ANNMARIE STOCKINGER

EVALUATION OF KDD CUP 2009 CRM DATA: AN EXPLORATION OF RANDOM FOREST MODELS

THE PROBLEM

- Can we predict the probability that a customer will buy new products?
- How can we do this with the data available?
  - missing data
  - anonymous
  - class bias data

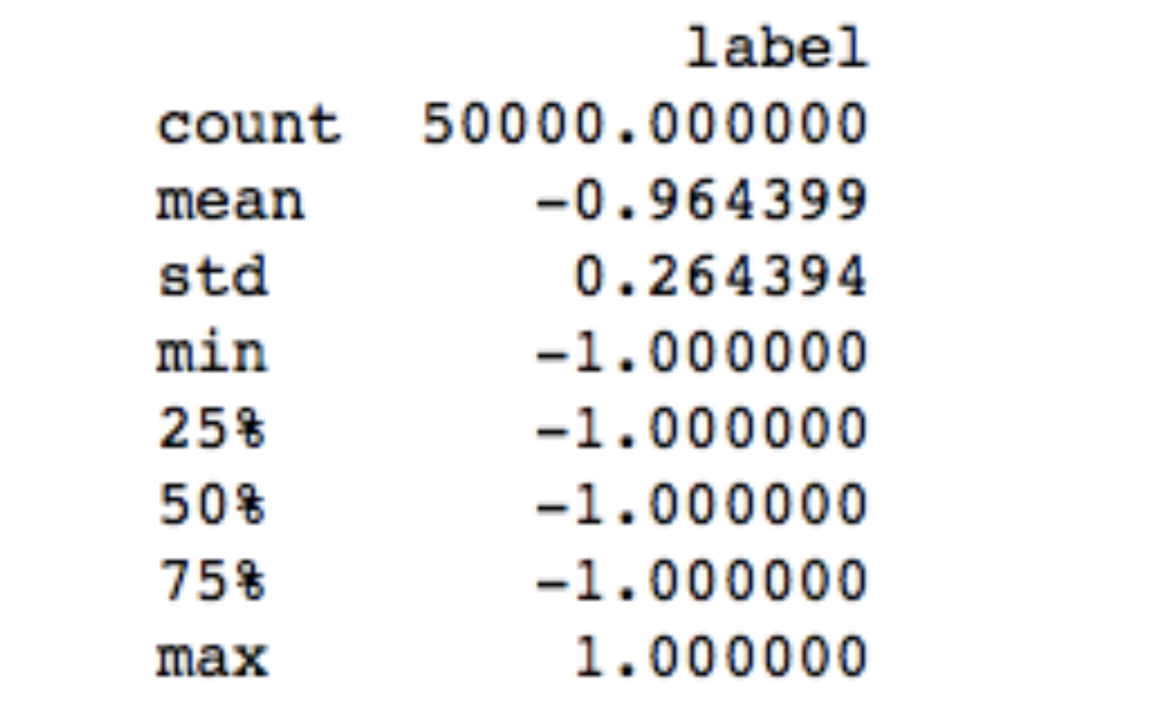
TIME TO PARSE

THE PARSING PROCESS

- Preprocessing
  - imputation of missing data
  - cleaning of columns that were fully NaN’s
  - conversion to csv from database format

EXPLORING THE DATA

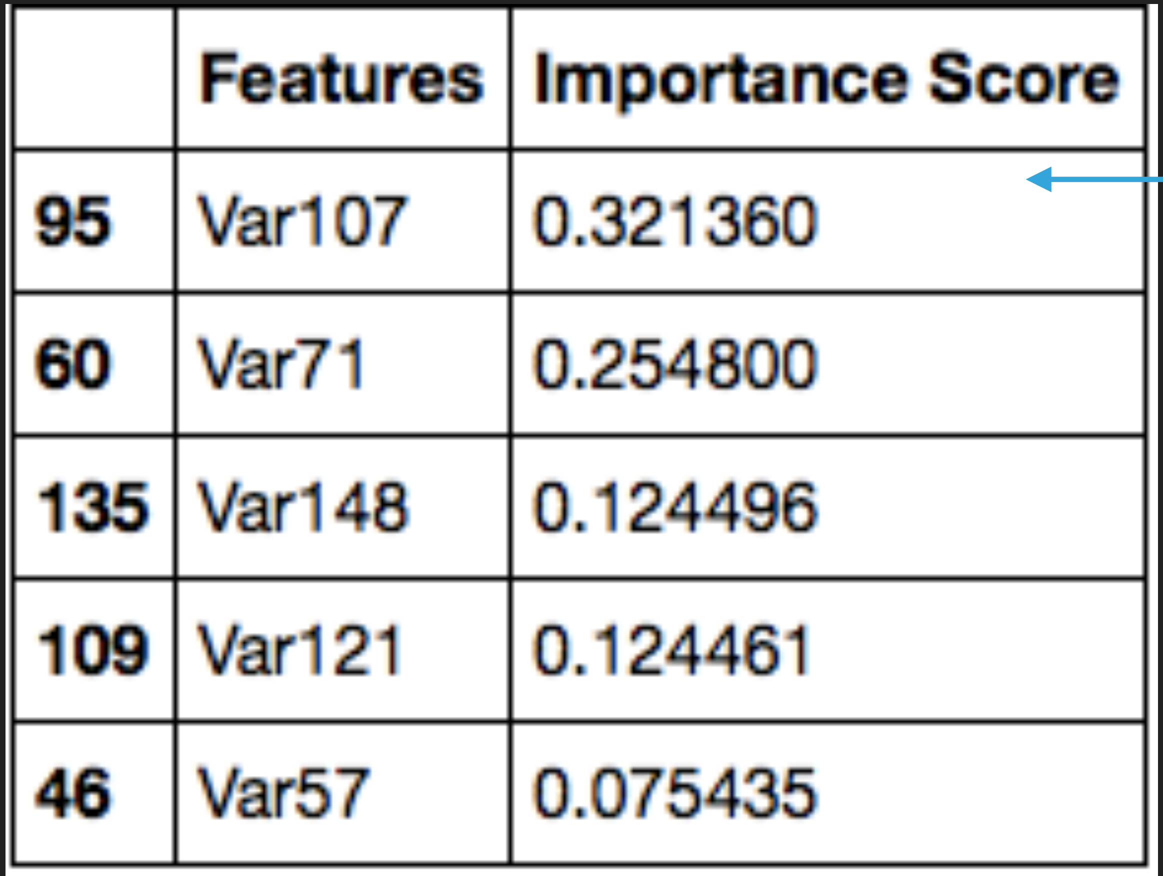
- 5000 rows by 230 columns
- class bias towards -1 (unlikely to buy)
    - very very few people were likely to buy product
    - makes sense from an industry point of view– your change of selling is always lower.
- every row has missing data
- categorical data was unreadable



```
                 label
count   50000.000000
mean       -0.964399
std         0.264394
min        -1.000000
25%        -1.000000
50%        -1.000000
75%        -1.000000
max         1.000000
```

SKLEARN RANDOM FOREST CLASSIFIER

FEATURES



	Features	Importance Score
95	Var107	0.321360
60	Var71	0.254800
135	Var148	0.124496
109	Var121	0.124461
46	Var57	0.075435

Var107 was found to be the most ‘important’ feature.

CHOOSING A MODEL

- random forest
  - handles missing data well
  - you don't need to ‘know’ your features to use it
  - makes sense for future production purposes
- XGboost
  - newer model
  - really cool!!!!!! (not necessarily the best)

WHAT IS XGBOOST

- xgboost is special because of its parameters
- xgboost became popular through papers and kaggle and it was released in 2014
- can drop a math link in slack

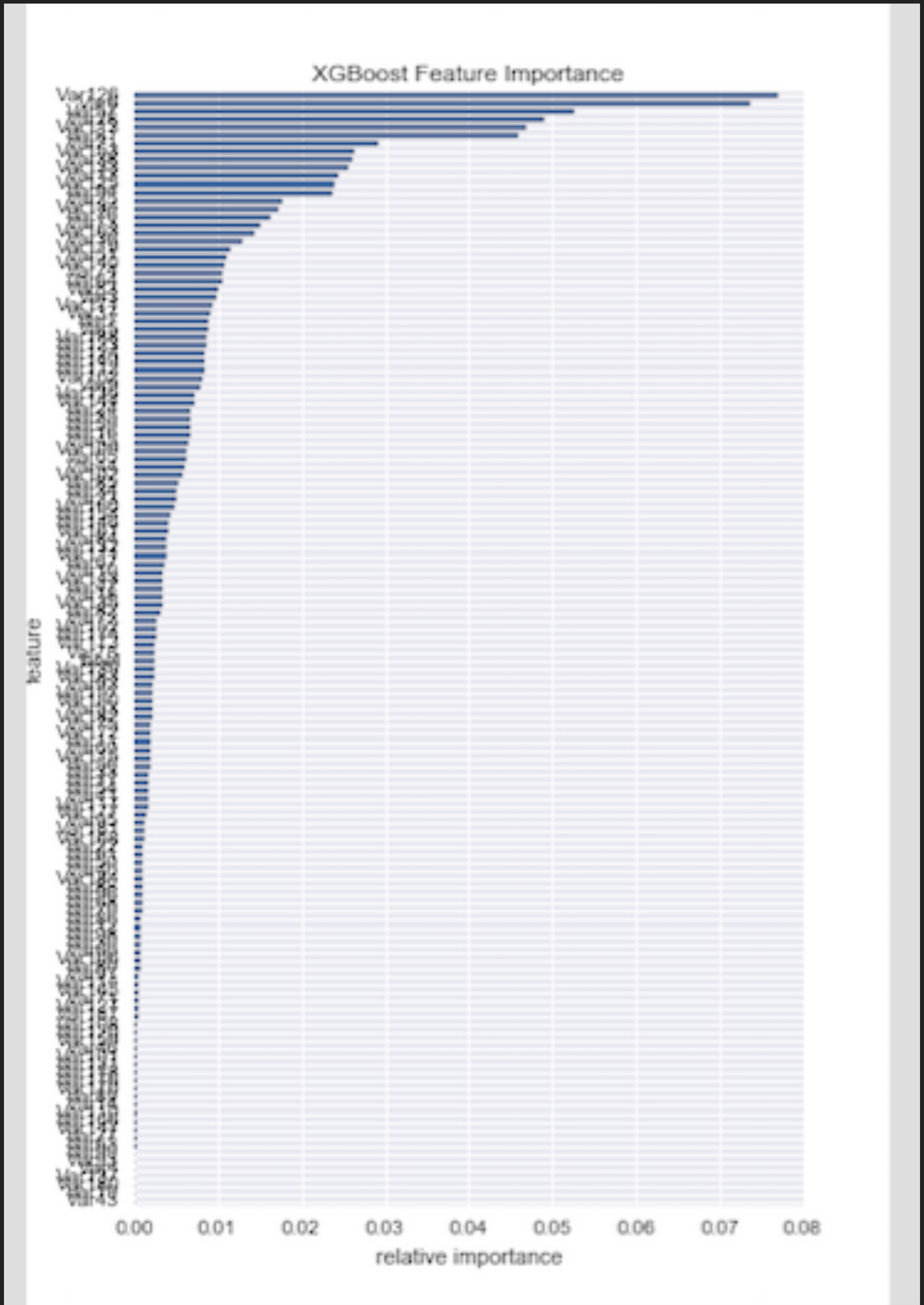
ANALYSIS

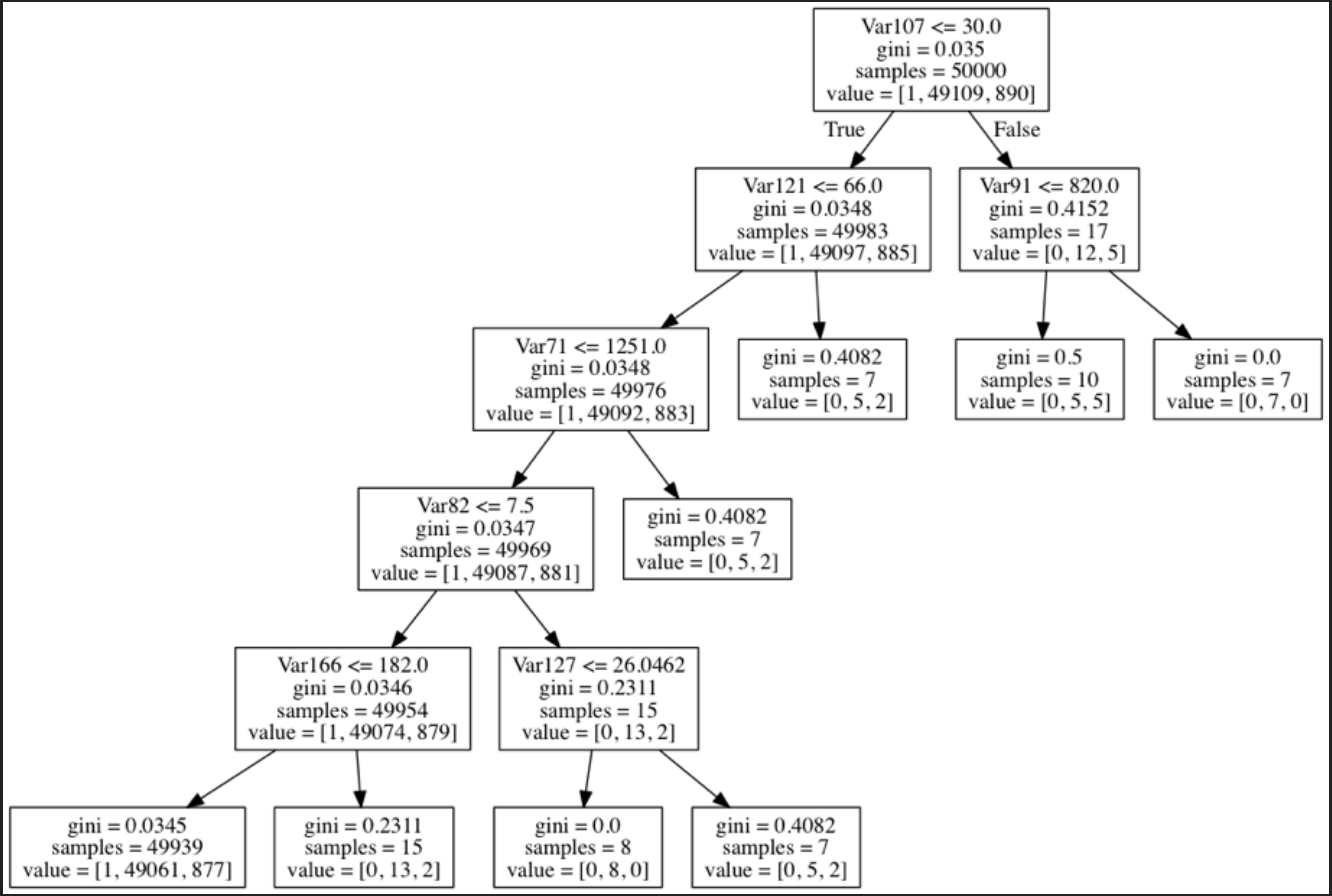
- XGBoost while really cool will require more parameter tuning to avoid overfit and provide more reliable results, however xgboost recognizes Var126 as the most important feature

- requires data to be in a dmatrix format which only takes float32

- some feature have little to no importance

- regularization term will provide more insight




Var107 <= 30.0
gini = 0.035
samples = 50000
value = [1, 49109, 890]
True
False
Var121 <= 66.0
gini = 0.0348
samples = 49983
value = [1, 49097, 885]
Var91 <= 820.0
gini = 0.4152
samples = 17
value = [0, 12, 5]
Var71 <= 1251.0
gini = 0.0348
samples = 49976
value = [1, 49092, 883]
gini = 0.4082
samples = 7
value = [0, 5, 2]
gini = 0.5
samples = 10
value = [0, 5, 5]
gini = 0.0
samples = 7
value = [0, 7, 0]
Var82 <= 7.5
gini = 0.0347
samples = 49969
value = [1, 49087, 881]
gini = 0.4082
samples = 7
value = [0, 5, 2]
Var166 <= 182.0
gini = 0.0346
samples = 49954
value = [1, 49074, 879]
Var127 <= 26.0462
gini = 0.2311
samples = 15
value = [0, 13, 2]
gini = 0.0345
samples = 49939
value = [1, 49061, 877]
gini = 0.2311
samples = 15
value = [0, 13, 2]
gini = 0.0
samples = 8
value = [0, 8, 0]
gini = 0.4082
samples = 7
value = [0, 5, 2]

MOVING FORWARD

- tune xgboost parameters
- get AUC functioning
- test model on real CRM data
- train model to more predictors (churn, up-selling)
- create web application and *hopefully* go the startup route

SELF-ANALYSIS

- WENT WELL
  - attempted new model (xgboost) and was able to clearly see the differences in feature selection
- DISAPPOINTING/FUTURE CONCERNS
  - while the project is complete there is still a lot to learn and analyze
  - xgboost lacks significant laymen accessible documentation and this made it difficult to get the model running