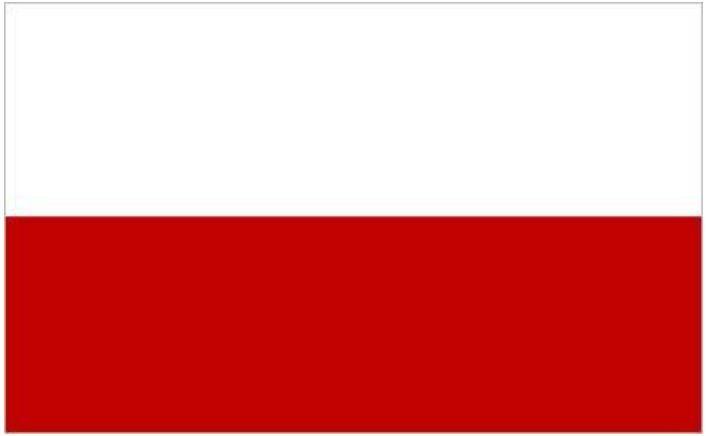
# 412: Predicting Bankruptcy

By: David Contento, John Macke, Pujan Thakrar, and Mark Vandre

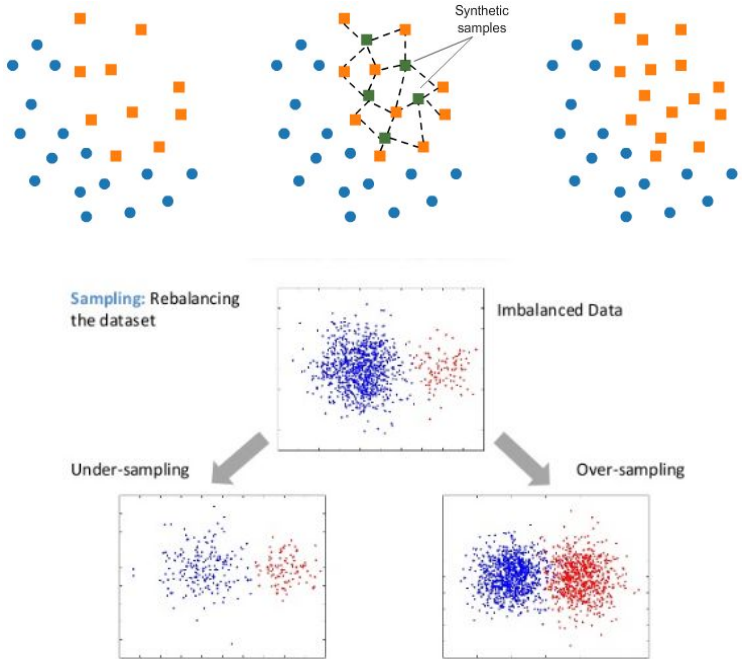
# Objective and Data

- 10503 observation dataset, consists of 64 balance sheet variables
- Data is from 2010
- Indicator variable signifying bankruptcy
- Classification problem
- Wanted to fit different models and compare accuracy
- Unbalanced dataset



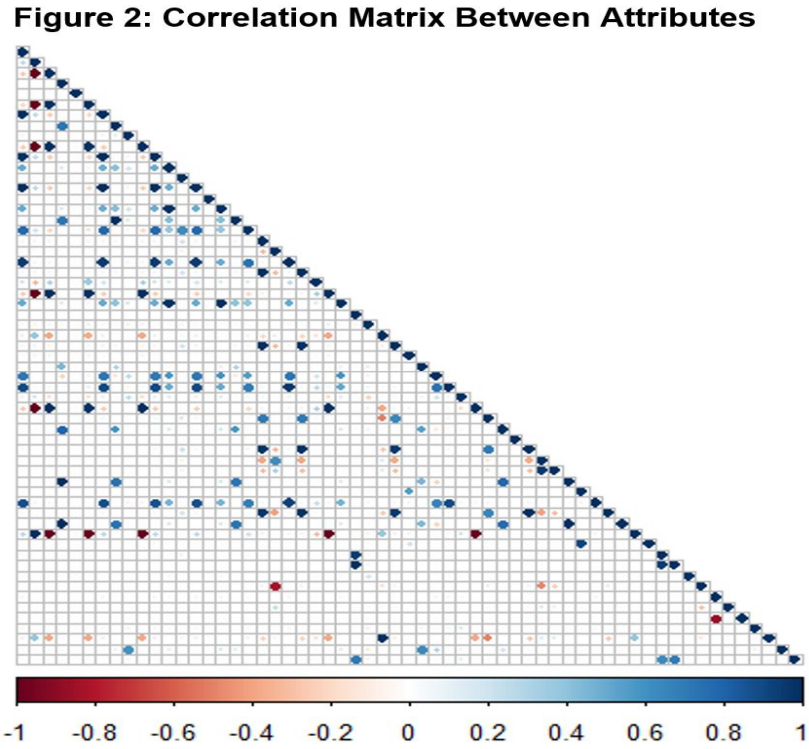
# SMOTE: Synthetic Minority Oversampling Technique

- Only 5% of observations resulted in bankruptcy
- Reduces model prediction performance
- Generate synthetic samples (KNN)
- Undersample overrepresented class and remove excess
- rebalance

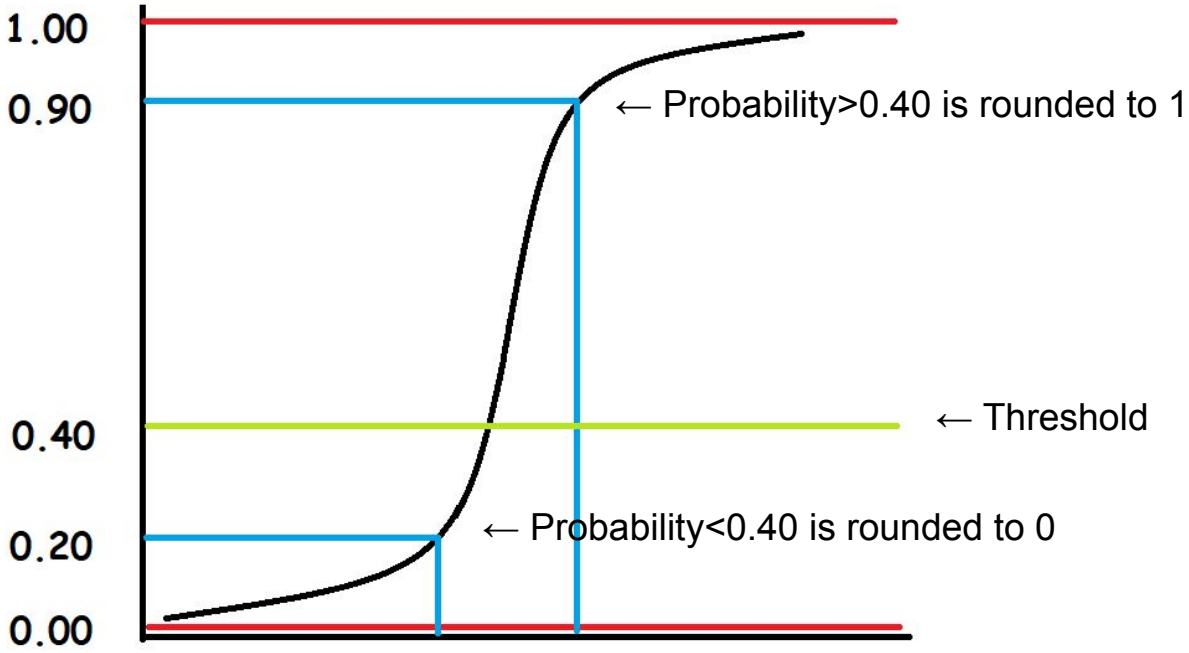


# Multicollinearity in Our Dataset

- Some variables are combinations of others.
- Extreme amount of collinearity
- Looked at VIF
- Ended up with 38 variables out of 64

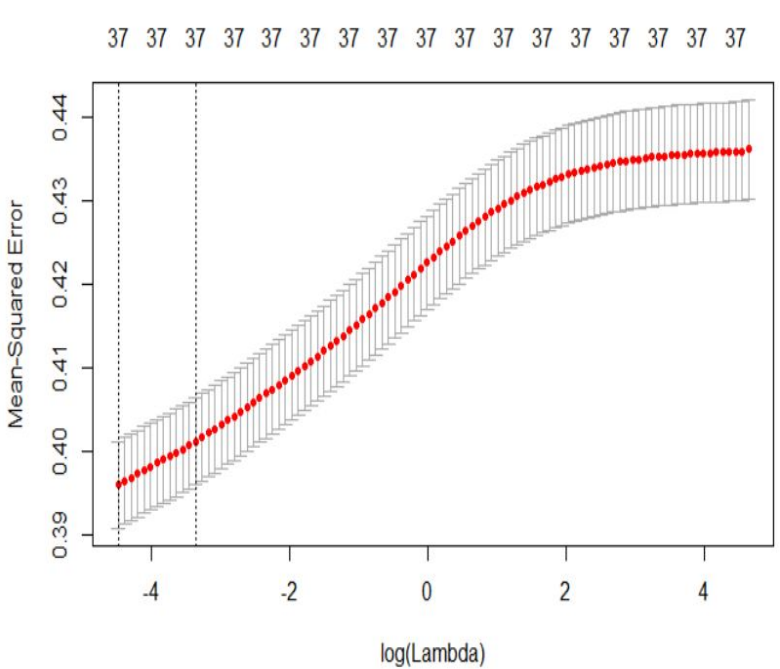


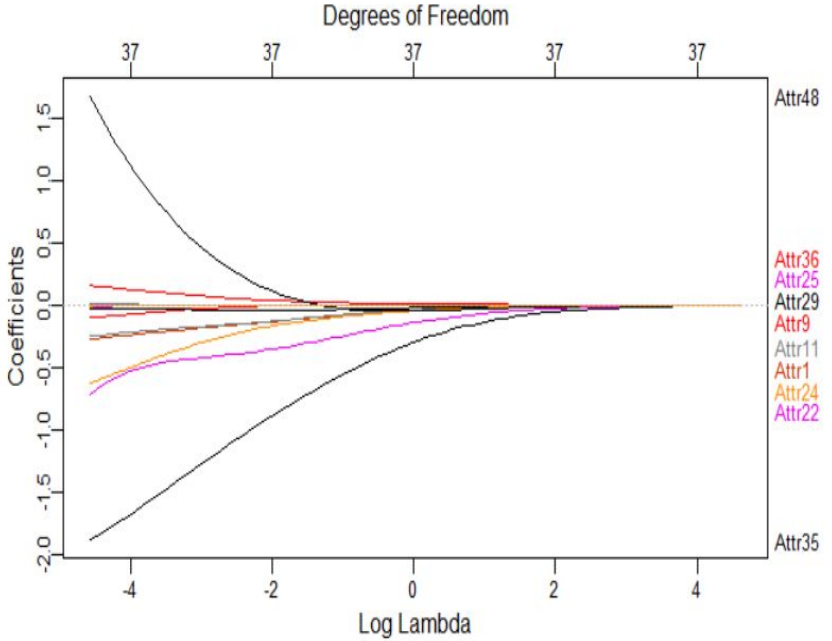
# Logistic Regression




Accuracy: 71%  Precision: 58%  Recall: 50%

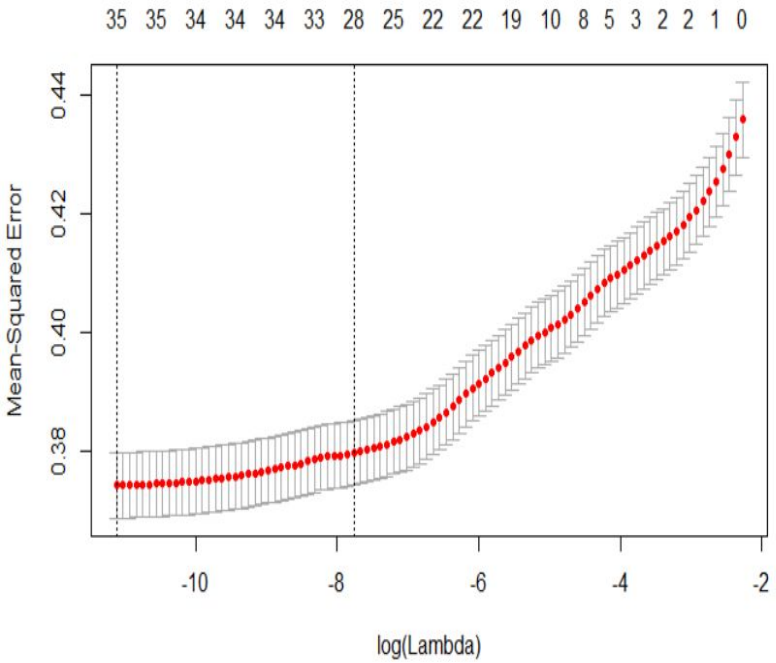
# Ridge Regression

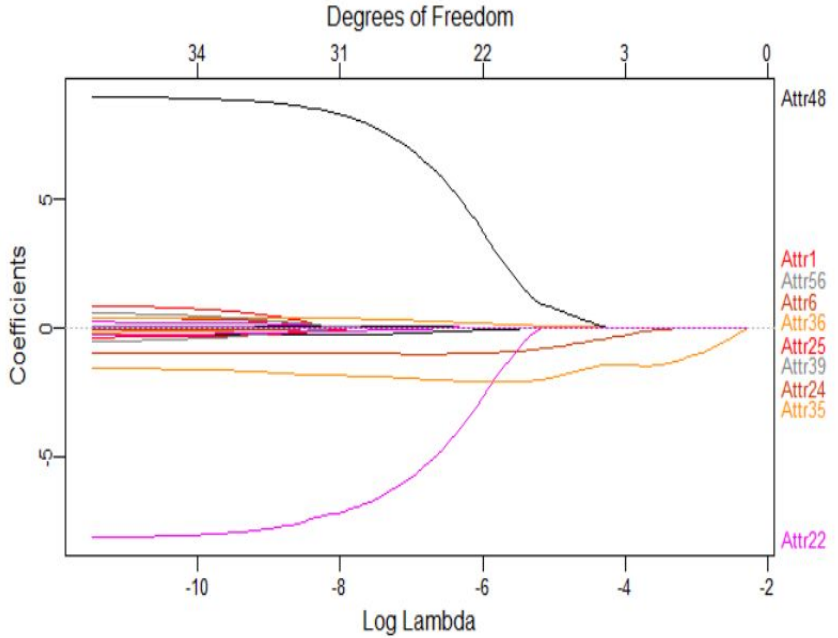



Accuracy: 70%  Precision: 69%  Recall: 28%

# LASSO



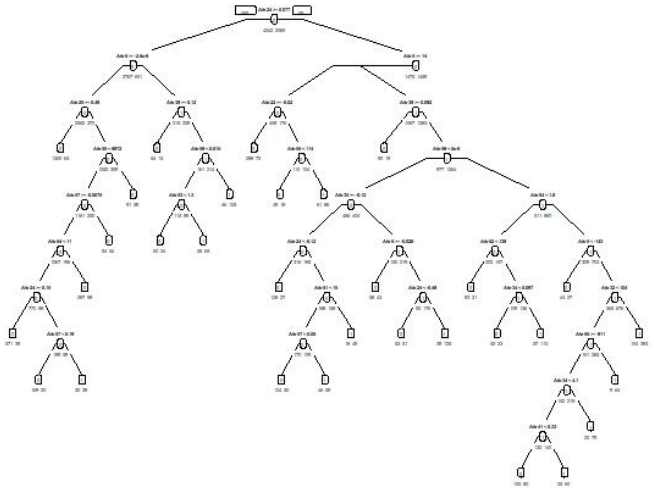




Accuracy: 69%  Precision: 62%  Recall: 19%

# Single Decision Tree vs. Pruned Decision Tree

Cp = 0 Minsplit = 200 nsplit = 31

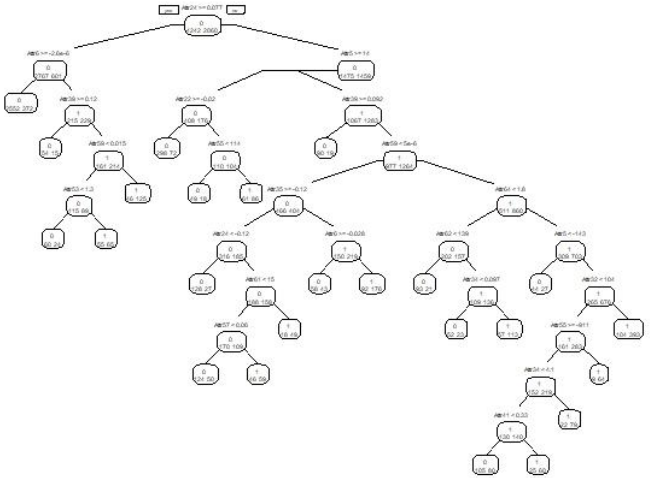
Figure 10: Default Classification Tree



Accuracy: 77% Precision : 69% Recall = 58%

Cp = 0.001 Minsplit = 200 nsplit = 23

Figure 13: Pruned Tree



Accuracy: 77% Precision : 69% Recall = 60%

# Best Pruned Decision Tree

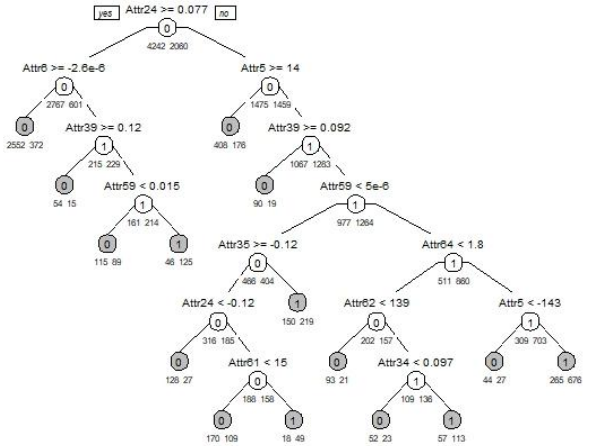
**Figure 12: Complexity Parameters Associated with Tree Errors**

| | CP | nsplit | rel error | xerror | xstd |
|---|---|---|---|---|---|
| 1 | 0.05242718 | 0 | 1.00000 | 1.00000 | 0.018076 |
| 2 | 0.03446602 | 2 | 0.89515 | 0.90534 | 0.017590 |
| 3 | 0.03179612 | 3 | 0.86068 | 0.87087 | 0.017390 |
| 4 | 0.02184466 | 5 | 0.79709 | 0.82233 | 0.017085 |
| 5 | 0.01359223 | 6 | 0.77524 | 0.81117 | 0.017011 |
| 6 | 0.01286408 | 8 | 0.74806 | 0.78155 | 0.016807 |
| 7 | 0.01262136 | 10 | 0.72233 | 0.77718 | 0.016776 |
| 8 | 0.00825243 | 11 | 0.70971 | 0.76602 | 0.016696 |
| 9 | 0.00752427 | 12 | 0.70146 | 0.76311 | 0.016674 |
| 10 | 0.00728155 | 14 | 0.68641 | 0.76068 | 0.016657 |
| 11 | 0.00631068 | 15 | 0.67913 | 0.75825 | 0.016639 |
| 12 | 0.00606796 | 16 | 0.67282 | 0.75631 | 0.016625 |
| 13 | 0.00485437 | 18 | 0.66068 | 0.75631 | 0.016625 |
| 14 | 0.00303398 | 19 | 0.65583 | 0.75340 | 0.016603 |
| 15 | 0.00097087 | 23 | 0.64369 | 0.74757 | 0.016560 |
| 16 | 0.00072816 | 24 | 0.64272 | 0.75922 | 0.016646 |
| 17 | 0.00001000 | 30 | 0.63835 | 0.76068 | 0.016657 |

Xerror = 0.76413 (0.74757+0.016560) → Cp =0.0075

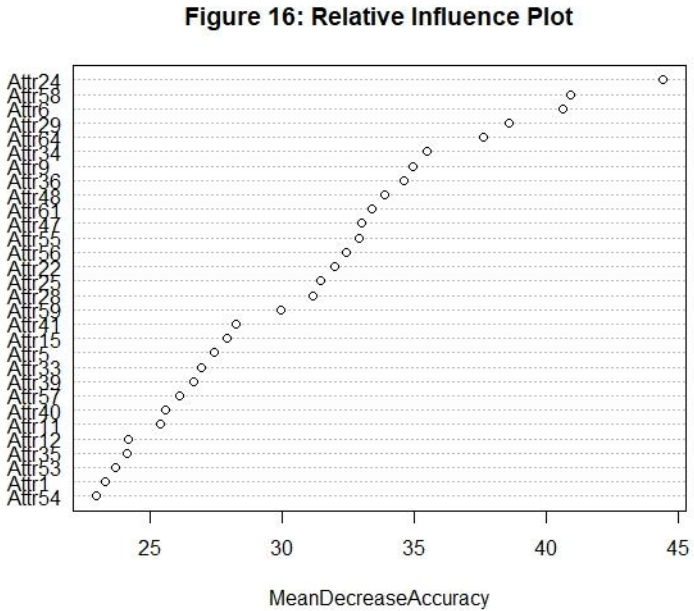
Cp = 0.0075 nsplit = 12

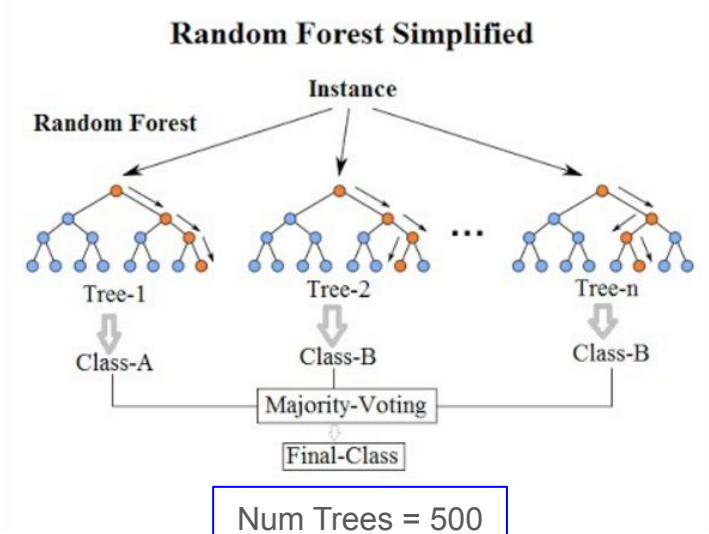
**Figure 15: Best Pruned Tree**




Accuracy: 76% Precision : 68% Recall = 58%

# Random Forest




Num Trees = 500

Not transparent → still good for prediction!

Validation: Accuracy: 94% Precision : 96% Recall = 86%

Test: Accuracy: 92% Precision : 91% Recall = 82%

# Results

| Model | Accuracy(%) | Precision(%) | Recall(%) |
|---|---|---|---|
| Logistic Regression | 71 | 58 | 50 |
| Ridge Regression | 70 | 69 | 28 |
| Lasso Regression | 69 | 62 | 19 |
| Decision Tree | 77 | 69 | 58 |
| Pruned Tree | 77 | 69 | 60 |
| Best Pruned Tree | 76 | 68 | 58 |
| Random Forest | 92 | 91 | 82 |

# Results

Worst Predictive Model

**Lasso**

| Model | Accuracy(%) | Precision(%) | Recall(%) |
|---|---|---|---|
| Logistic Regression | 71 | 58 | 50 |
| Ridge Regression | 70 | 69 | 28 |
| Lasso Regression | 69 | 62 | 19 |
| Decision Tree | 77 | 69 | 58 |
| Pruned Tree | 77 | 69 | 60 |
| Best Pruned Tree | 76 | 68 | 58 |
| Random Forest | 92 | 91 | 82 |

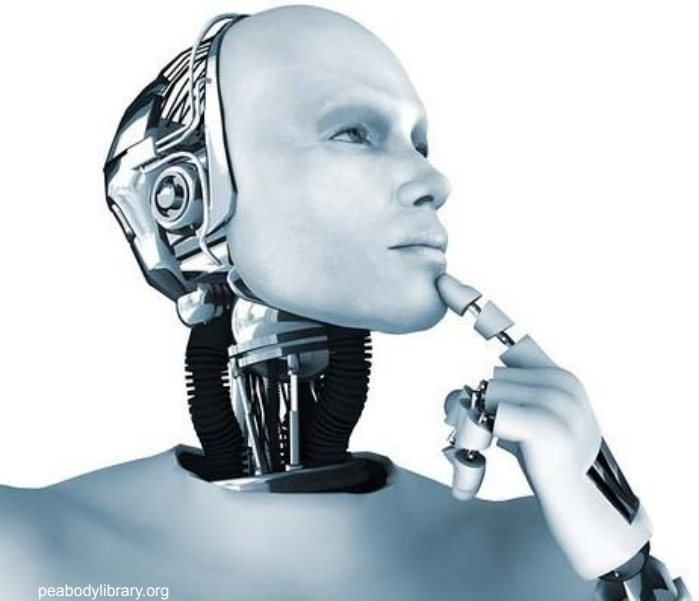
# Results

Best Predictive Model

**Random Forest**

| Model | Accuracy(%) | Precision(%) | Recall(%) |
|---|---|---|---|
| Logistic Regression | 71 | 58 | 50 |
| Ridge Regression | 70 | 69 | 28 |
| Lasso Regression | 69 | 62 | 19 |
| Decision Tree | 77 | 69 | 58 |
| Pruned Tree | 77 | 69 | 60 |
| Best Pruned Tree | 76 | 68 | 58 |
| Random Forest | 92 | 91 | 82 |

# Future Work




Additional Machine Learning Algorithms

- Support Vector Machine (SVM)
- Neural Network
- Boosted Trees/XG Boost

Different Settings

- United States
- Western Europe
- China & India