Loan Default Prediction

Nihar Patel
Practical Data Science

Problem Definition

Problem:

- To increase the bank's profits they must incur less borrowers that default and create a loss for the bank

Objective:

- Predict what types of borrowers default based on their credit profile of relevant features by building a classification model

Can this model be used for deployment?

Problem Importance

Data Analysis

Insights from data exploration

- 70/30 breakdown of loan request ‘REASON’, both similar default %s
- Sales and Self-employed workers defaulted the most
- High ‘DELINQ’ and ‘DEROG’ mean default but this was <25%
- ‘DEBTINC’ contained the most missing values

Data Manipulation

- Treatment of Outliers
- Treatment of Missing Values
- Categorical to Numerical Transformations

Model Selection Criteria

What are the goals of our model?

- Maximize recall score through minimizing false negatives
- Predict rest of the borrowers (precision, accuracy)
- Perform well on test data and not overfit
- Be easy to understand and apply

Solution Approach - Tuned Decision Tree

- Models Ran on train and test data:
  - Logistic regression
  - Decision trees
  - Random forests
- Original Model vs Tuned Models
  - GridSearchCV gave stronger results
  - Precision score sacrificed for recall score
- Tuned Decision Tree
  - Insights and Scalability
  - ‘DEBTINC’ variable

Final Model Performance

- Recall score of 78%, performs well on test data
  - Of all the clients that were accepted for a loan, only 6% of these are projected to default
  - More overall defaults projected but that can be good sign
- Model simplicity with variable breakdown
- Enhancements
  - Gauge performance of max precision and accuracy
  - Additional tuning through gradient boosting or pruning
  - Reassess the treatment of outliers and missing values

Proposed Business Solution

What are our recommendations?

1. Check for data privacy to ensure ECO Act compliance
2. Setup timeline and guidelines
3. Implement cross functional infrastructure between systems and stakeholders
4. Intertwine current approval process with working model
5. Monitor progress and update model routinely

Executive Summary

- Tuned Decision Tree will dynamically solve the bank's dilemma by
    - Decreasing loan default rate
    - Efficiently using human, physical, and computational resources
    - Highlighting key features like debt to income to understand borrower's credit health
- A 78% recall score is a reliable metric to begin real world application
- Enforcing a standard of data completeness will reduce risk of default
- Continuously monitoring data and borrower trends will serve fruitful

Risks & Challenges

- Over reliance on model may make loan approval a transactional task and not a human decision
- Overemphasis on one metric such as debt to income ratio may reject too many borrowers that may have pay off their loan]
  - Keep examining data
- Capturing missing values may decrease the amount of applications received since borrowers may not want to to put their debt to income ratio

Thank You!

Appendix A

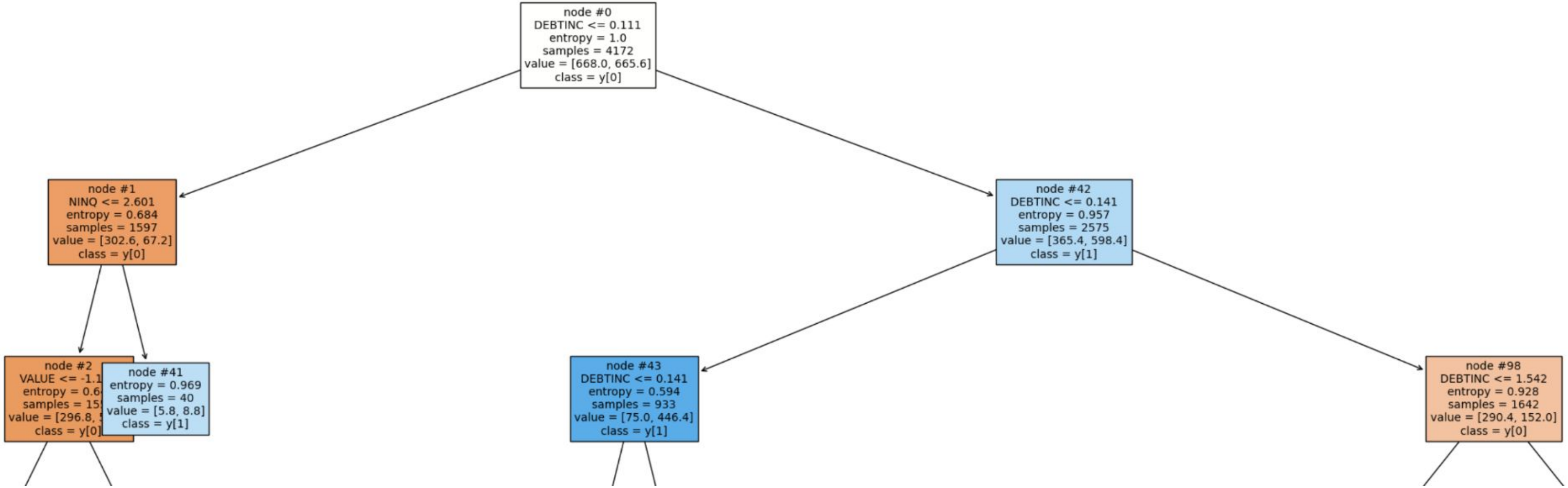
```
DecisionTreeClassifier

DecisionTreeClassifier(class_weight={0: 0.2, 1: 0.8}, criterion='entropy',
                       max_depth=12, min_samples_leaf=25, random_state=1)
```

	Model	Train_Accuracy	Test_Accuracy	Train_Recall	Test_Recall	Train_Precision	Test_Precision
0	Logistic Reg	0.808	0.809	0.088	0.084	0.640	0.682
1	Decision Tree	1.000	0.847	1.000	0.574	1.000	0.627
2	Tuned Decision Tree	0.838	0.825	0.864	0.779	0.560	0.543
3	Random Forest	1.000	0.891	1.000	0.619	1.000	0.789
4	Tuned Random Forest	0.838	0.825	0.864	0.779	0.560	0.543

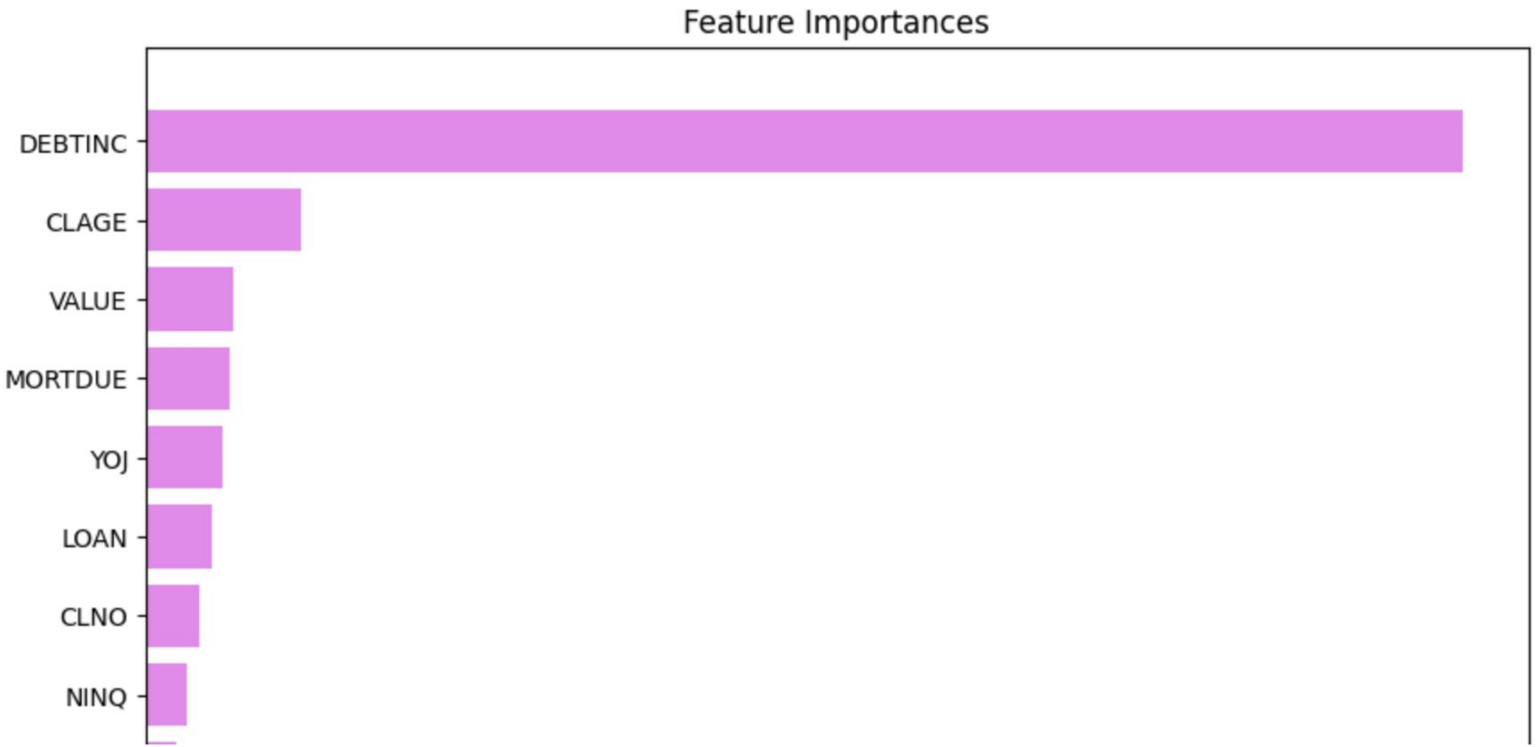
Appendix B

	precision	recall	f1-score	support
0	0.94	0.84	0.88	1431
1	0.54	0.78	0.64	357
accuracy			0.82	1788
macro avg	0.74	0.81	0.76	1788
weighted avg	0.86	0.82	0.84	1788

Appendix C


node #0
DEBTINC <= 0.111
entropy = 1.0
samples = 4172
value = [668.0, 665.6]
class = y[0]
node #1
NINQ <= 2.601
entropy = 0.684
samples = 1597
value = [302.6, 67.2]
class = y[0]
node #42
DEBTINC <= 0.141
entropy = 0.957
samples = 2575
value = [365.4, 598.4]
class = y[1]
node #2
class = y[0]
node #41
entropy = 0.969
samples = 40
value = [5.8, 8.8]
class = y[1]
node #43
DEBTINC <= 0.141
entropy = 0.594
samples = 933
value = [75.0, 446.4]
class = y[1]
node #98
DEBTINC <= 1.542
entropy = 0.928
samples = 1642
value = [290.4, 152.0]
class = y[0]

Appendix D



Feature Importances
DEBTINC
CLAGE
VALUE
MORTDUE
YOJ
LOAN
CLNO
NINQ