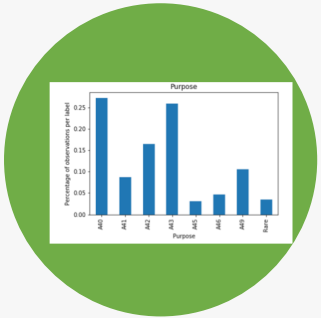
Evaluation of Consumer Credit scoring Strategies using Neural Network and Reinforcement Learning



Background

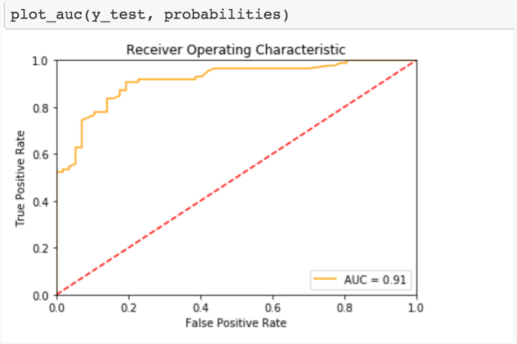


Tools Used in this project



Methodology

Data Pre-Processing, modelling, classification , tuning and optimization using RL 21 features, risk encoding, equal binning part of pre-processing. 80-20 split. Between train and test dataset



Classification and Optimization Models

skearn library to build Decision Tree, Random forest and SVM. K-means clustering (non supervised). CheckingAccountStatus, Duration and Age – most important features.

950 observations, 22 features
iPython Console, Profiler, Debugger, Pandas, Matplotlib, NumPy, Scilearn

Random forest and Boost with highest accuracy (lowest misclassification rate)

CV Score of 80.49% and Dev set score - 83.52%

Agent class incorporates the RL - functionality to interact with the environment passing actions sampled using a value function model instance

Powered by customer level account and transaction data

Neural Networks and Reinforcement learning bring the best predictive performance of default

```
# cross validation
print("Cross Validation Score: {:.2%}".format(np.mean(cross_val_score(rfc, X_train, Y_train, cv=10))))
```

Version 1 of program focused on simple logistic regression, Naïve Bayes and Trees/Random Forest

- **Logistic Regression – 76.22% accuracy**
- **Gaussian Naïve Bayes – 76.22%**
- **Random Forest – 79.72%**
- **Trees Boost – 80.41%**
- **CV Score of 80.49% for RF.**

```
KNN classifier Accuracy: 0.7622377622377622
Random Forest Classifier Accuracy: 0.7972027972027972
Accuracy of SVM classifier: 0.7412587412587412
Decision Tree Classifier Accuracy: 0.6993006993006993
GNB Accuracy: 0.7622377622377622
LDA Accuracy: 0.7762237762237763
Ridge Accuracy: 0.7622377622377622
Lasso Accuracy: 0.7622377622377622
```

```
print('Test all features RFC ROC AUC=%f' % (auc_score_all))
print('Test selected features ROC AUC=%f' % (auc_score_final2))
print ('Model ROC AUC change is',(auc_score_final2-auc_score_all))
```

```
                    Confusion matrix
                    Score positive    Score negative
Actual positive            46                11
Actual negative            11                75

Accuracy  0.85

             Positive      Negative
Num case        57            86
Precision     0.81          0.87
Recall        0.81          0.87
F1            0.81          0.87
```

* Source – CREDIT SCORING WITH A FEATURE SELECTION APPROACH, DEEP LEARNING https://cyberleninka.org,

Version 2 – Feature selection and Optimization

RF ROC AUC for all features = 87.08%

Test iteratively and see which feature increases ROC AUC - moved from 22 features to 4 features, still resulted in high ROC AUC of 90.932% (training) and 89.90% (test)

Best RF performance with max depth = 180 and number of trees = 600 91.42%

Version 3 – RL

Neural Network – Prediction highest with 83.93%

100 episodes of training

*** Ongoing work to simulate learning episodes and find V-Optimal for credit threshold rates**

optimized_value = self.get_q_table().max()