Inside College Scorecard

Presented by: Charles Chen, Issac Baron, Hong Fan, Jun Ouyan, Steven Gong, Richard Murphy, Shawn Xiang, Niall Smith, Suba Rajaram, Phoebe Wang, Zhening Zhang, Boli Zhang

Introduction

Kaggle: Inside College Scorecard challenge

Data on more than 4,000 colleges ranging from 1996-2013 (1996 - 2013)

Ranging from admission statistics, demographic, major offered and earning of graduates

Research Question

What determines the average income of graduates from a college?

- Admission Statistics (Admit rate, SAT)
- Demographic (Race, Gender)
- Family income
- Major (STEM, Humanity)
- Or is it the College Name Only?

Methods used to solve the problem

How we attempted to solve the question?

- Data Cleaning
- Linear regression
- Neural Networks
- K Nearest Neighbors
- Random Forest
- Support Vector Machine

Data Cleaning

- Overview of the raw data.
- Year: 1996-2013, Size: 1.2 G
- “.csv” versus “.sql”
- How we cleaned
- “Null”s & “Privacy Suppressed”s
- Most recent year
- Result: 1500+ universities, ~50 features, 6 earning measurements.

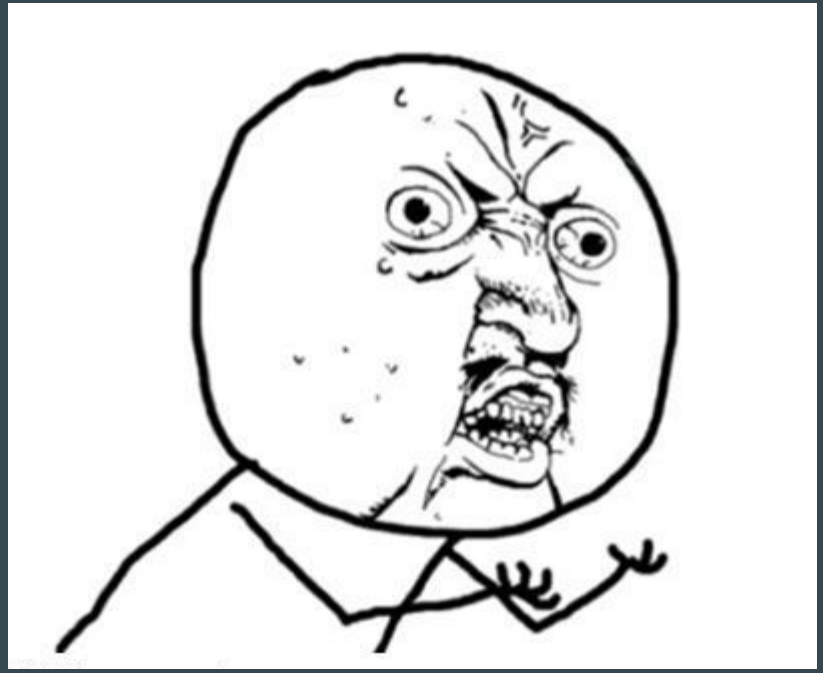
Linear regression

Challenges

- Large number of samples
- Large number of features
- excessive run times
- The feature were not all numbers

Possibility of Overfitting

- large number of features



linear regression —Gradient Descent

randomly choose features

The features we choose: SAT score, Average age of entry, Average family income , Median family income, Average family income for independent students , Median earnings of students

Problems : not the optimized features

Thought : Find the effective features

- Testing Accuracy : 87%

Linear regression —normal equations

randomly choose features

$$\theta = (X^T X)^{-1} X^T y$$

The features we choose: same

Solving with normal equation

when the size of features is low, you can think of the normal equations as the better option calculation theta. however, for greater values gradient descent is much more faster!

- Testing Accuracy : 89%

Linear Regression

Lasso

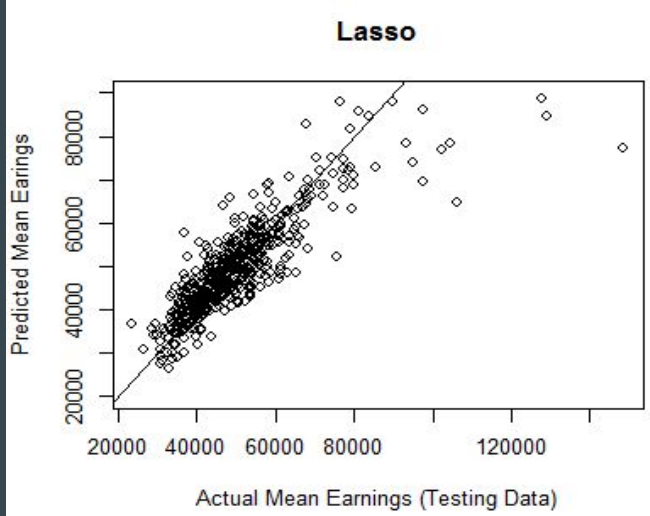
- Goal: Select informative features
- Approach: Use 10-fold cross-validation to select tuning parameter. Coefficients for Lasso is estimated by minimizing the sum of Residual sum of squares and penalty term. Some coefficients will be reduced to zero due to the penalty term.
- Results: 21 features are selected out of 46 and the 6 most informative features for Lasso: Family income(0.260), average SAT scores(0.249), major in Health(0.228), major in Engineering(0.225), percentage of students born in US(-0.141), female percentage(-0.106); smaller test MSE compared to Linear regression(all 46 features)

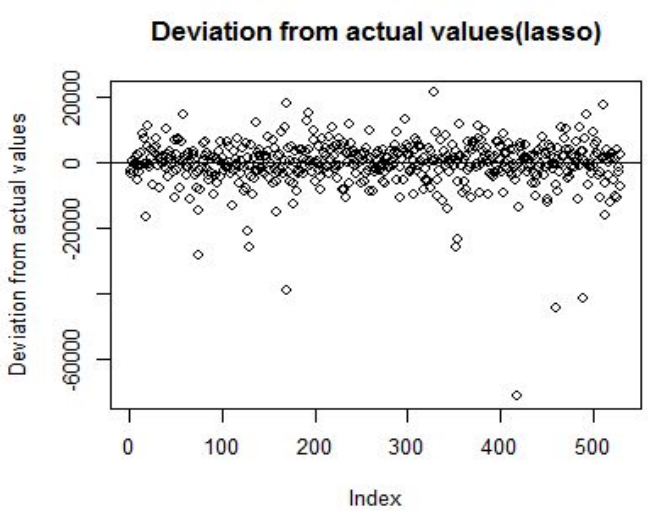
Exploration in Boosting

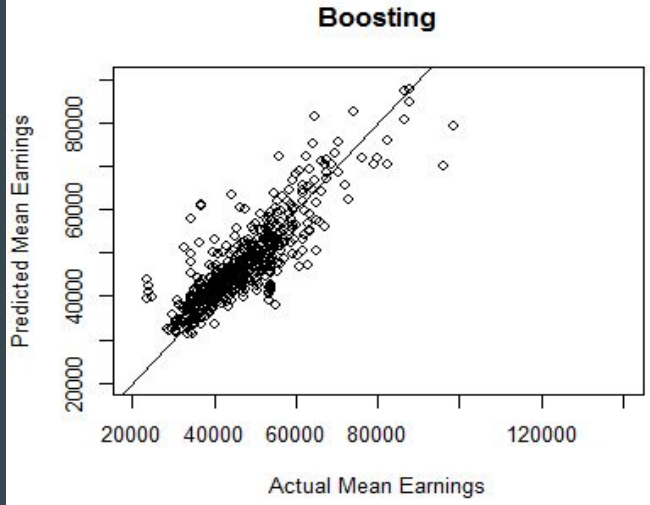
- Boosting uses decision trees as building blocks to construct powerful prediction models.
- Methods:

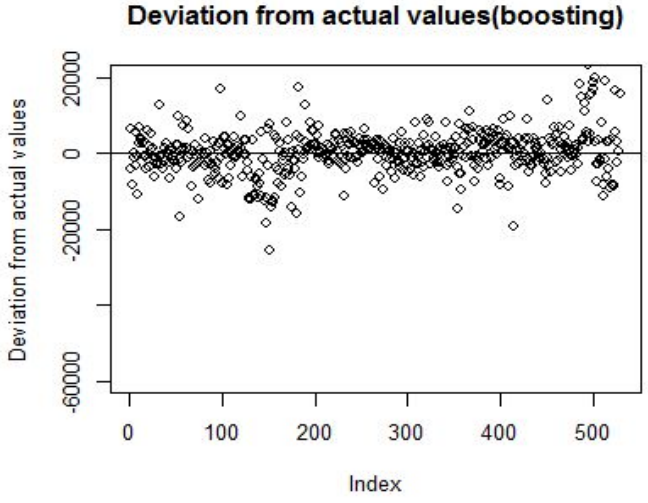
Initially, the boosted model $f(x)$ is set to be zero and residual $r_i = y_i$ for all i in the training set. Then decision tree is built to the training data. Update model $f(x) \leftarrow f(x) + \lambda f^b(x)$ and residuals $r_i \leftarrow r_i - \lambda f^b(x)$ after each tree is built.

The final model for the boosting is then: $f(x) = \sum \lambda f^b(x)$


Lasso
Predicted Mean Earings
20000
40000
60000
80000
20000
40000
60000
80000
120000
Actual Mean Earnings (Testing Data)


Deviation from actual values(lasso)
Deviation from actual values
20000
0
-20000
-60000
0
100
200
300
400
500
Index


Boosting
Predicted Mean Earnings
20000
40000
60000
80000
20000
40000
60000
80000
120000
Actual Mean Earnings


Deviation from actual values(boosting)
Deviation from actual values
20000
0
-20000
-60000
0
100
200
300
400
500
Index

Linear –Final thoughts

- Did we meet our goal ?(accuracy)

  Lasso regression has better prediction performance than linear regression but the predictions are not accurate especially when the mean incomes are higher than $70000. In comparison with Lasso, boosting has better prediction .

- features work? How could things have been done better ?

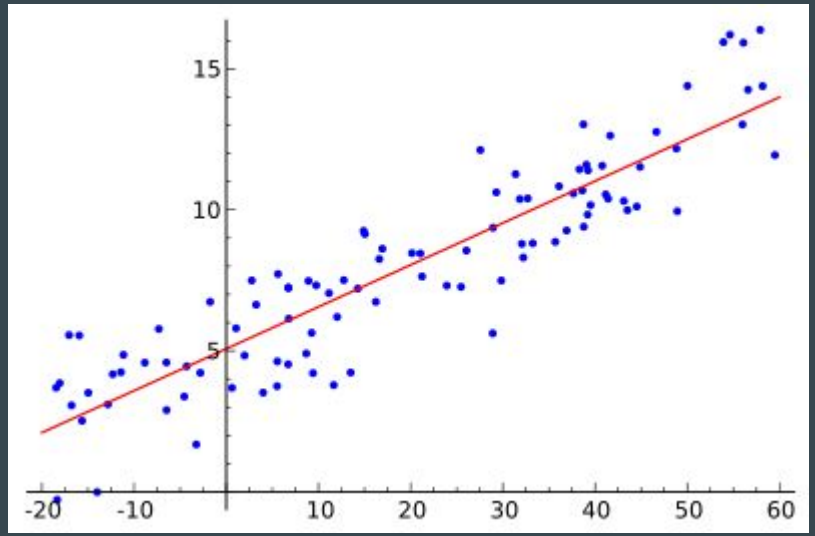
  After feature selections, test MSE is smaller.

- How could things have been done better ?

  Some other feature selections criteria should be used eg. AIC.

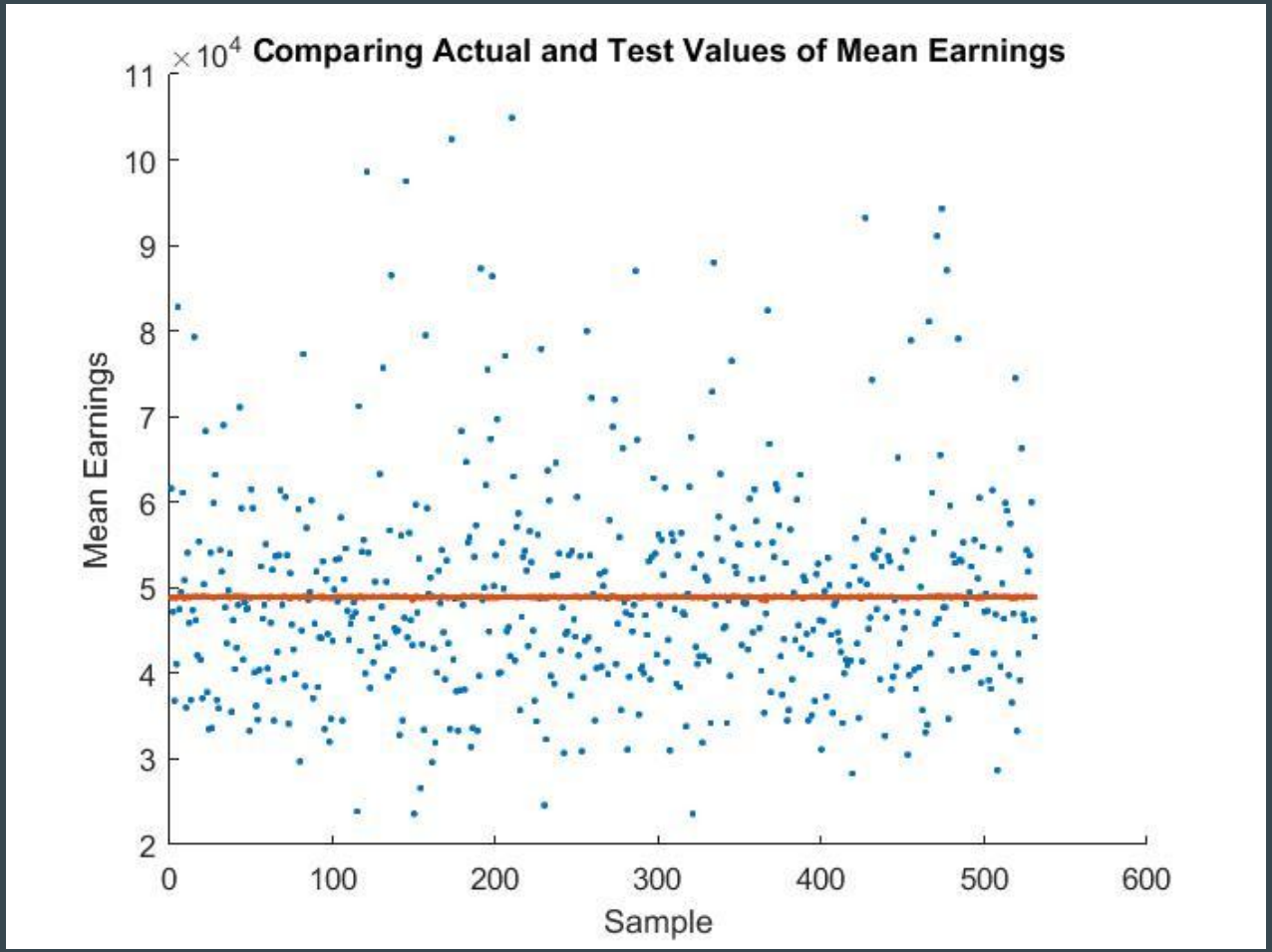
Linear Regression–Final thoughts

- Advantage:
  - simplicity, effectiveness
  - completeness
- Disadvantage:
  - limitations in shapes
  - curve relatively slowly..



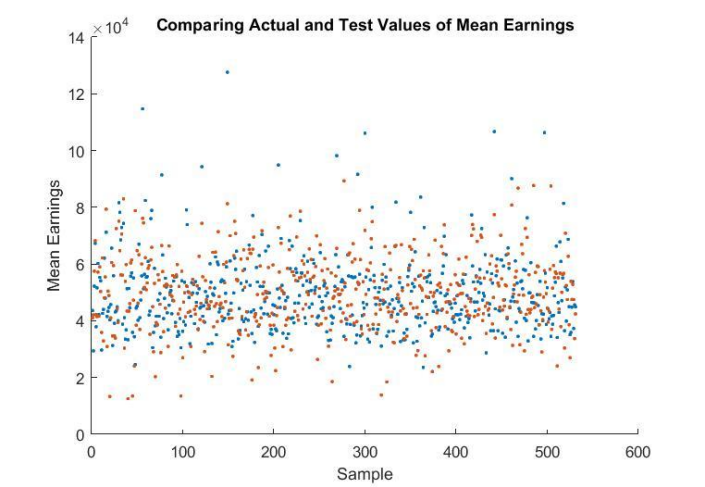
Neural Network

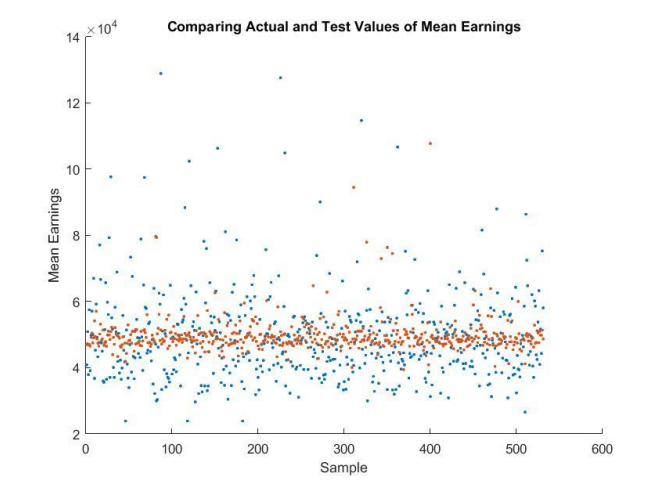
- Goal: Create an artificial neural network that functions as a predictor for earnings after college.
- Initial Approach: Train many neural networks with varying numbers of hidden nodes and layers to find optimal architecture.
- Architecture: No specific architecture with lowest error value for every trial, but 2 hidden layers with 10 nodes each seemed to be consistently among the lowest.
- Results: Low testing and training error, but severe overfitting due to number of inputs, number of outputs, and complexity of neural network.

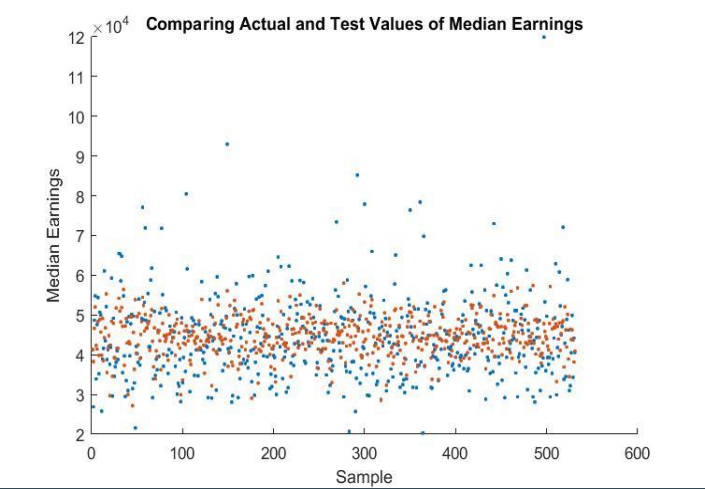

Comparing Actual and Test Values of Mean Earnings
×10^4
11
10
9
8
7
6
5
4
3
2
Mean Earnings
0
100
200
300
400
500
600
Sample

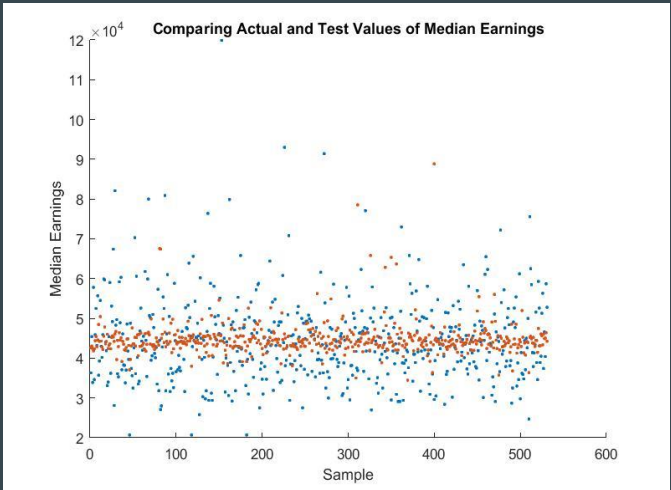
Neural Network

- New Approach:
  - Limit Outputs: Instead of predicting six values related to earnings after college, we chose the two that we felt were most significant: mean earnings and median earnings.
  - Limit Inputs: Ignored 24 input values that represented percentage of student body in given majors. New input set had 22 features instead of 46.
  - Reduce Complexity: Using reduced input and output sets we trained several neural networks with simpler hidden layer architectures to reduce the network's ability to produce complex functions.
- Results: Reduced overfitting. Different trials yielded different results, but new architecture improved on the initial approach.


Comparing Actual and Test Values of Mean Earnings
×10^4
Mean Earnings
Sample


Comparing Actual and Test Values of Mean Earnings
×10^4
Mean Earnings
Sample


Comparing Actual and Test Values of Median Earnings
×10^4
Median Earnings
Sample


Comparing Actual and Test Values of Median Earnings
×10^4
Median Earnings
Sample

Neural Network

- Strength of ANN: Resulted in decent predictor of mean and median earnings after college using features of different universities. Simplified architecture yields less running time.

- Weakness of ANN: Not the most efficient method for finding which features affect earnings most significantly.

K-Nearest-Neighbors

K-Nearest Neighbors is a simplistic non-parametric supervised machine learning method that utilizes the distance between observations and predicts based on majority vote

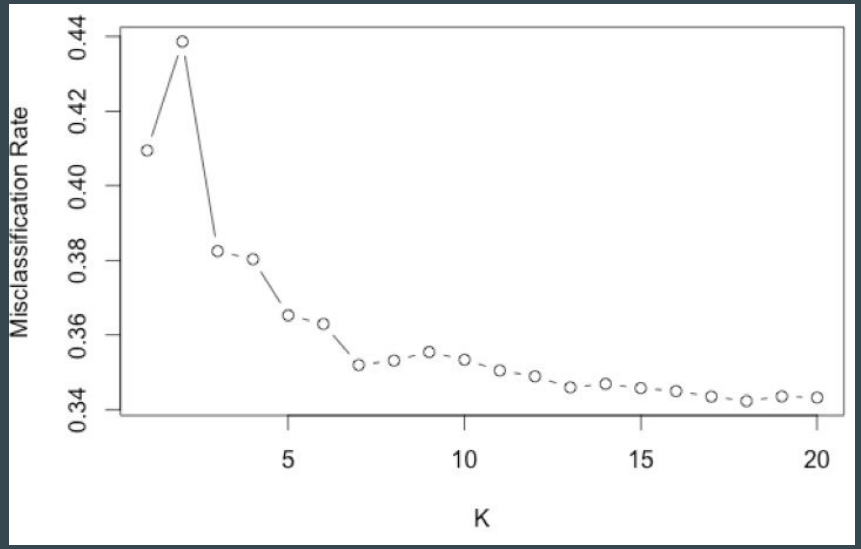
Goal: Predict median earnings after college with K-Nearest-Neighbors utilizing euclidean distance

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Approach: Using K-NN with 10 Fold Cross Validation, predict Low, Medium, and High incomes using K's between 1 to 20

K-Nearest-Neighbors

Average Misclassification Rate over 1000 Iterations



Lowest Average Misclassification rate at K = 18 with 34%. Could only achieve a 66% accuracy despite only using 3 labels (High, Medium, Low).

Advantages of KNN: Non-Parametric, Simple Classifier that works well on basic recognition problems.

Disadvantages of KNN: “Lazy Learner” as it does not learn anything from the training data and simply uses the training data itself for classification. It can be computationally intensive for large amount of observations

Conclusion: Not very accurate for this problem and does not achieve our goals

Random Forest

An ensemble classifier that uses divide and conquer method to differentiate between stronger and weaker predictors

With five groups of output, it has 70% accuracy with 10-fold cross validation

Results are

SAT(38.6%), Family Income(9%)

Admit rate(5.2%), Major in Engineering(3.9%), Major in Health(3.1%), Gender (3.1%), Birthright Citizen(2.9%), Race (2.8%)

SVM

Challenge

- Output is continuous.
- Too many features
- Takes too long to run

Solution

- Feature selection
- Dimensionality Reduction using PCA
- Classify output into binary and five-group outcome

SVM

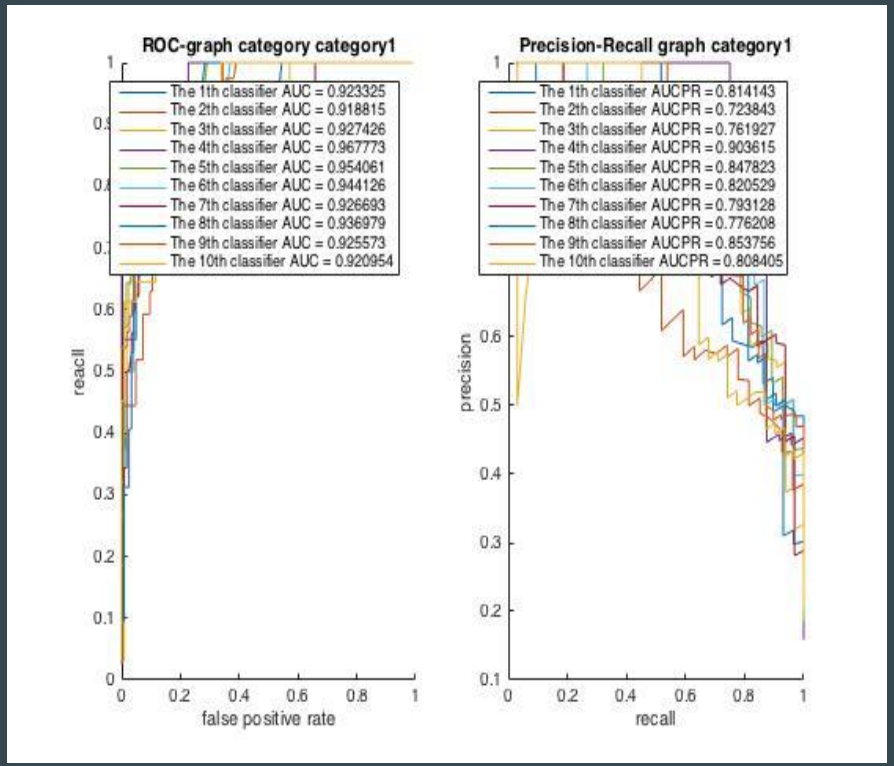
With SMO and newton method, an improved version of SVM, it gives 84.57% with binary output and 48.37% with five classes prediction with 10-fold cross validation

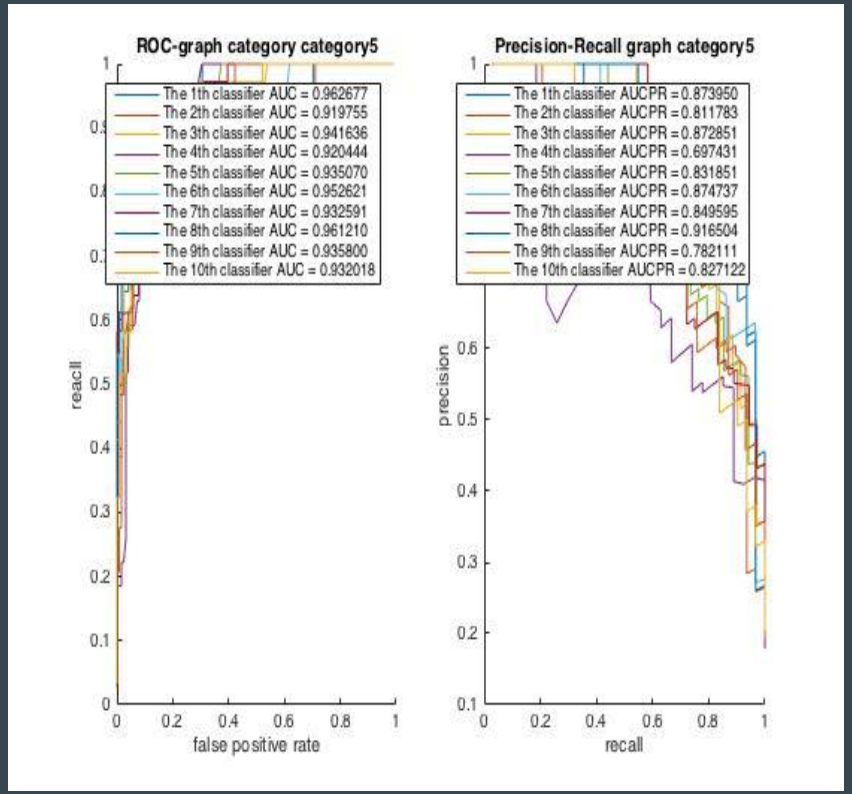
It works particularly well with top 20% and bottom 20% of the earning group

With grouped input using normalized SVM:

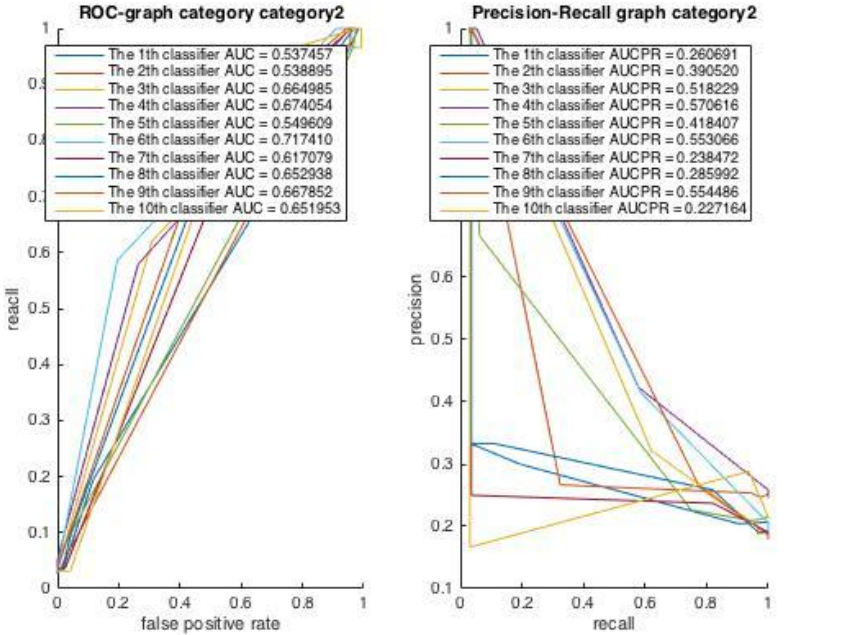
- Family Income, Academic Achievement are the higher (74%-76% for binary)
- Race and Demographic are lower(about 68% for binary)

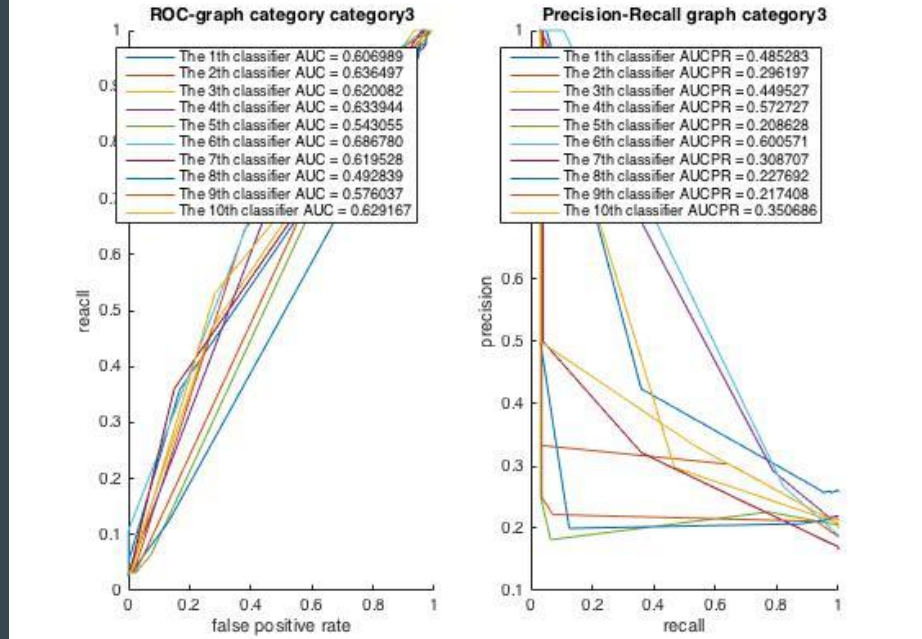
AUC and PR Curve





AUC and PR Curve





Sequential Minimal Optimization (SMO)

max $\sum a_i$ - ½ * $\sum y_i y_j a_i a_j \langle x_i, x_j \rangle$; $\langle x_i, x_j \rangle$ is the kernal function

Conditions: $\sum a_i y_i = 0$ $0 <= a_i <= C$ $w = \sum \alpha_i y_i x_i$ ==> $a_i y_i + a_j y_j =$ constant ; C = 1 in this case

prediction function:

rewrite $f(x) = wx + b$ as $f(x) = \sum \alpha_i y_i \langle x_i, x \rangle + b$

Because $0 <= a_i <= C$ and $0 <= a_j <= C$

If $y_i != y_j$, $L=\max(0,\alpha_j - \alpha_i)$, $H=\min(C,C+\alpha_j - \alpha_i)$ If $y_i = y_j$, $L=\max(0,\alpha_i+\alpha_j-C)$, $H=\min(C,\alpha_i+\alpha_j)$

L, H are the lower and upper bound for a_j

$a_j := a_j - y_j(E_i - E_j) / \eta$

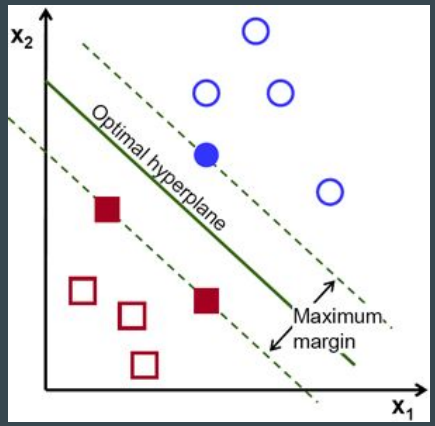
where η is the second derivative given by: $\eta = 2\langle x_i, x_j \rangle - \langle x_i, x_i \rangle - \langle x_j, x_j \rangle$, $E_i = f(x_i)$ - y_i

$a_i := a_i + y_i y_j (\alpha_i(old) - a_j)$

b = - (max:y_i =–1 w*x_i + min:y_i =1 w*x_i)/2

Run until converge: in 100 continuous iterations, there is

1. no change in a_i and a_j
2. the change of ai is less than 10^(-5)
3. $\eta >= 0$



Possible reasons and improvements on SVM

Reason:

Gaps are really small for the middle ones: 36300 40900 45500 51000

Therefore, it is very hard to classify the middle ones.

Possible Solutions:

Try different Cs, try different kernal functions

Change the definition of convergence to a higher standard

Collect more data, Try different partitions for testing set and training set

Conclusion

- Linear Regression had fairly good predictions with accuracies above 70%
- Neural Networks did fair but not good enough to yield good enough predictions
- Support Vector Machine stood out with a high accuracy of 84%
- Random Forest stood out with the second best accuracy of about 70% followed by K-Nearest Neighbor yielded fair results 66%

The results of show that it is possible to predict the average income of a college graduate based on its features.

Potentially better results from better feature selection from original data set

QUESTIONS?