Credit Card Fraud Detection

Valentina Aparicio · Hayley Cromer · Chloe Griffin · Abbigale Outlaw

Spring 2020 · MTH 480

PICMath

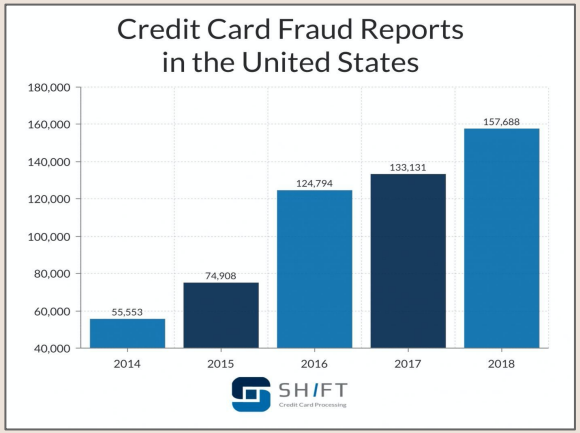
MAA
MATHEMATICAL ASSOCIATION OF AMERICA

CONVERSE

Credit Card Fraud Detection

Problem:

- Number of fraudulent transactions keeps rising
- Lots of money lost daily due to credit card fraud



[1]

Project Goal:

- Find new creative ways to detect fraud data
- Compare various detection methods

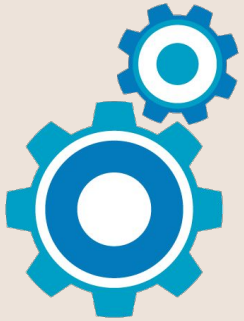
Results

- Preferred Method: Random Forest [11]

- SMOTE to handle imbalanced data
- Importance of specific variables
- Orange 3 for data visualization [5]

Approach

1. Researched our anonymous industry partner
2. Explored several methods and techniques we could use.
3. Implemented those into several Python and Orange 3 programs.
4. Improved and updated the programs in order to get better results.
5. Compared the final outputs, scores and overall performance of the programs.
6. Made final conclusions on preferred methods and recommendations.



[4]

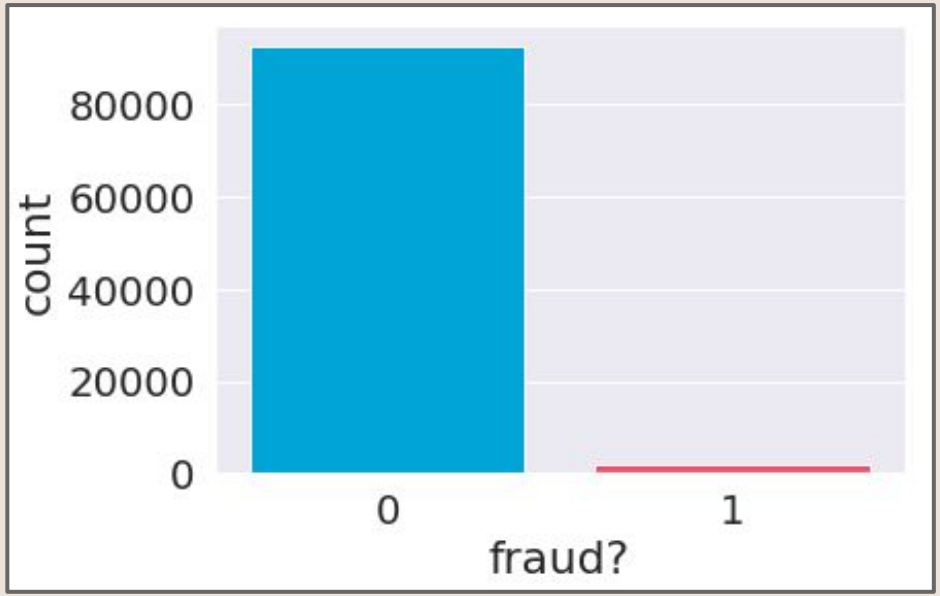
Data Sets

"CCDataMining1"

- 20 variables (columns)
- 65,534 samples (rows)

"CCDataMining2"

- 20 variables (columns)
- 100,000 samples (rows)



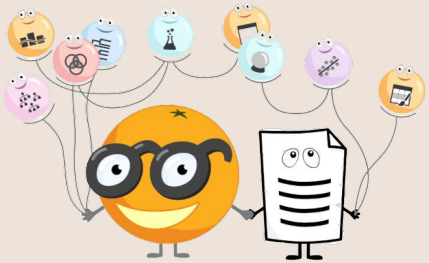
"CCDataMining1:" fraud vs not fraud

Program Development

Orange 3 Software

- Component based data mining framework.
- Data visualization and data analysis.

[6]



Jupyter Notebooks

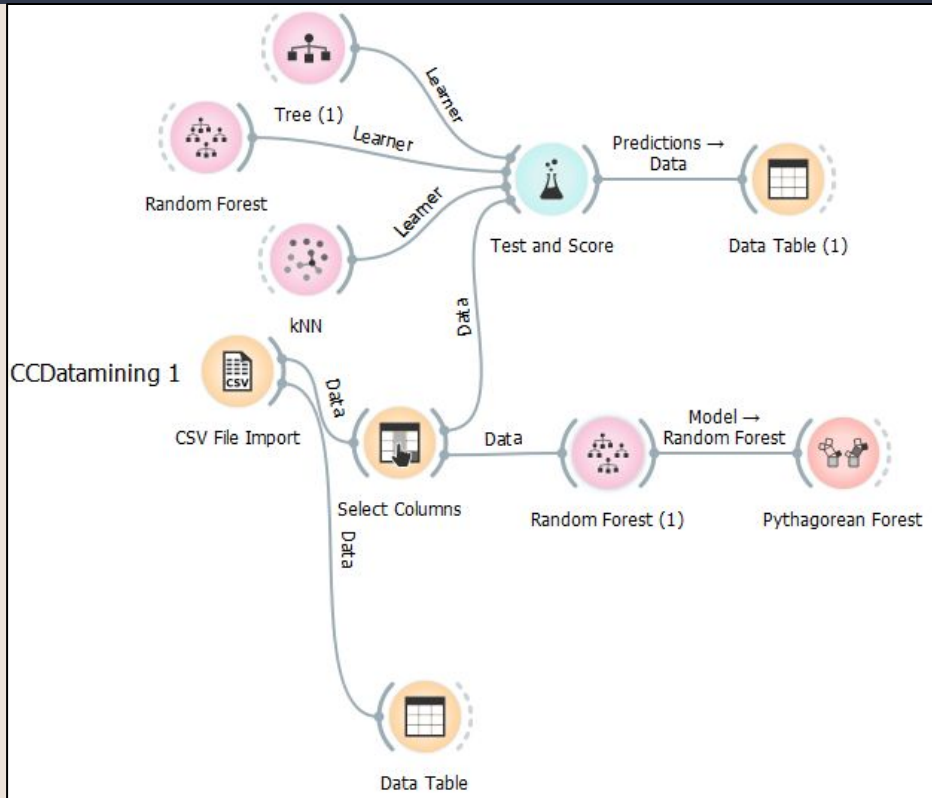
- Uses code cells to run an IPython kernel.
- Already used within partner financial institution.

[9]

[3]

Method Testing - Orange 3

1) Inputs data from CCDatamining1
2) Selects target variable
3) Tests and Scores three methods
4) Gives results and illustrates predictions

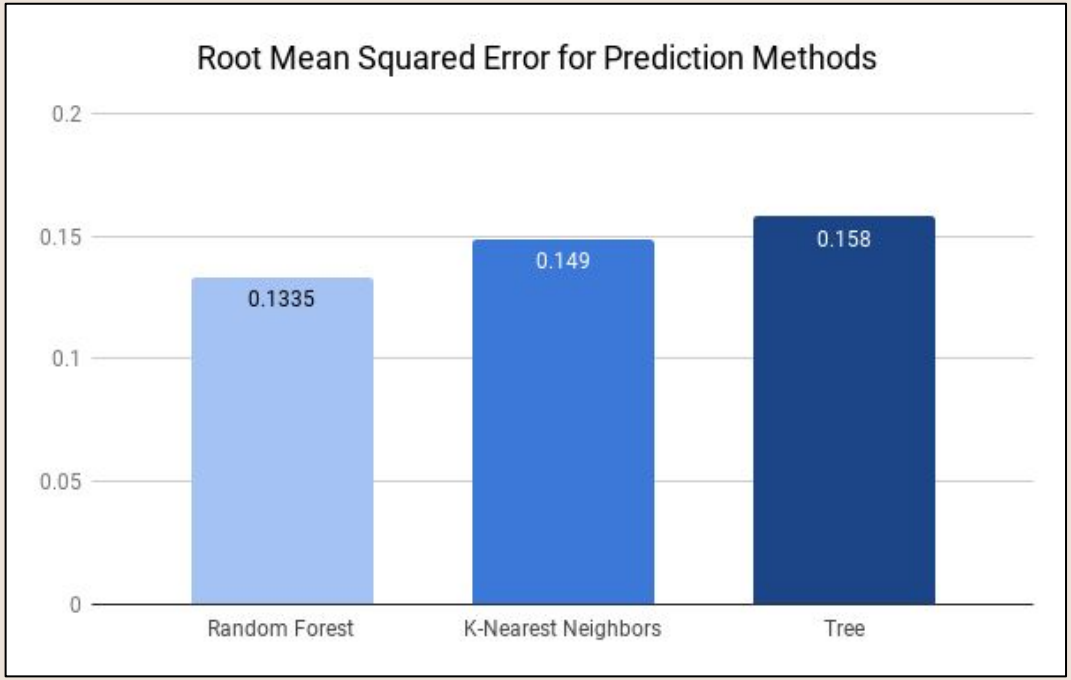


Method Testing - Interpreting Results

Root Mean Squared Error:

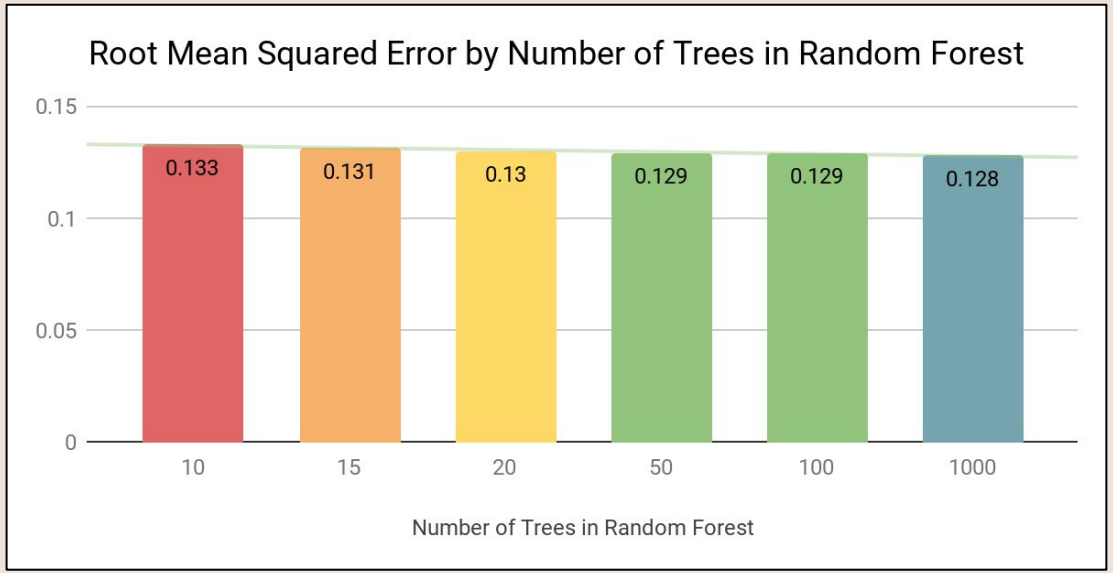
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}$$

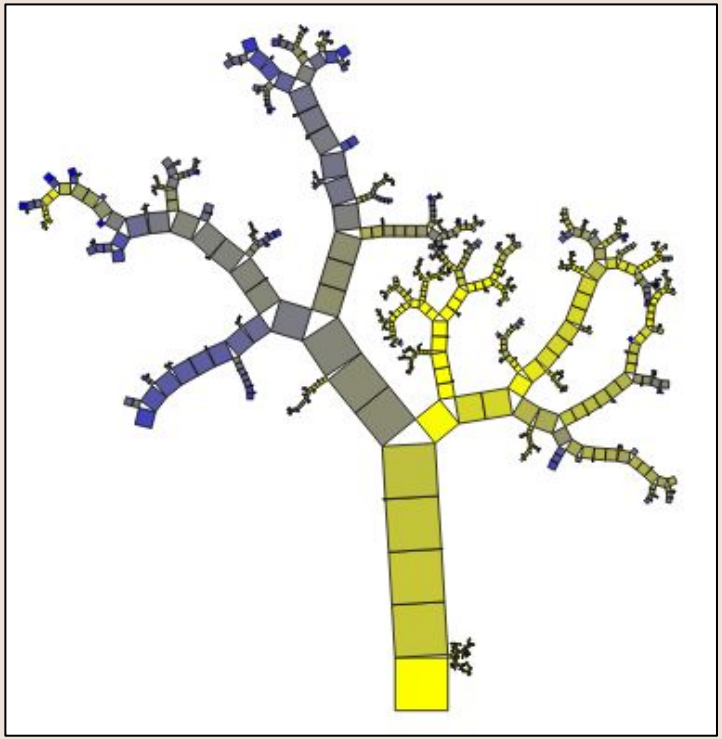
- Initial method comparison
- Least error: Random Forest



Method Testing - Random Forest Trees

- More trees = Less error = Longer runtime
- Visual illustration of Random Forest tree



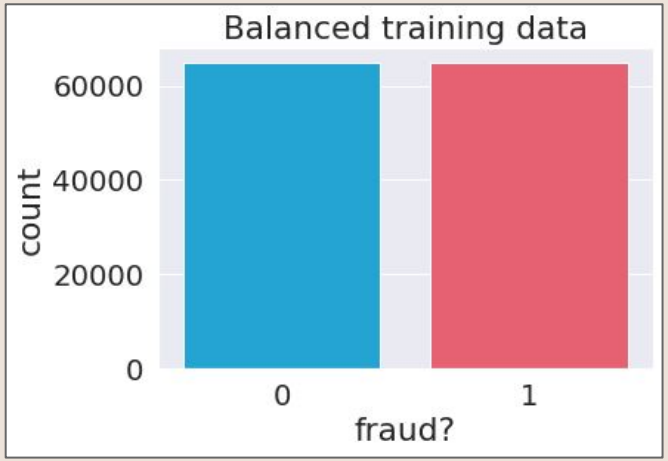


Random Forest in Python

- Built using pandas and Scikit-Learn RandomForestClassifier library and imbalanced-learn SMOTE on Jupyter Notebooks.

[7]

- What does the program do?
  - Provides features comparison,
  - Splits into training and testing sets,
  - Uses SMOTE to deal with the imbalanced data,
  - Trains the model, and
  - Tests predictions.



Random Forest Results

Tests provides:

- Recall
- Precision, and
- F1 scores.

	precision	recall	f1-score	support
0	0.98	1.00	0.99	27776
1	0.73	0.32	0.45	629
accuracy			0.98	28405
macro avg	0.86	0.66	0.72	28405
weighted avg	0.98	0.98	0.98	28405

CCDatamining1 Results

	precision	recall	f1-score	support
0	0.98	1.00	0.99	29204
1	0.73	0.43	0.54	796
accuracy			0.98	30000
macro avg	0.86	0.71	0.77	30000
weighted avg	0.98	0.98	0.98	30000

CCDatamining2 Results

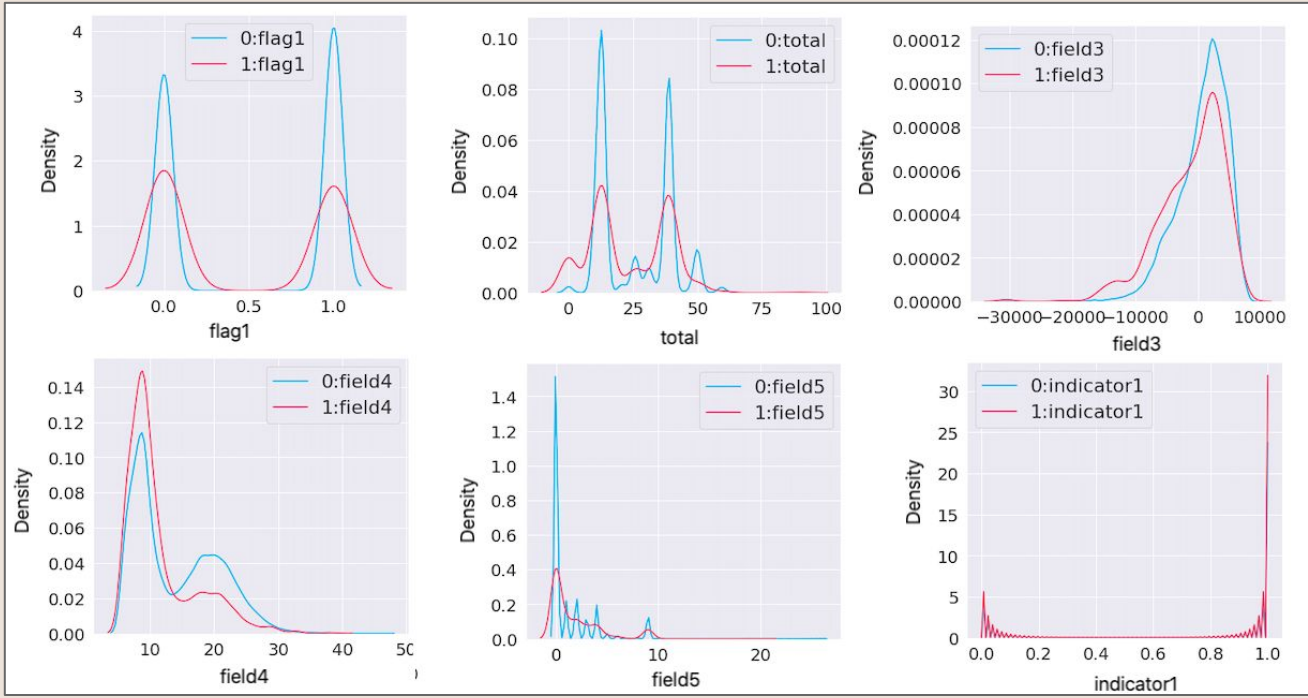
CCDataMining1 Features Comparison

Field 3 & Field 4

- Similar for normal and fraudulent
- Not good indicators

Total & Field 5

- Differences in values
- May be better indicators



K-Nearest Neighbors

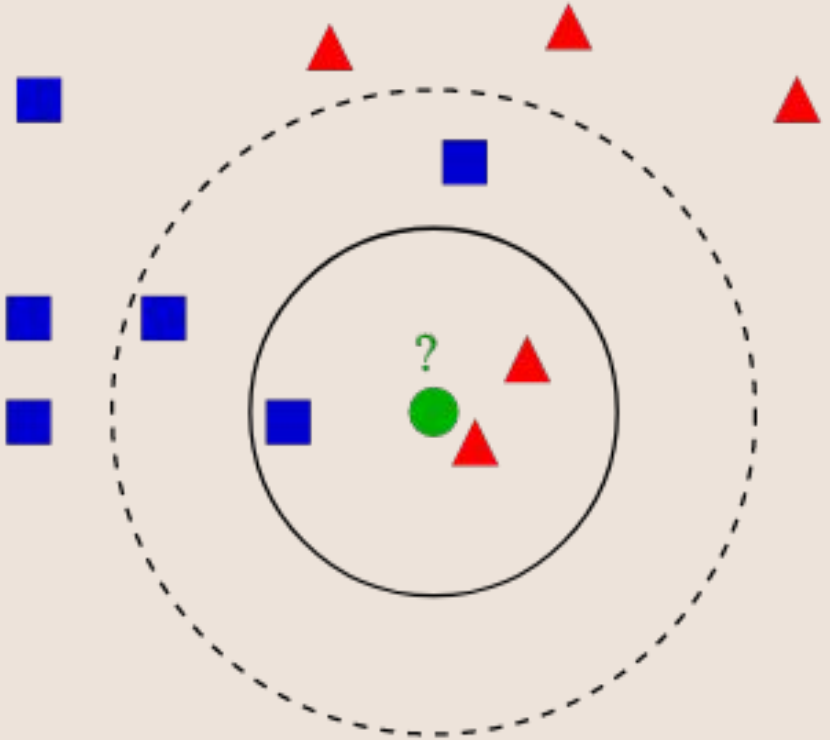
- A non-parametric method for classification and regression

Advantages

- Strong consistency rate
- Simple to implement

Disadvantage

- Slows as the size of the data set increases



[8]

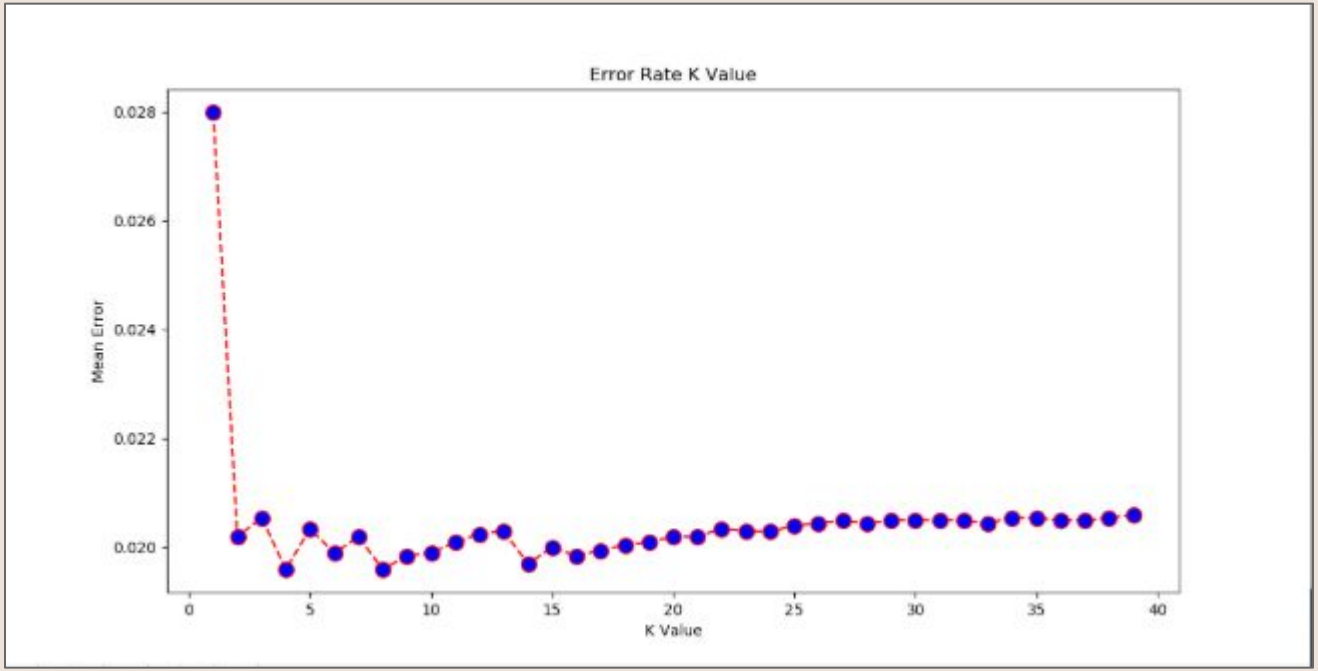
K-Nearest Neighbors Program

- Built using Scikit-Learn KNeighborsClassifier library
- 2 versions of the program
  - Splits the data
  - Trains the model
  - Determines the “score” for non-fraud and fraud predictions
  - Prints classification report

	precision	recall	f1-score	support
0	0.98	1.00	0.99	19413
1	0.76	0.34	0.47	587
accuracy			0.98	20000
macro avg	0.87	0.67	0.73	20000
weighted avg	0.97	0.98	0.97	20000

Classification Report, CCDataMining2 (k=4)

K-Nearest Neighbors Program



k error comparison with CCDataMining2

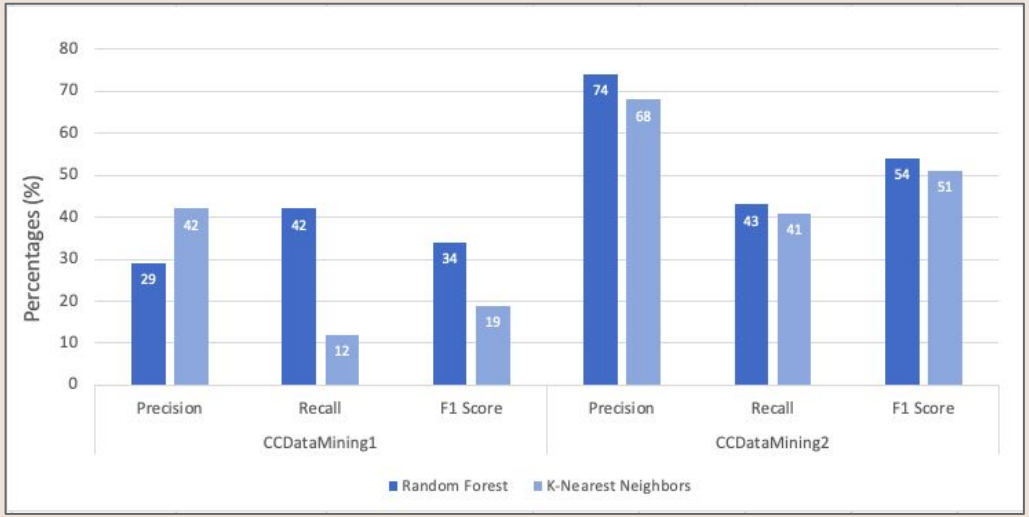
Project Recap

Problem

- Credit Card Fraud causes companies to lose a great deal of money everyday.
- Our goal was to compare methods and provide suggestions to improve the detection process.

Results

- The Random Forest method outperformed K-Nearest Neighbors on Python and Orange 3 softwares.



Suggested Future Work

- Try other data sets within existing programs,
- Determine time and recall balance for Random Forest Programs, and
- Consider implementing Orange 3 and SMOTE programs

[3]

[5]

[10]

[9]

Acknowledgements

This project is a part of the **PIC Math** (Preparation for Industrial Careers in Mathematics) program, which is sponsored by the **Mathematical Association of America** (MAA). Support is provided by the **National Science Foundation** (NSF grant DMS-1722275) and the **National Security Agency** (NSA).

Faculty Advisor: Jessica Sorrells

References

1. “Credit Card Fraud Statistics.” *[Updated February 2020] Shift Processing*, shiftprocessing.com/credit-card-fraud-statistics/.
2. “File:Python-Logo-Notext.svg.” *File:Python-Logo-Notext.svg - Wikimedia Commons*, commons.wikimedia.org/wiki/File:Python-logo-notext.svg.
3. “Process Icon: Symbols, Lettering.” *Pinterest*, ar.pinterest.com/pin/54184001750129392/.
4. "Google Summer of Code 2016 Wrap-up: Orange." *Google Open Source Blog*, opensource.googleblog.com/2017/01/google-summer-of-code-2016-wrap-up_25.html.
5. University of Ljubljana. “Orange Data Mining - Data Mining.” Orange Data Mining - Data Mining, orange.biolab.si/.
6. "Pandas."NumFOCUS, numfocus.org/project/pandas.
7. “File:KnnClassification.svg.” File:KnnClassification.svg - Wikimedia Commons, commons.wikimedia.org/wiki/File:KnnClassification.svg.
8. “File:Jupyter Logo.svg.” *File:Jupyter Logo.svg - Wikimedia Commons*, commons.wikimedia.org/wiki/File:Jupyter_logo.svg.
9. Flatart. “'Logos and Brands - Line Filled' by Flatart.” *Iconfinder*, www.iconfinder.com/icons/4519136/kaggle_icon
10. “Collection of Animated Forest Cliparts (47).” *Free Animated Forest Cliparts, Download Free Clip Art, Free Clip Art on Clipart Library*, clipart-library.com/animated-forest-cliparts.html.

Thank you!