Capstone project

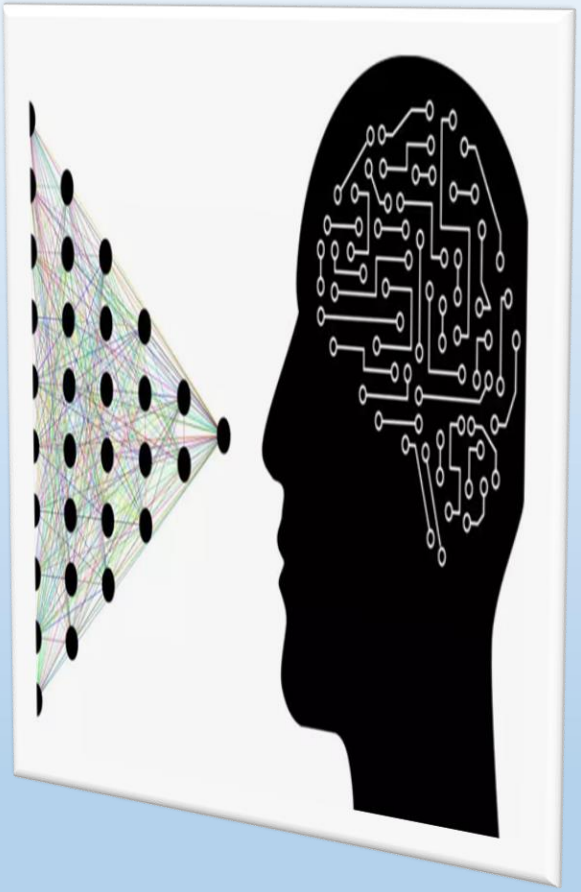
UNVEILING INSIGHTS AND ENHANCING
DECISION-MAKING WITH DATA SCIENCE

A Loan Default Prediction System

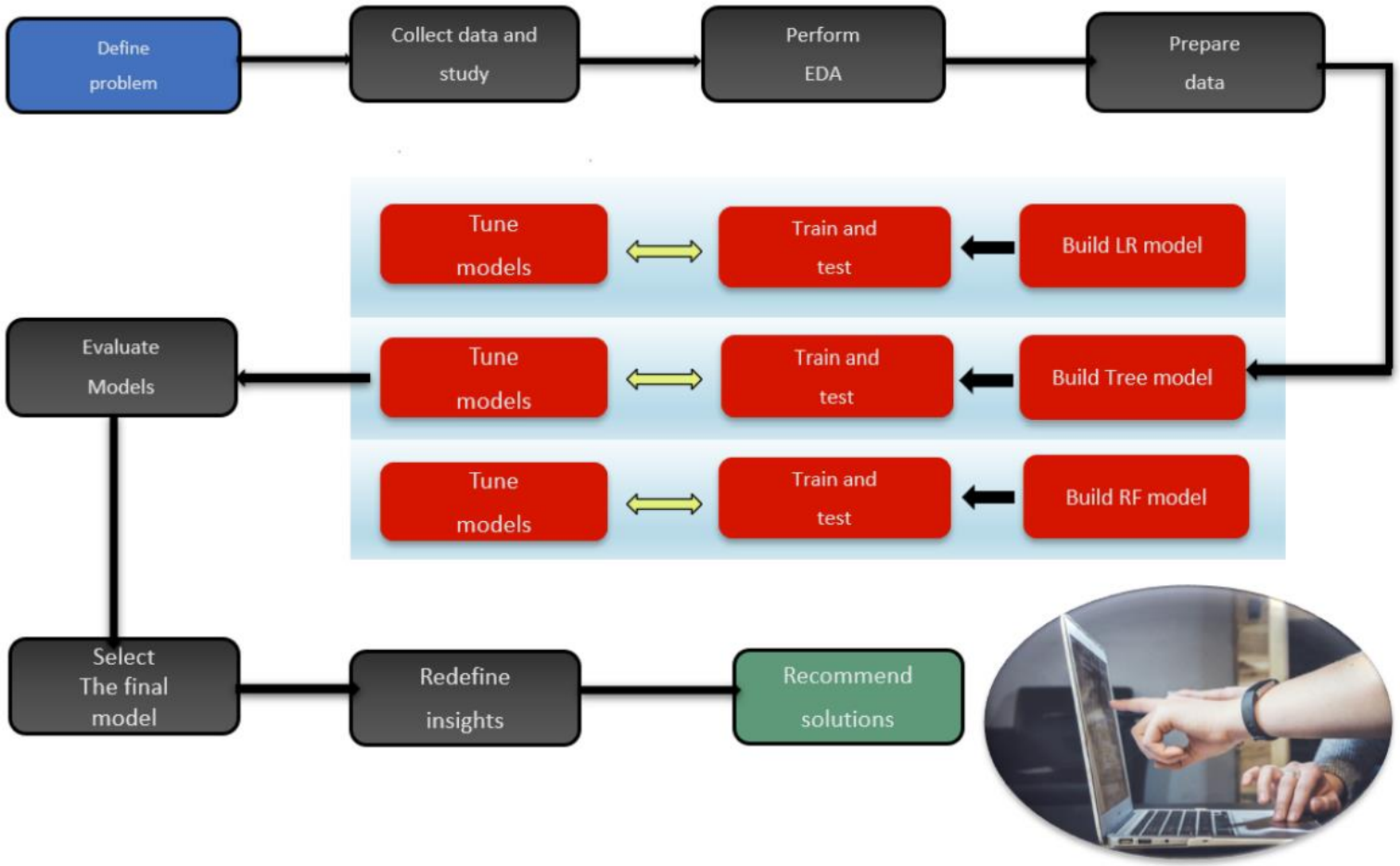
Author: Sudhir Behera

Introduction:

With the advent of ***data science*** and **machine learning** techniques, the emphasis has now shifted toward developing machines capable of learning the *loan* ***approval*** *process*. The aim is to eliminate biases and enhance efficiency. However, it is crucial to ensure that these machines do not inherit the ***biases*** that were previously present due to human approval processes.



Loan Default Prediction



Problem Statement

A bank's consumer credit department aims to simplify the decision-making process for home equity lines of credit to be accepted. To do this,

- they will adopt the Equal Credit Opportunity Act's guidelines to establish an empirically derived and statistically sound model for credit scoring.
- the model will be based on the data obtained via the existing loan underwriting process from recent applicants who have been given credit.
- the model will be built from predictive modeling techniques, but the
- model created must be interpretable enough to provide a justification for any adverse behavior (rejections).

Context

A significant portion of retail bank profits is derived from the interest earned on home loans, which are typically borrowed by customers with regular income or high earnings. One of the biggest **concerns** for banks is the ***occurrence*** of defaulters, as "***bad***" loans, commonly known as non-performing assets (NPAs), can significantly erode their profits. Thus, it is crucial for banks to exercise prudence when approving loans for their customer base.

Main article: Financial crisis of 2007–08 Wikipedia

The International Monetary Fund estimated that large U.S. and European banks lost more than $1 trillion on toxic assets and from bad loans from January 2007 to September 2009. These losses were expected to top $2.8 trillion from 2007 to 2010. U.S. banks losses were forecast to hit $1 trillion and European bank losses will reach $1.6 trillion. The IMF estimated that U.S. banks were about 60 through their losses, but British and eurozone banks only 40%.[338]

Objective

•Utilizing advanced Machine Learning techniques like ***logistic regression*** and ***tree-based algorithms*** to construct an optimal customer profile that predicts the likelihood of loan default based on the given *Home Equity dataset*(HMEQ).

The key questions:

- Can you identify
  - patterns and relationships and
  - key indicators that contribute to loan defaults?.
- What *insights* you can and and *solutions* you can recommend to the bank to ***minimize*** *the risk of future loan defaults*?.

Methodology:

Three data science techniques were employed in this project.

1. Logistic regression(LR).
2. Decision tree (Dtree) and
3. Random forest(RF)
   machine learning models.

These techniques are used due to their success in predicting or solving classification problem.

EDA

- Performed Data Overview
- Performed Univariate analysis
- Performed Bivariate analysis
- Multivariate Analysis
- Solved Outliers
- Treated Missing Values
- Feature selection

Data Collection and Exploration(EDA)

•A bank has captured information on past approved credits in the given *Home Equity dataset*(HMEQ).

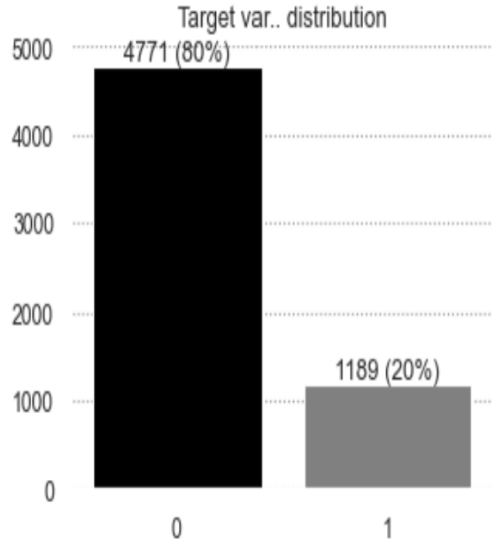
• There were 12 *features* registered for each customer.

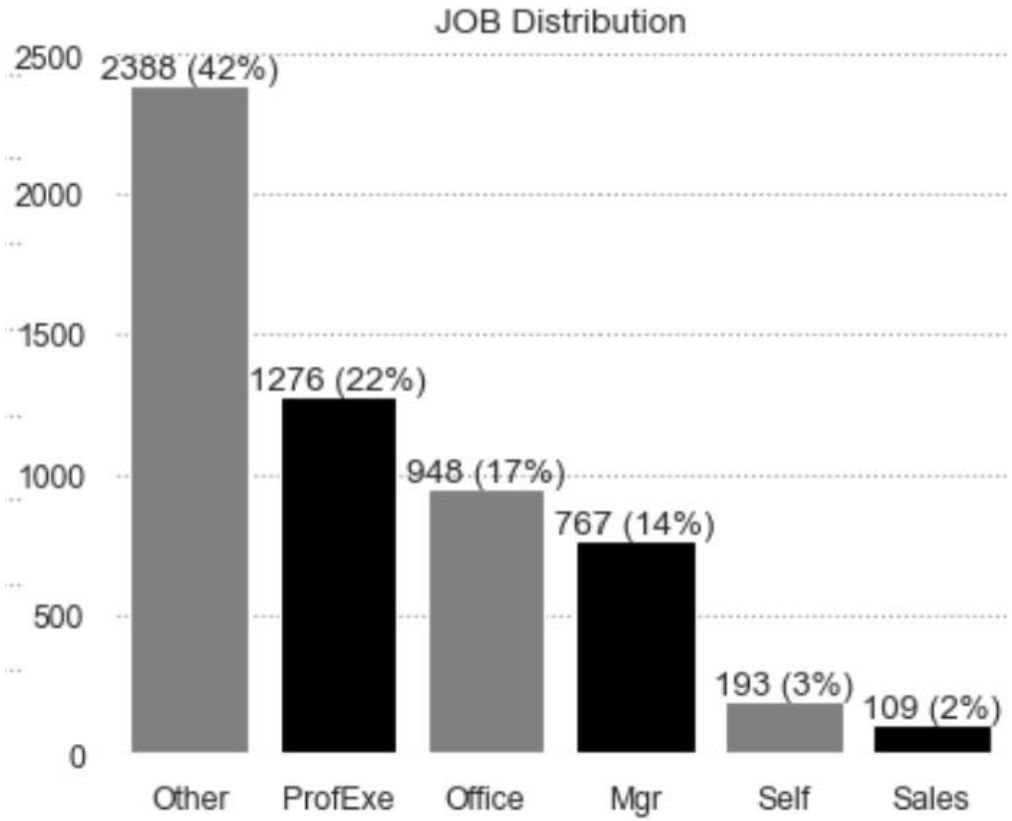
• **BAD**: is the target variable
"0" for repaid the loan and
"1" for defaulted on loan.

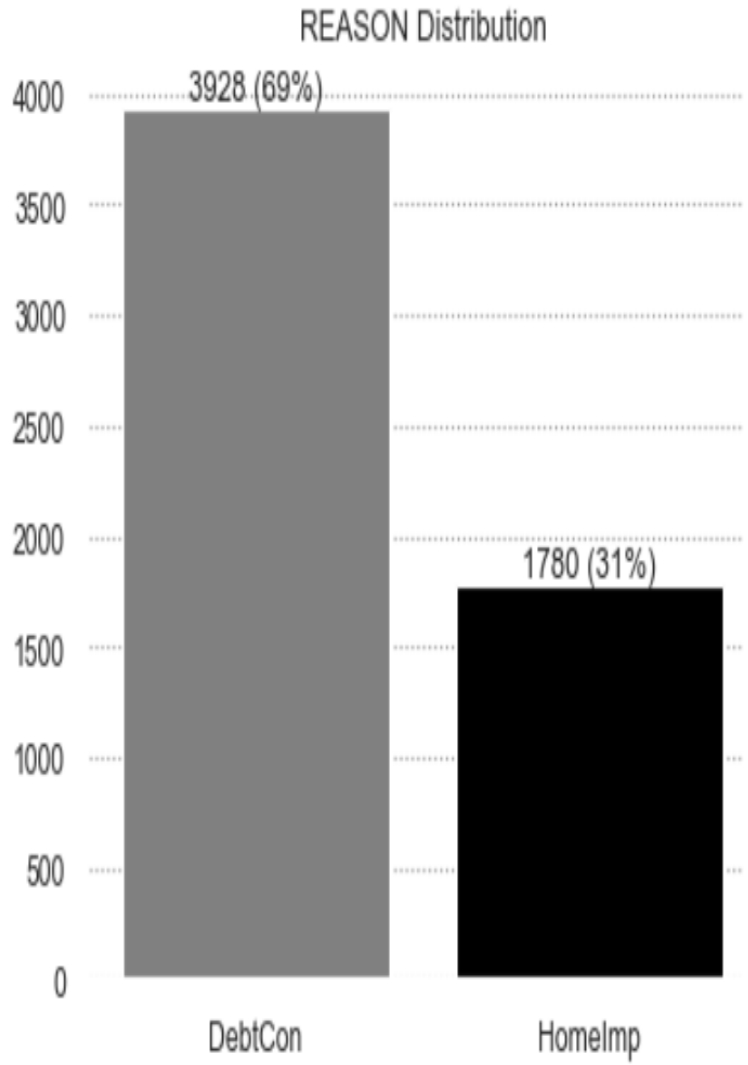
• LOAN, MORTDUE, VALUE, YOJ, DEROG, DELINQ, CLAGE, NINQ, CLNO, DEBTINC, REASON and JOB are predictor variables..

There are some takeaways from EDA

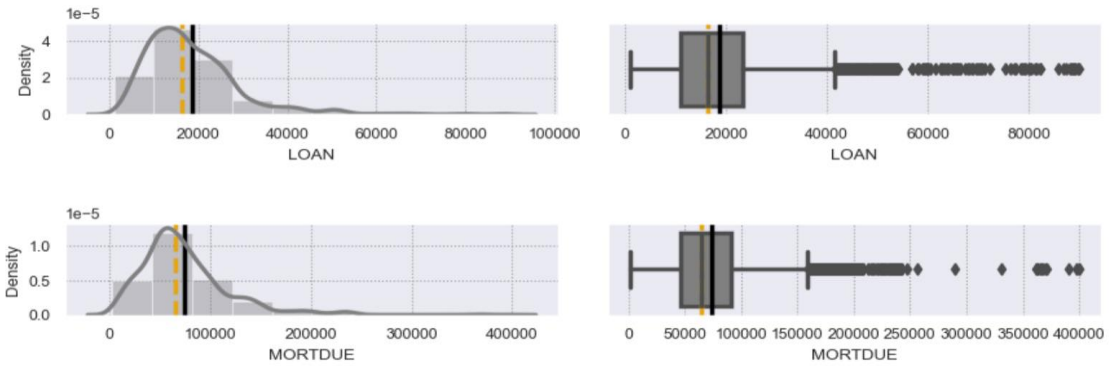
Categorical features distribution

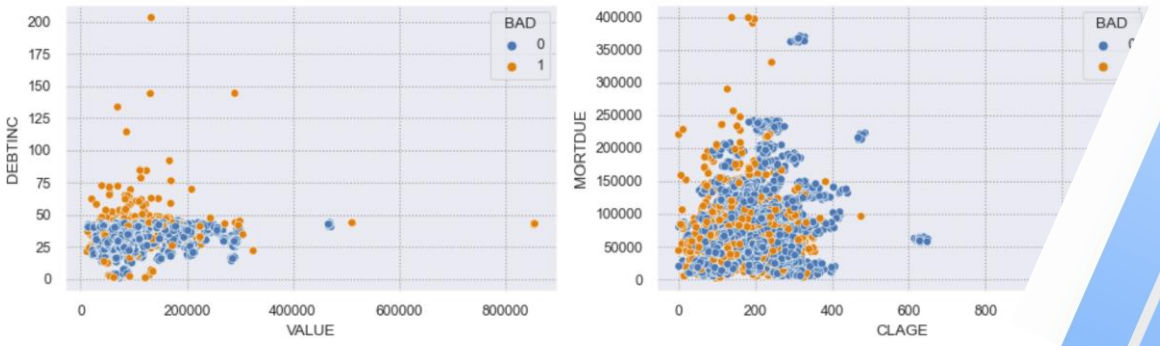


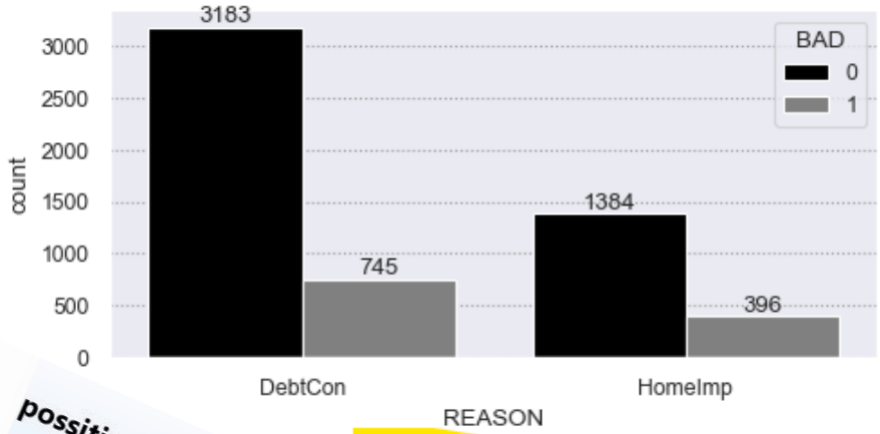




Numerical features
distribution







DebtCon default rate, 19 %

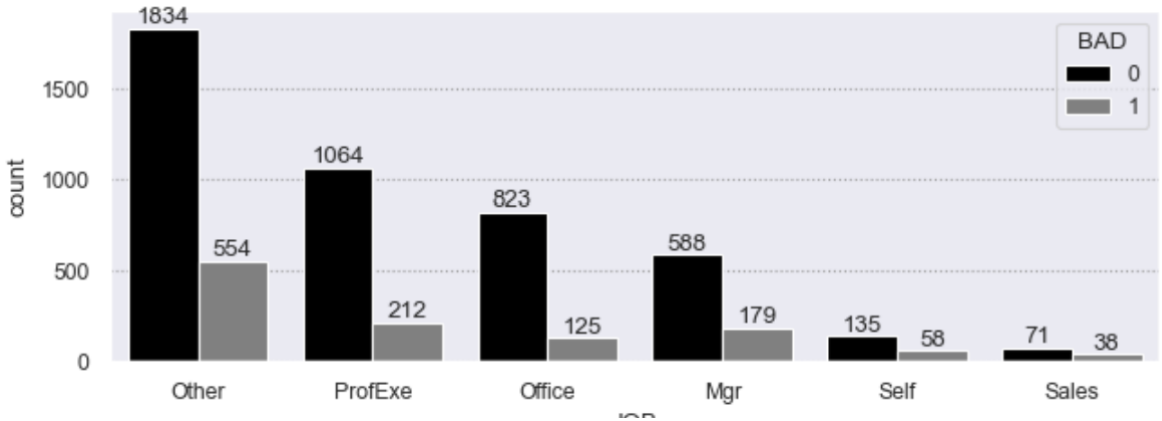
HomeImp default rate, 22 %

possitive correlations

- VALUE **and** MORTDUE
- DELINQ **and** BAD
- VALUE **and** LOAN
- MORTDUE **and** CLNO
- DEROG **and** BAD
- CLNO **and** CLAGE
- CLNO **and** VALUE
- DEBTINC **and** BAD

negative correlations

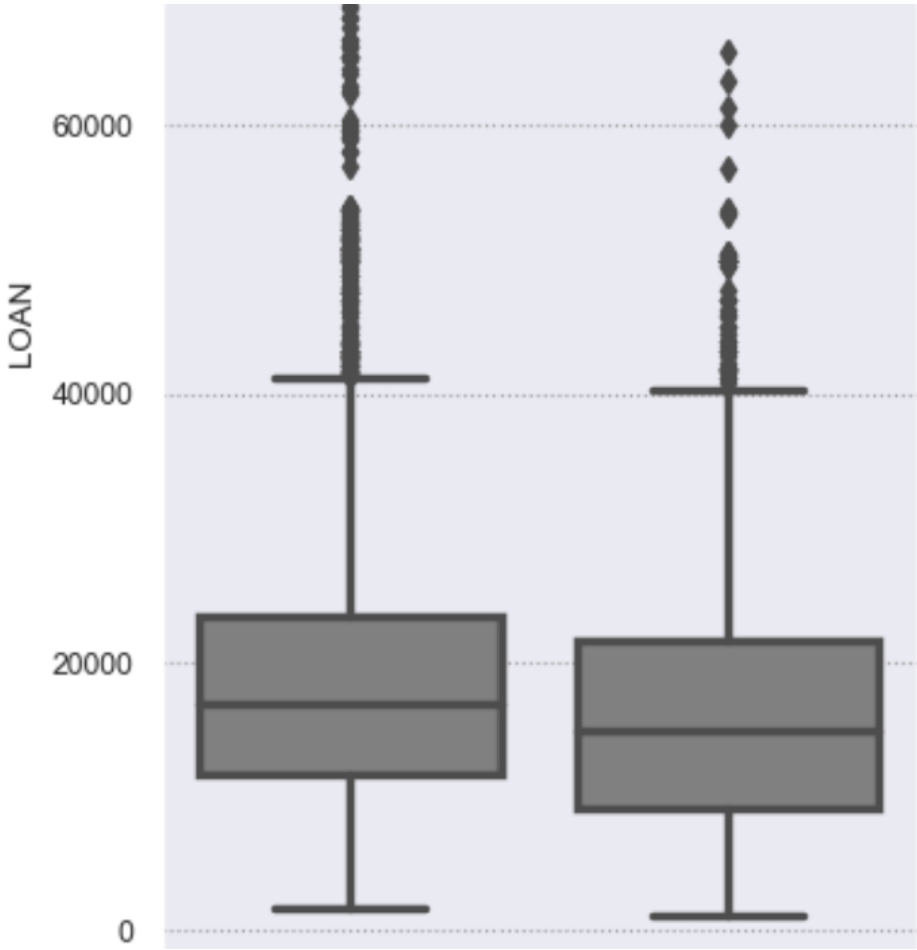
- CLAGE **and** BAD
- CLAGE **and** NINQ
- DEBTINC **and** YOJ
- MORTDUE **and** YOJ
- LOAN **and** BAD
- DEBTINC **and** CLAGE
- VALUE **and** BAD
- MORTDUE **and** BAD




```
Other: default rate, 23 %
ProfExe: default rate, 17 %
Office: default rate, 13 %
Mgr: default rate, 23 %
Self: default rate, 30 %
Sales: default rate, 35 %
```

Observations

- Customer with ***Office*** job have lowest default rate.
- Customer with ***Sales*** job have the highest default rate.



- It seems there is very slight *negaive correlation* of aprox -0.08 exist between LOAN and BAD variable. Which means custmers with larger loan amount ***slightly*** do better in repaying the loan.
- Also, from the boxplot you could see that ***mean*** of class 0 is higer than the mean of class 1. Which reaffirms larger loan amount performs better on loan replayment.

Feature selection using Pearson p-value technique.

	Pearson Corr.	p-value
LOAN	-0.1088	0.0000
MORTDUE	-0.0806	0.0000
VALUE	-0.0831	0.0000
YOJ	-0.0584	0.0000
DEROG	0.2641	0.0000
DELINQ	0.3415	0.0000
CLAGE	-0.1761	0.0000
NINQ	0.1369	0.0000
CLNO	-0.0302	0.0197
DEBTINC	0.0815	0.0000

Chi-Square test

target	0	1
REASON		
DebtCon	3387	793
HomeImp	1384	396

p-value: 0.00422343653605062804845 54

target	0	1
JOB		
Mgr	588	179
Office	823	125
Other	2090	577
ProfExe	1064	212
Sales	71	38
Self	135	58

p-value: 0.000000000000001643693816 57

missing value treatment

Before

```
'LOAN': number of missing values '256' ---> '4.295%'
'MORTDUE': number of missing values '752' ---> '12.617%'
'VALUE': number of missing values '432' ---> '7.248%'
'REASON': number of missing values '252' ---> '4.228%'
'JOB': number of missing values '279' ---> '4.681%'
'YOJ': number of missing values '606' ---> '10.168%'
'DEROG': number of missing values '708' ---> '11.879%'
'DELINQ': number of missing values '580' ---> '9.732%'
'CLAGE': number of missing values '355' ---> '5.956%'
'NINQ': number of missing values '687' ---> '11.527%'
'CLNO': number of missing values '441' ---> '7.399%'
'DEBTINC': number of missing values '1361' ---> '22.836%'
```

After

```
'BAD': number of missing values '0' ---> '0.000%'
'LOAN': number of missing values '0' ---> '0.000%'
'MORTDUE': number of missing values '0' ---> '0.000%'
'VALUE': number of missing values '0' ---> '0.000%'
'REASON': number of missing values '0' ---> '0.000%'
'JOB': number of missing values '0' ---> '0.000%'
'YOJ': number of missing values '0' ---> '0.000%'
'DEROG': number of missing values '0' ---> '0.000%'
'DELINQ': number of missing values '0' ---> '0.000%'
'CLAGE': number of missing values '0' ---> '0.000%'
'NINQ': number of missing values '0' ---> '0.000%'
'CLNO': number of missing values '0' ---> '0.000%'
'DEBTINC': number of missing values '0' ---> '0.000%'
```

Important Insights from EDA

Observations

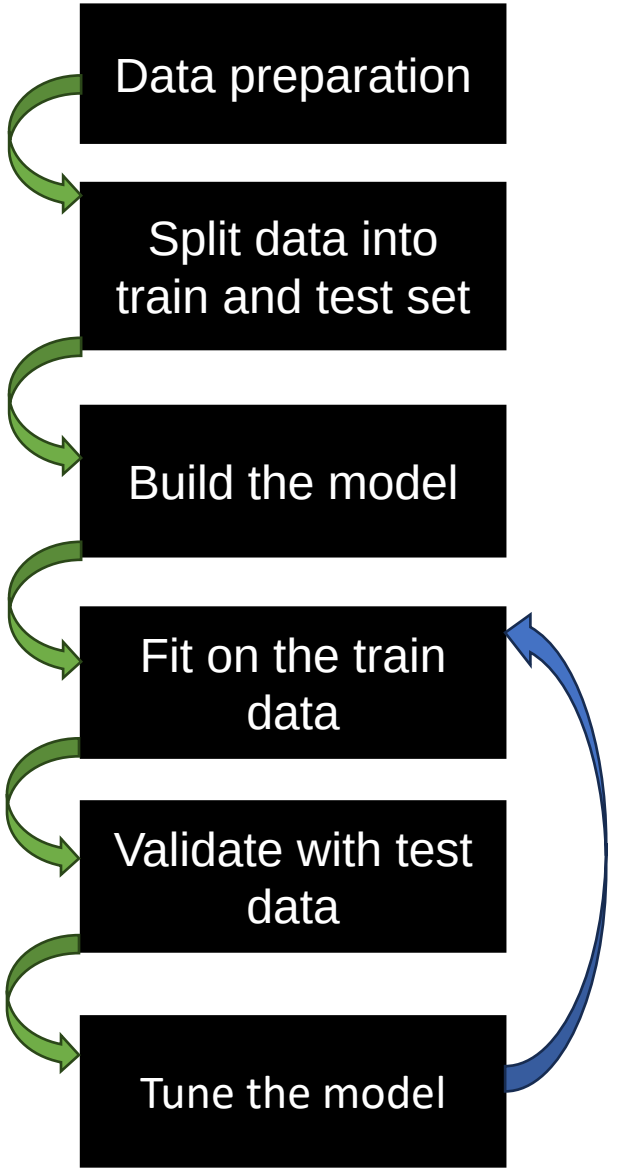
- There is ***80-20*** split in favour of class 0.
- 4771 customers have ***repaid*** the loan (class 0) and
- 1189 customers have ***defaulted*** on the loan.
- Customers with "*Office*" job have ***lowest*** default rate.
- "Sales" and "Self employed" customers have very high default rate.
- Loans on ***home improvement*** has higher defualt rate compared to loans on debt consolidation.
- In chi-square and pearson p-value test we did not find any ***insignificant*** valiables. those retained all of them
- Successfully ***treated*** outliers.
- Successfully ***treated*** missing values.

Model Building

Model Development and Evaluation:

- ✓ Separated the target variable and predictor variables.
  - Target :BAD
  - Predictor Variables:
- ✓ Creating dummy variables for categorical variables.
- ✓ Normalized the data using minmax scalar.
- ✓ Split the data into *train* and *test* sets with 25% for test data.

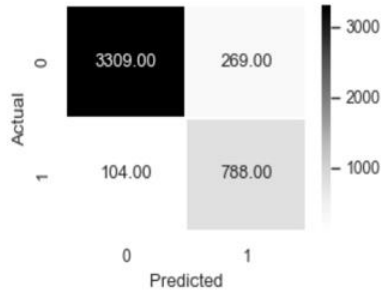
  applied **stratify** method to the split operation.
- ✓ Built and fit the model with train data.
- ✓ Validated against the test set.
- ✓ Captured performance of these models (e.g., accuracy, precision, recall, F1-score, confusion matrix and ROC-AUC scores.



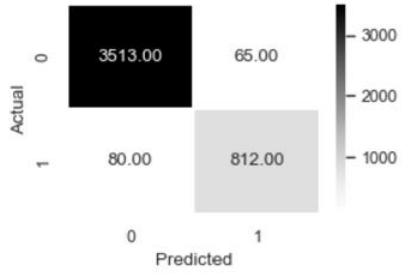
Dummy Variables

- **for REASON**
- REASON_DebtCon
- REASON_HomeImp
- **for JOB**
- JOB_Mgr
- JOB_Office
- JOB_Other
- JOB_ProfExe
- JOB_Sales
- JOB_Self

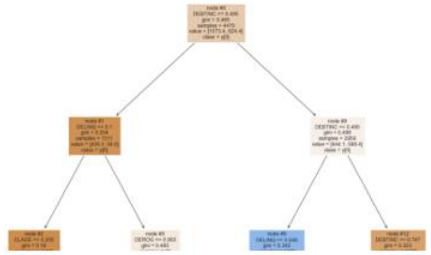
Figure 4

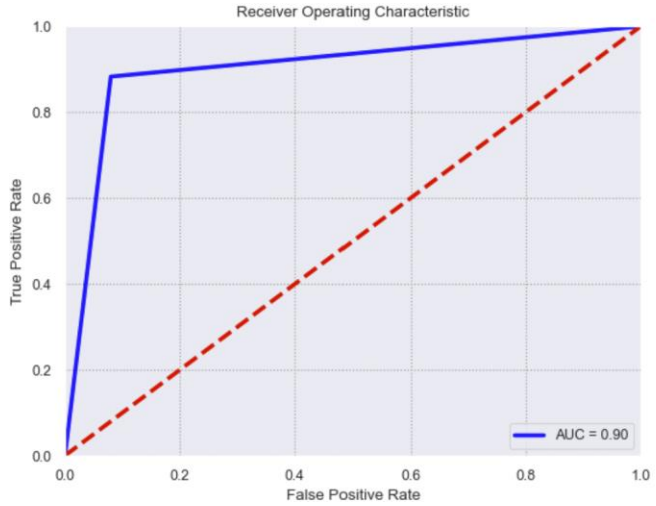


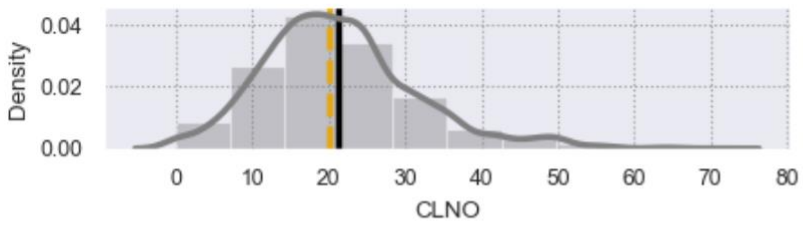
Logical Regression Tuned

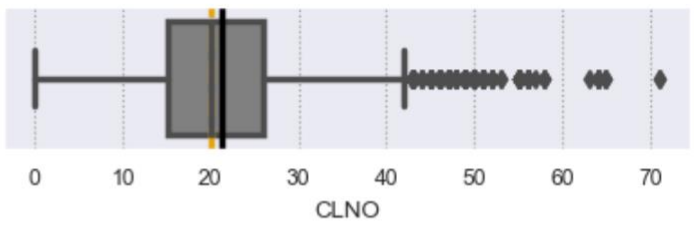


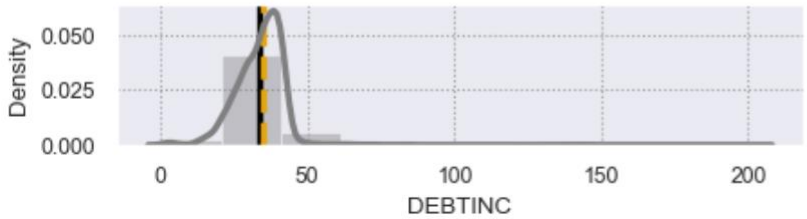
DTree Tuned

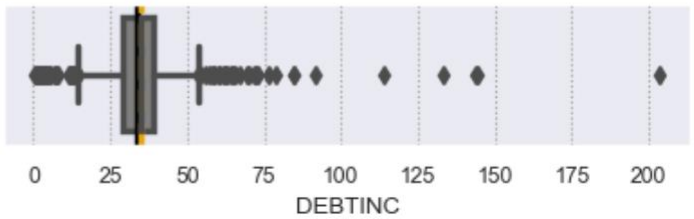


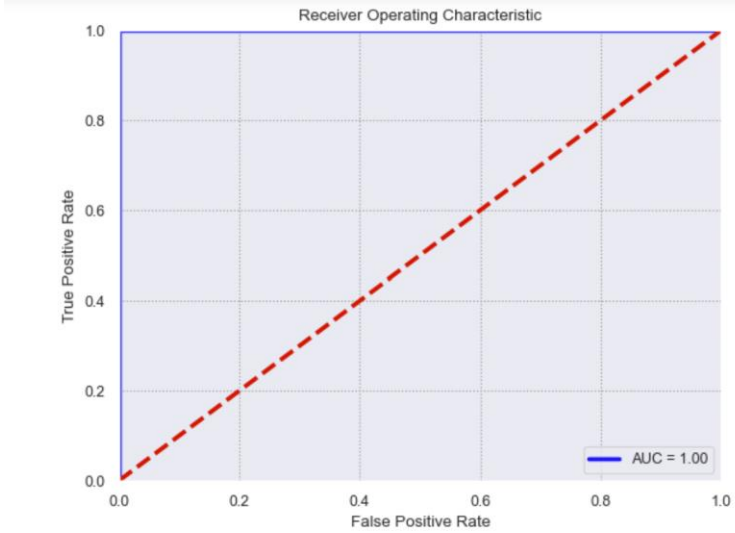










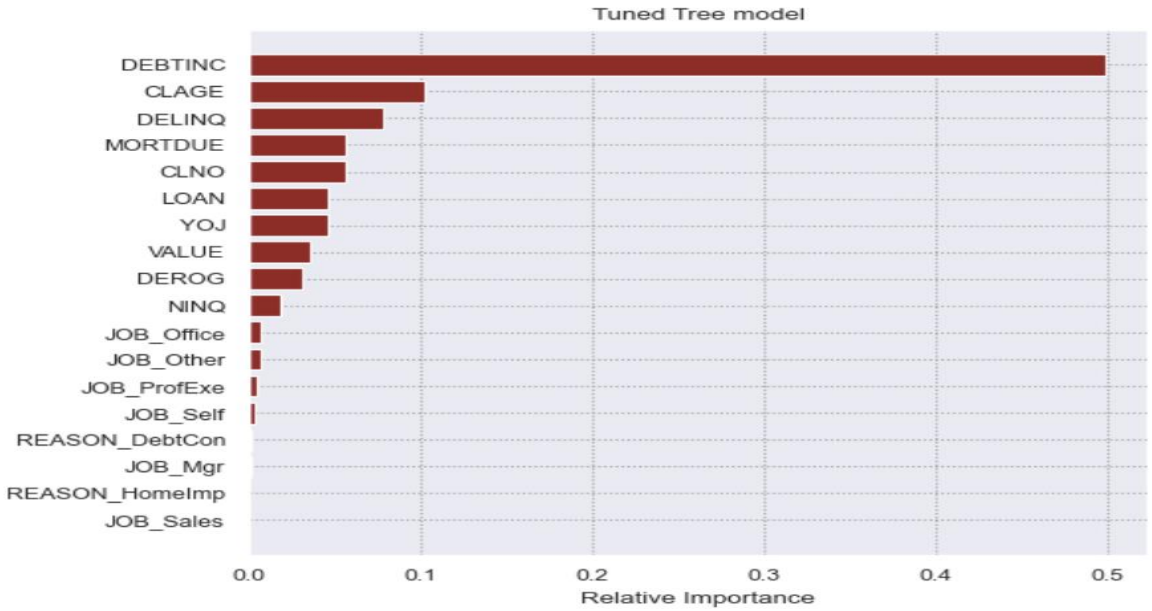


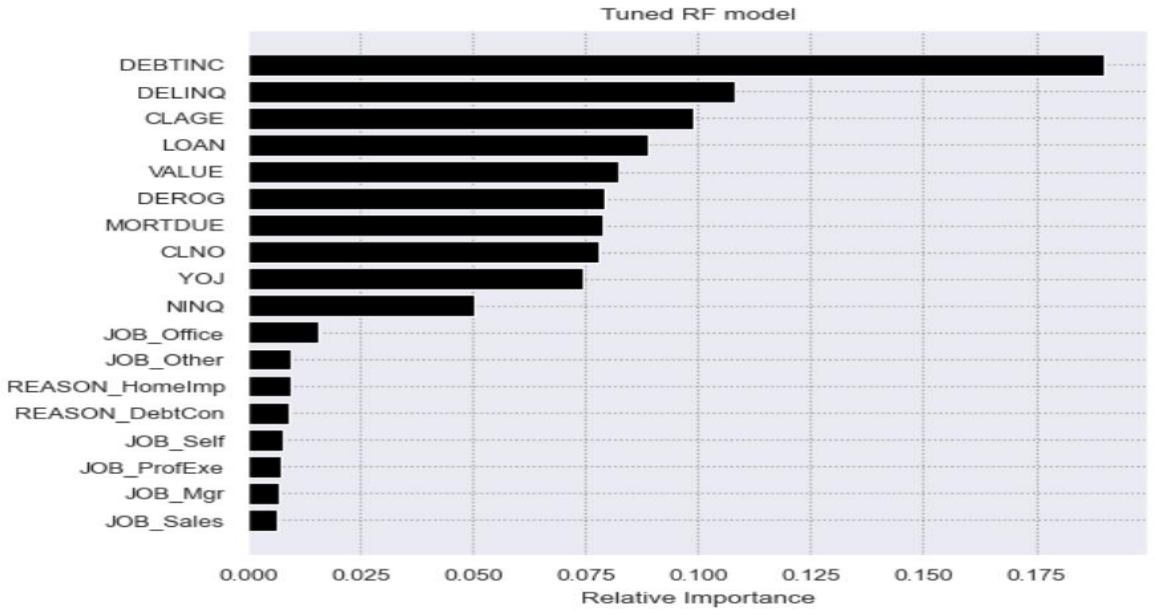
Model Performance

Model Performance Scores

	model_id	Data	f1-score	precision	recall	timelapsed
11	6:RandomF Tuned	Test	0.859922	0.883639	0.840983	1657.522707
10	6:RandomF Tuned	Train	0.948906	0.951809	0.946074	1657.000362
9	5:RandomF Orig	Test	0.851847	0.901615	0.819069	1.649158
8	5:RandomF Orig	Train	1.000000	1.000000	1.000000	1.161915
7	4:DTree Tuned	Test	0.805656	0.793018	0.821334	11.547906
6	4:DTree tuned	Train	0.877633	0.857517	0.904113	11.109076
5	3:DTree Orig	Test	0.787739	0.795374	0.780881	0.699065
4	3:DTree Orig	Train	1.000000	1.000000	1.000000	0.078837
3	2:LogisticR tuned	Test	0.663578	0.653599	0.712636	0.976619
2	2:LogisticR Tuned	Train	0.669923	0.659119	0.719788	0.421475
1	1:LogisticR Orig	Test	0.638528	0.771895	0.614535	0.623972
0	1:LogisticR Orig	Train	0.640136	0.809842	0.614654	0.062501

Feature Importances



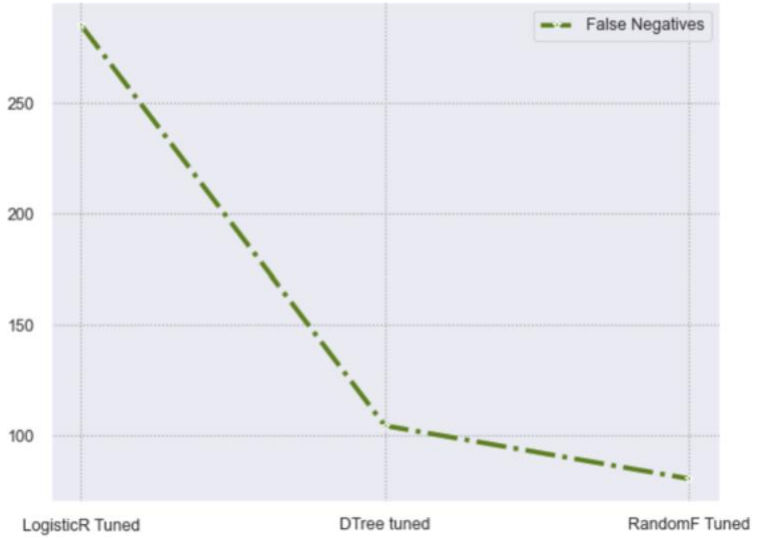


Observations

- RF model found ***more*** features of importance than the tree model.
- ***Top*** THREE features of importances are same on these models.
  - With CLAGE and DELINQ swaping 2nd and 3rd positions.
- **DTree** model did not give any importances to JOB_Sales, Reason_HomeImp, JOB_Mgr and REASON_DebtCon. Where as RF models given some importance to them.
- feature importance of CLNO and MORTDUE is ***higer*** in DTree model compared to RF.
- feature importance of LOAN and VALUE is ***higer*** in RF model compared to DTree.

Model selection

Tuned ***random forest*** model gave us best f1-scores of ***.95*** and ***.86*** on train and test data respectively, which is better than to tuned **tree model** f1-score of ***.88*** and ***.81*** on train and test data respectively. Which tells random forest done better with ***unseen*** data. Most likely to better on ***new*** data.



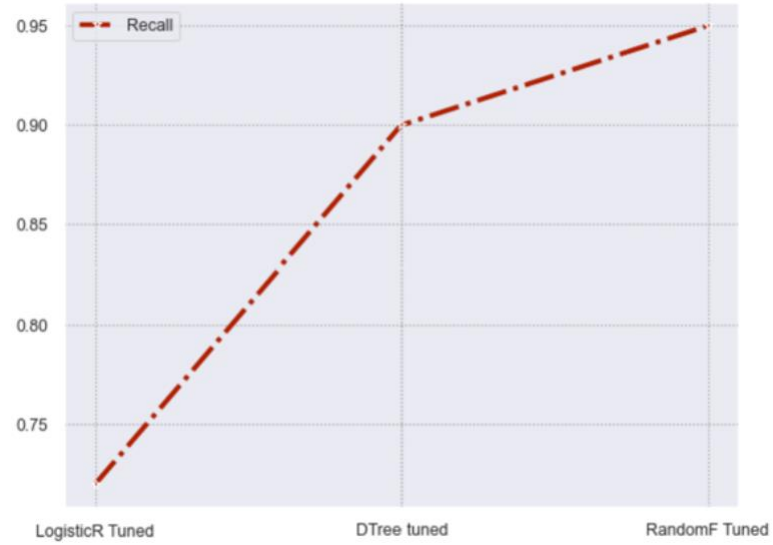
Tradeoff performance vs interpretability

We do not want to ***miss out*** any defaults. Therefore, we want "false Negatives" to be as low as possible. In these situations, we can compromise with the low precision, but recall should be high.

Logistic Regression(tuned) model, given very high ***false negatives*** lowering the recall score. Which means this model ***failed*** to detect many true defaults. This could be costly to the bank, using this model they will underwrite lots of bad loans.

Tree model(tuned) model, given lower ***false negatives*** improving the recall score. Which means this model ***did better*** to detect true defaults.

Random Forest(tuned) model, given ***lowest*** *false negatives* and ***most improved*** recall score. Which means this model ***succeeded*** to detect most true defaults. This could be very beneficial to the bank, using this model they will prevent underwriting bad loans.



Cost – benefit analysis

```
rf_net_savings = (TP * benefit_per_TP) - (FP * cost_per_FP) - (FN * cost_per_FN)
# Obtain predicted values from decision tree tuned model.
predicted = model4_d_tree_tuned.predict(X_test)
FP = np.sum((predicted == 1) & (actual == 0))
TP = np.sum((predicted == 1) & (actual == 1))
FN = np.sum((predicted == 0) & (actual == 1))

# Calculate net savings for DT
dt_net_savings = (TP * benefit_per_TP) - (FP * cost_per_FP) - (FN * cost_per_FN)
print( " ----------------EXAMPLE-------------")
print("cost per False Positive(FP):",cost_per_FP)
print("cost per False Negative(FN):",cost_per_FN)
print("benefit per True Positive(TP):",benefit_per_TP)
print('Net Savings from random forest tuned model:', rf_net_savings)
print('Net Savings from decission tree tuned model:', dt_net_savings)
```

```
 ----------------EXAMPLE-------------
cost per False Positive(FP): 100
cost per False Negative(FN): 200
benefit per True Positive(TP): 500
Net Savings from random forest tuned model: 85500
Net Savings from decission tree tuned model: 82900
```

Random forest model delivered highest savings.

Considering all these factors I've selected **"Random Forest(tuned) "** as the ***F I N A L*** model to recommend to the bank.

Deployment and Real-World Application:

An F1-score of 0.93 and recall score of .92 indicates a high level of accuracy and balanced performance in terms of both precision and recall. This is a positive indication for deploying the model in a production environment. Before deciding I suggest evaluating these factors.

- ✓ Performance on Real-Time or Recent Data.
- ✓ Scalability and Efficiency: Consider the computational requirements of the model, including memory, processing power, and response time. Ensure that the model can handle the expected workload in a timely manner, without causing significant delays or resource bottlenecks.
- ✓ Integration with Existing Infrastructure: Assess whether the model can seamlessly integrate with the existing production infrastructure and workflows. Consider compatibility with programming languages, libraries, and frameworks used in the deployment environment.
- ✓ Risk Management: Evaluate the potential risks associated with deploying the model. Assess factors such as regulatory compliance, ethical considerations, and potential biases or unintended consequences. Implement appropriate safeguards and monitoring mechanisms to mitigate risks.
- ✓ Maintenance and Updates: Consider the long-term maintenance and update requirements of the model. Models deployed in production may need periodic retraining or fine-tuning to adapt to changing data patterns and ensure continued optimal performance.

Conclusion

By leveraging data science and machine learning algorithms, the bank can create a more objective and streamlined loan approval process. This technology-driven approach holds the potential to minimize human error and biases while efficiently assessing the creditworthiness of loan applicants. Striking the right balance between automation and fairness is a key consideration to ensure a robust and reliable loan approval system in the banking industry.

Although, the proposed model is a great one but not the ultimate one. With the advent of data science and machine learning techniques, the emphasis has now shifted toward developing machines capable of learning the loan approval process. Besides these three techniques used other techniques can be tried.

Q&A