PREDICTING CONSUMER CREDIT DEFAULT

KAGGLE CHALLENGE

EIGENAUTS

Bernard Ong | Emma (Jielei) Zhu | Nanda Rajarathinam | Trinity (Miaozhi) Yu

NYC Data Science Academy Capstone Project
September 2016

CONSUMER CREDIT DEFAULT

KAGGLE CHALLENGE

Improve on the state of the art in credit scoring by predicting the probability that somebody will experience financial distress in the next two years.

Credit scoring algorithms, which make a guess at the probability of default, are the method banks use to determine whether or not a loan should be granted. This challenge seeks to improve on the state of the art in credit scoring, by predicting the probability that somebody will experience financial distress in the next two years.

The goal of this challenge is to build a model that borrowers can use to help make the best financial decisions.

Predict : Serious Delinquency in 2 Years (Default Risk)

Features	Features
Revolving Utilization of Unsecured Lines	**Debt Ratio**
Age	**Monthly Income**
Number of Time 30-59 Days Past Due	**Number of Open Credit Lines & Loans**
Number of Time 60-89 Days Past Due	**Number of Real Estate Loans or Lines**
Number of Times 90 Days Late	**Number of Dependents**

GOALS

- **Achieve the highest Area Under the Curve (AUC) Score on the Kaggle Private Leaderboard**
- **Achieve the Top 5% position placement in the Kaggle Private Leaderboard**

DATASET SIZE

- **Training Dataset = 150,000 Observations**
- **Test Dataset = 101,530 Observations**

SWOT ANALYSIS

A SWOT analysis allows our team to organize our thoughts and focus on leveraging our greatest strengths, understand our weaknesses, take advantage of opportunities, and have awareness of any threats to our undertaking.

SWOT allows us to put ourselves on the right track right away, and saves us from a lot of headaches later on.

STRENGTHS
Leveraging what we already know

WEAKNESSES
Areas that we need to improve upon

OPPORTUNITIES
Chance to apply & learn from experience

THREATS
Risks that we need to mitigate and manage

STRENGTHS

- Experience to leverage advanced stacking techniques and Agile
- A synergistic team with very complementary skills and experiences
- Lessons learned from previous Kaggle challenge can be applied
- Experience in the use of Agile process to parallel track work queues

WEAKNESSES

- Extreme time constraints would limit the scope/depth of exploration
- Unfamiliarity with new tools and models will limit firepower
- Learning while executing will slow down the entire process
- Sparse resources on some of the newer technologies employed

OPPORTUNITIES

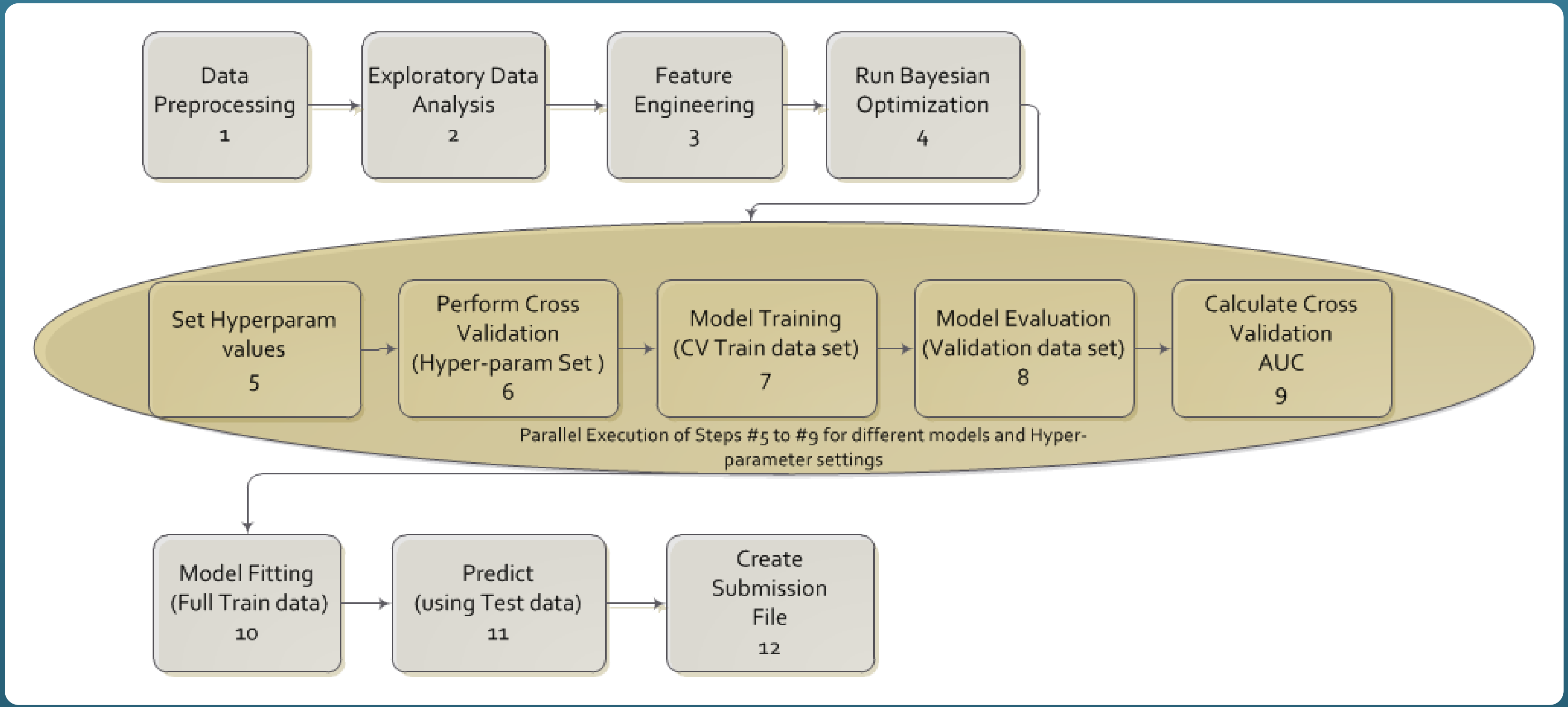
- Learn how to strategize, optimize, and fine-tune models, algorithms, and parameters to achieve the best default prediction possible
- Gain real time experience using Bayesian Optimizers
- Explore Deep Learning (Theano/Keras) in predicting default

THREATS

- Small dataset size presents tremendous challenges on generalization that would impact modeling and ultimately, prediction accuracy
- Extremely tight tolerances in the AUC scores in the 10,000th decimal
- Top 5% target goal presents a high risk – no idea how feasible this is

MACHINE LEARNING

AGILE PROCESS



With the constraint of time and resources, it is critical that the right process is utilized to maximize both our throughput and output. We employed an Agile process adapted for Machine Learning that allows us to parallelize our model build, train, validate, test, predict, and scoring cycles quickly in an iterative fashion.

MANAGING COMPLEXITY

MODEL EXECUTION STRATEGY

We employed a parallel tracking process where all modeling methods were being performed at the same time. Each and every time a better setting is found using automated optimization, it feeds up the entire process cycle and synergies are gained instantaneously.

Voting and Stacking Blended Models

6-Model Combination
(2) Gradient Boosting
(2) Random Forest
(1) Adaboost
(1) Restricted Boltzman Machine

Voting Classifier Models

12-Classifier Voting Model
(7) Gradient Boosting
(1) Naïve Bayes
(3) Random Forest
(1) Adaboost

Advanced Stacking Models

2-Layer Stacking Model

Base Classifiers
(1) Gradient Boosting
(1) XGBoost

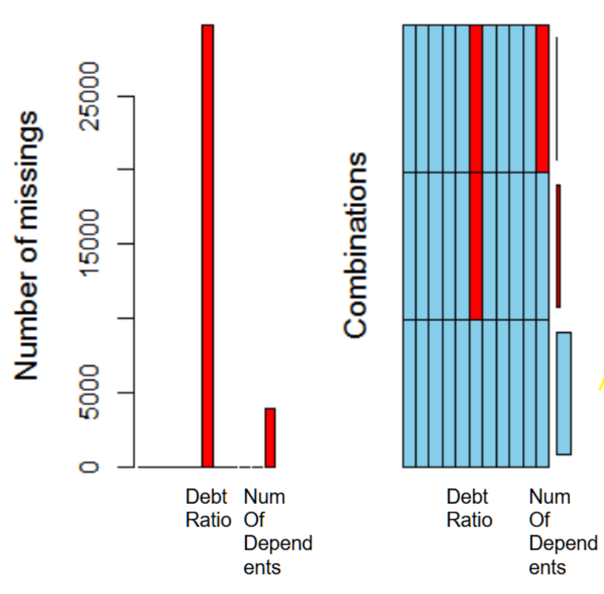
Meta Classifiers
(1) Logistic Regression

Single and Ensemble Models

Single Model
Logistic Regression
Naïve Bayes

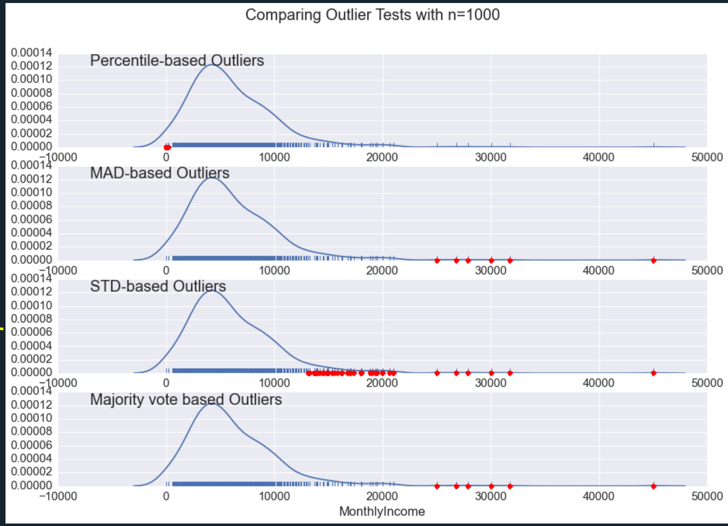
Ensemble Model
Gradient Boosting
XGBoost

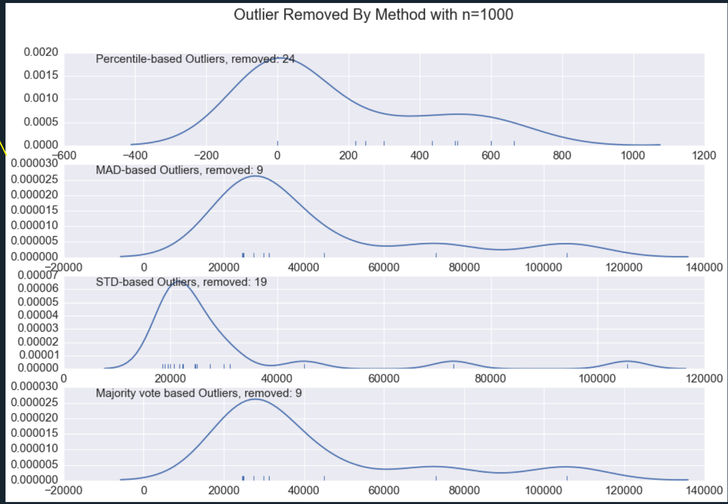
FEATURE ANALYSIS

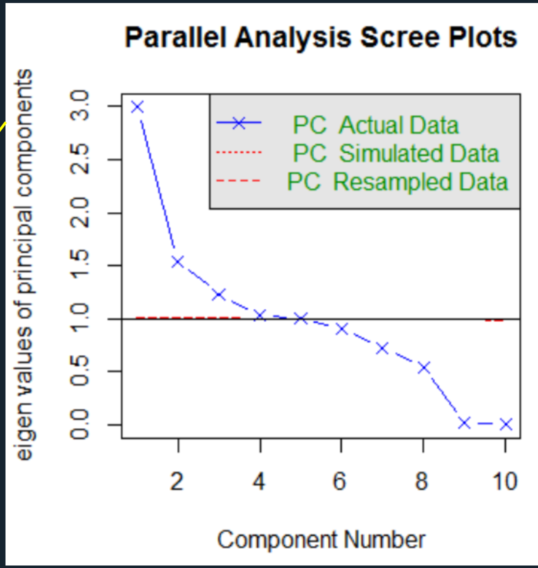


- "**Monthly Income**" has 29,731 NAs (~20%)
- "**Number of Dependents**" has 3,924 NAs (~3%)

Outliers
To filter or not to filter, that is a question…
almost all variables have outlier problems



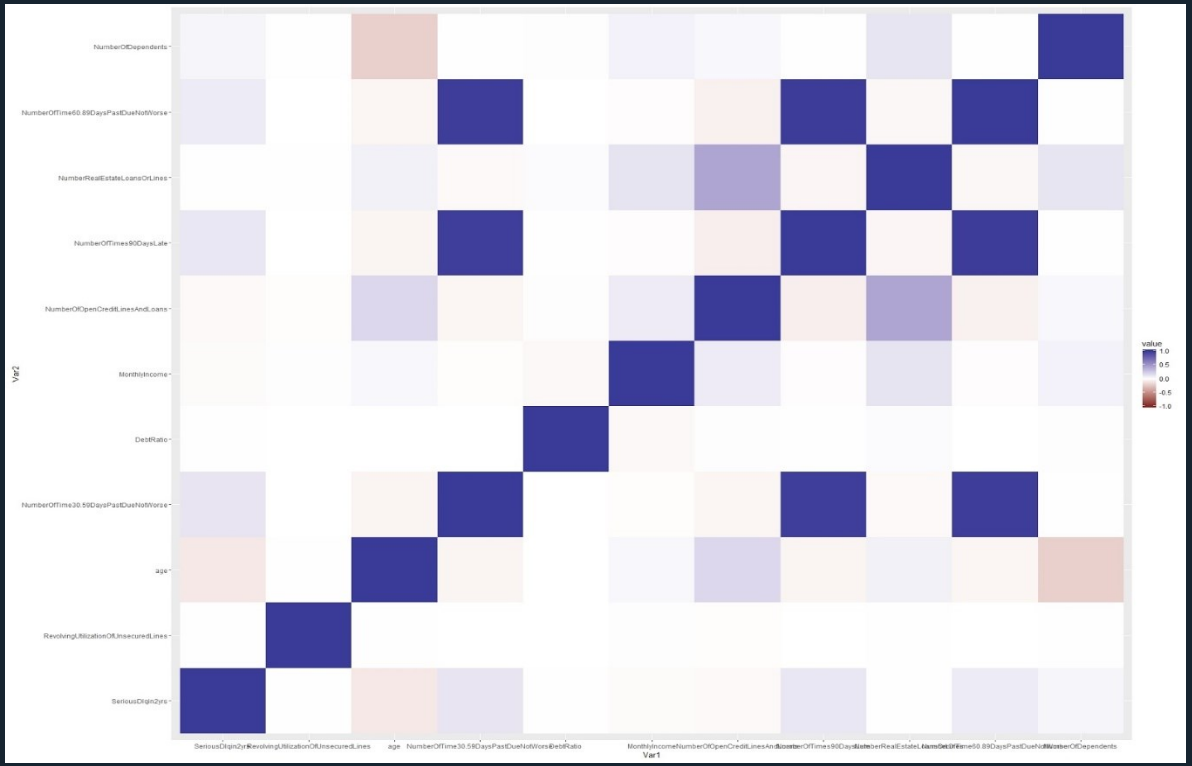


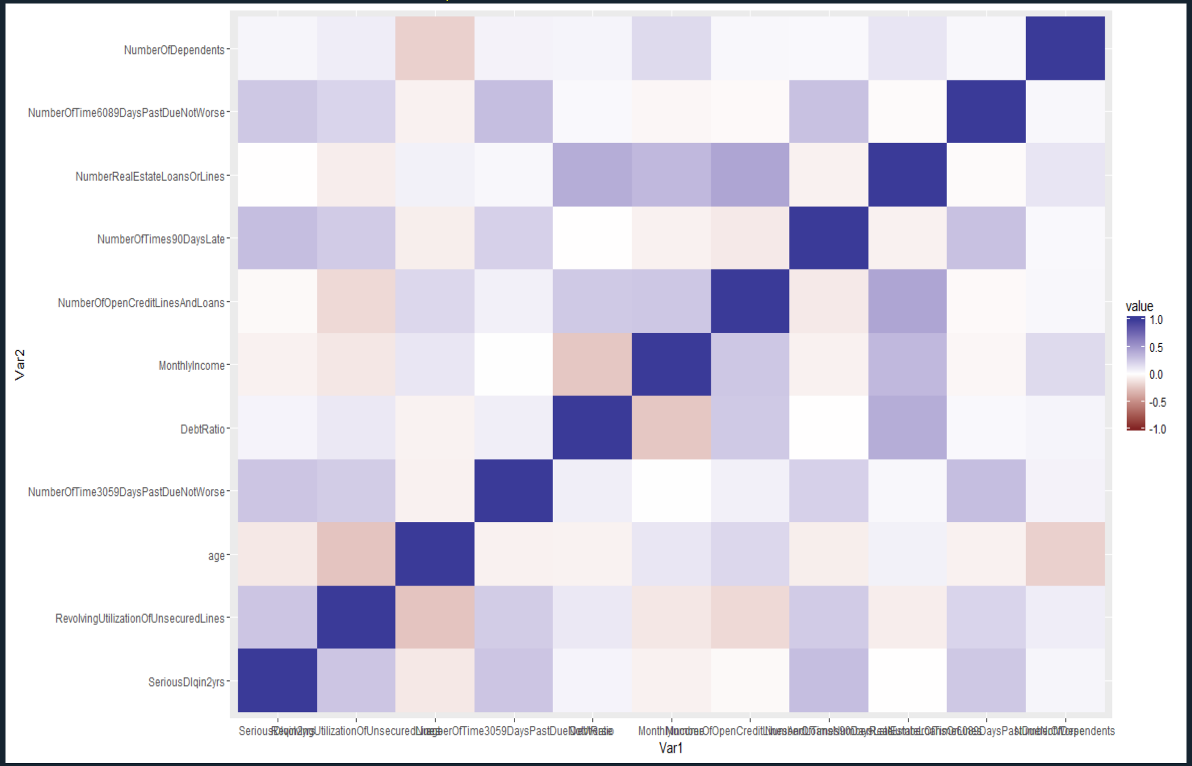


- **PCA** suggests choosing 5 principal components, which explains only 66% of the total variance
- We only have 10 features, thus PCA may not work very well

FEATURE CORRELATION
HEATMAP

- Correlation no longer exists after the outliers are filtered



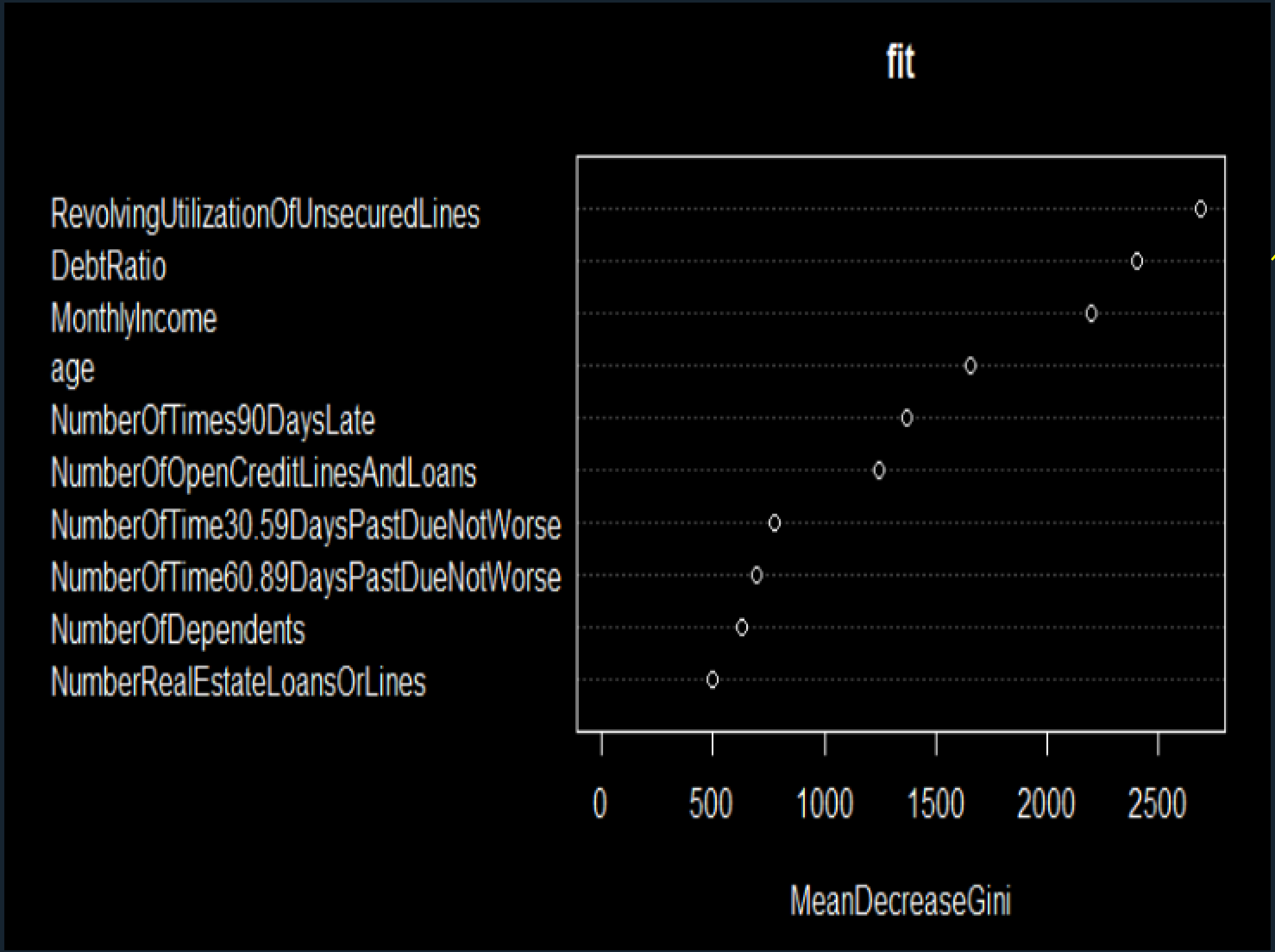


Number of 30-59 Days Past Due

Number of 90 Days Past Due

Number of 60-89 Days Past Due

VARIABLE IMPORTANCE



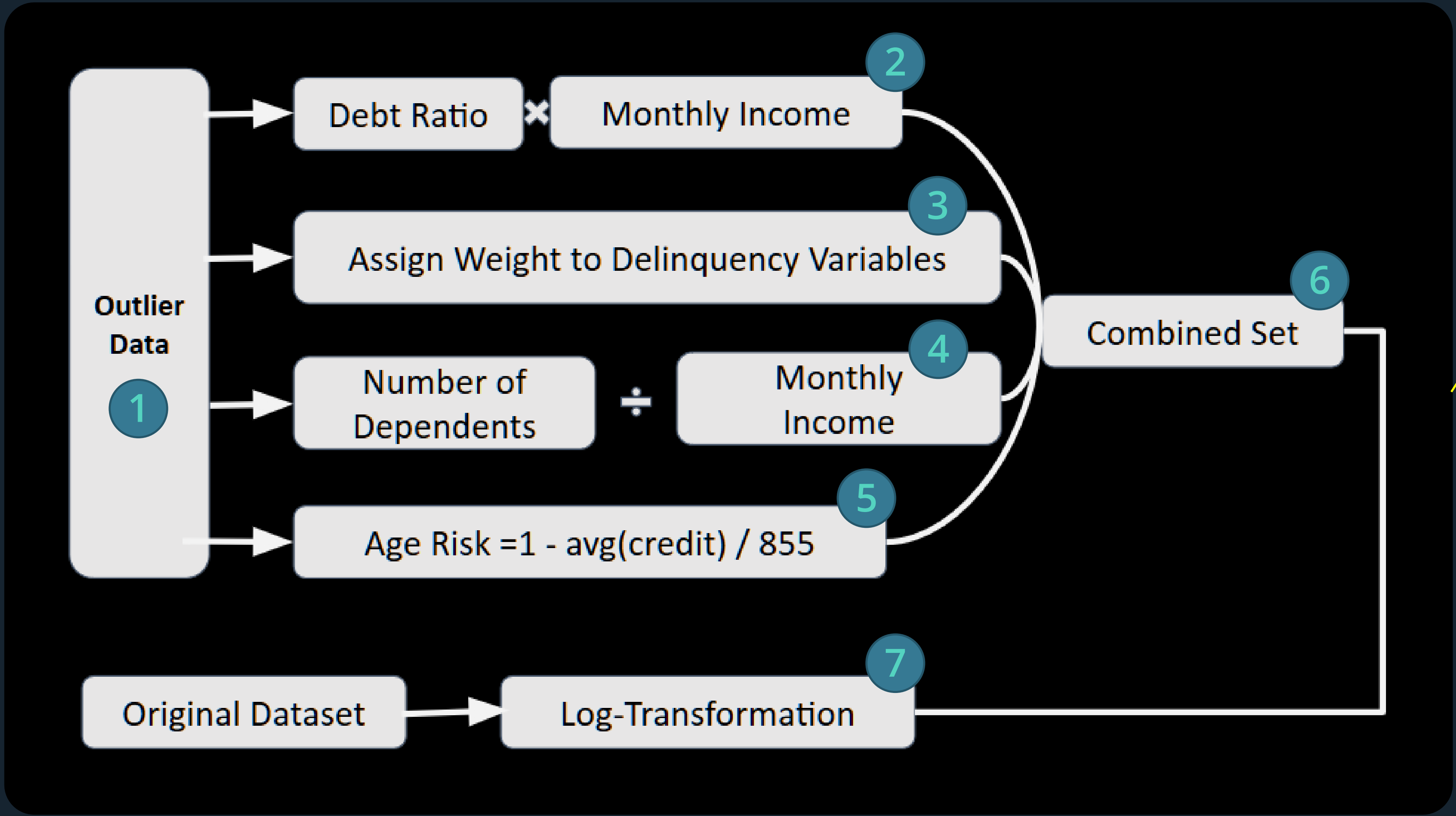
(3) Most Important Variables

“RevolvingUtilizationOfUnsecuredLines”

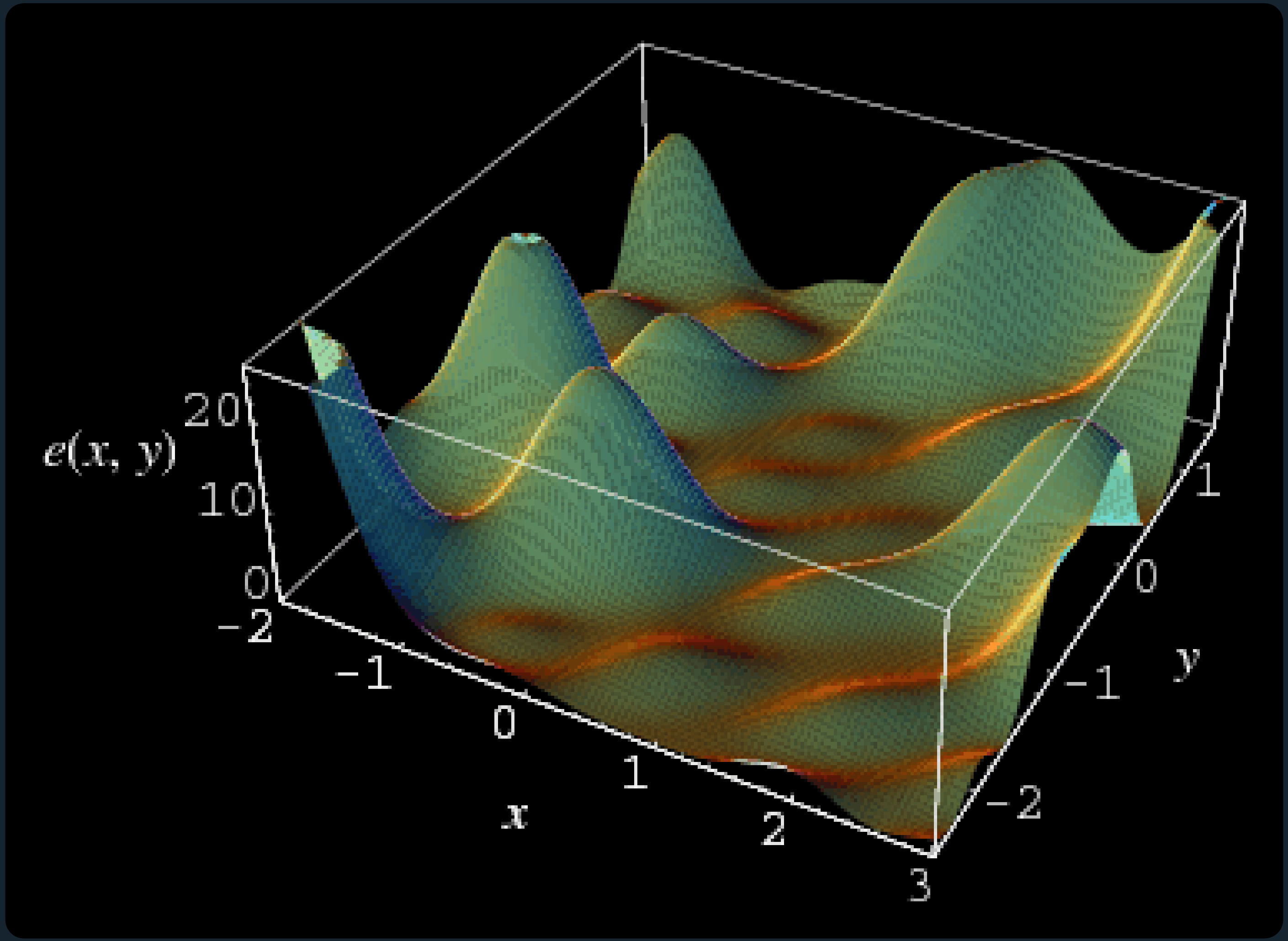
“DebtRatio”

“MonthlyIncome”

FEATURE ENGINEERING



- Created 7 training data sets and 7 test datasets
- **Results:** Naive Bayes and Logistic regression AUC scores improved from ~0.7 to ~0.85
- However, for tree-based methods, the new features did not improve the scores



Bayesian Optimization

- Machine learning algorithms require careful tuning of learning parameters and model hyperparameters.
    - This tuning is often a 'black art' requiring expert experience, rules of thumb, or sometimes brute-force search.
- Automated approaches help expedite the determination of optimal combination of parameters to fit a given learning algorithm/ model to the data at hand
- 4 types of hyperparameter optimization
    - grid search
    - random search
    - gradient-based optimization
    - bayesian optimization

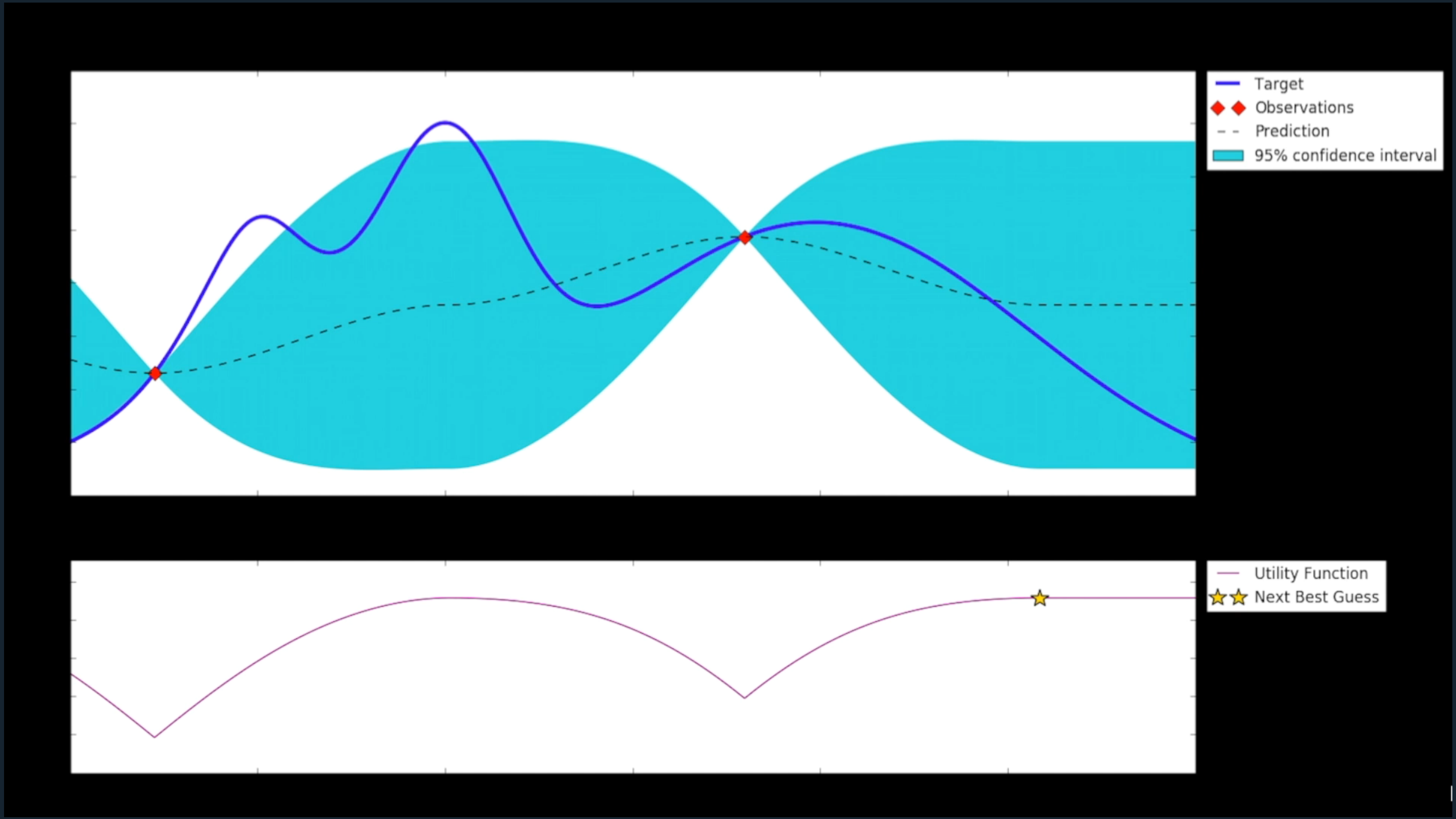
PARAMETERS vs HYPERPARAMETERS

Learned from Fitting Models

Fixed during Model Fitting

	Parameters	Hyperparameters
Ridge & Lasso Regression	• Beta coefficients	• Regularization parameters (λ)
Support Vector Machines (SVM)	• Beta coefficients	• C
Tree-Based Methods	• Which feature to split on at each interval node • At what threshold • …..	• Criterion ("gini" vs "entropy") • Number of trees • Maximum depth • …..

BAYESIAN OPTIMIZATION – bayes_opt



STACKING

- Stacking combines the Base Classifiers in a non-linear fashion. The combining task, called a meta learner, integrates the independently computed base classifiers into a higher level classifier, called a meta classifier, by learning over a meta-level training set
- Stacking model typically uses multiple layers for combining the probabilities and require more careful tuning of the parameters as the number of layers increase
- Stacking model generally works well with various combinations of algorithms (including XGBoost, Random Forest, Boosting)

VOTING

- Voting achieves the classification of an unlabeled instance according to the class that obtains the highest number of votes (the most frequent vote)
- The concept of layering does not apply to a Voting model and the final probability / label prediction is obtained using Majority Voting
- Voting may not work with certain combinations of algorithms
- For example, in our scenario, Voting classifier gave an error when XGBoost was used as one of the classifiers because XGBoost is not part of the Python Scikit-Learn library

Deep Learning

THEANO / KERAS NEURAL NETWORK

- Tried sequential neural network
- Tuned
  - Number of epochs
  - Batch size
  - Initialization functions
  - Optimization functions
- Poor performance
  - Best Kaggle ranking: 584

USE AGILE CHUNKING | FAIL FAST | KEEP ACCURATE LOGS | FOLLOW THE PROCESS

PATHWAY TO THE TOP SCORE

ONE

BASIC SINGLE & ENSEMBLE MODEL TESTING TO UNDERSTAND DATA HEALTH & HEARTBEAT, & INTERACTION WITH BEST MODEL

TWO

PARALLEL FEATURE ANALYSIS & ENGINEERING RESULTS FEEDING TRAINING & TESTING PIPELINE

THREE

PARALLEL BAYESIAN OPTIMIZATION AGAINST TRAINING DATA TO GET RELATIVELY BEST PARAMS & HYPERPARAMETERS

FOUR

PARALLEL TRACK MODEL STACKING & VOTING CLASSIFIER TRAINING & TESTING TO VALIDATE INITIAL HYPERPARAMETERS SETTING

FIVE

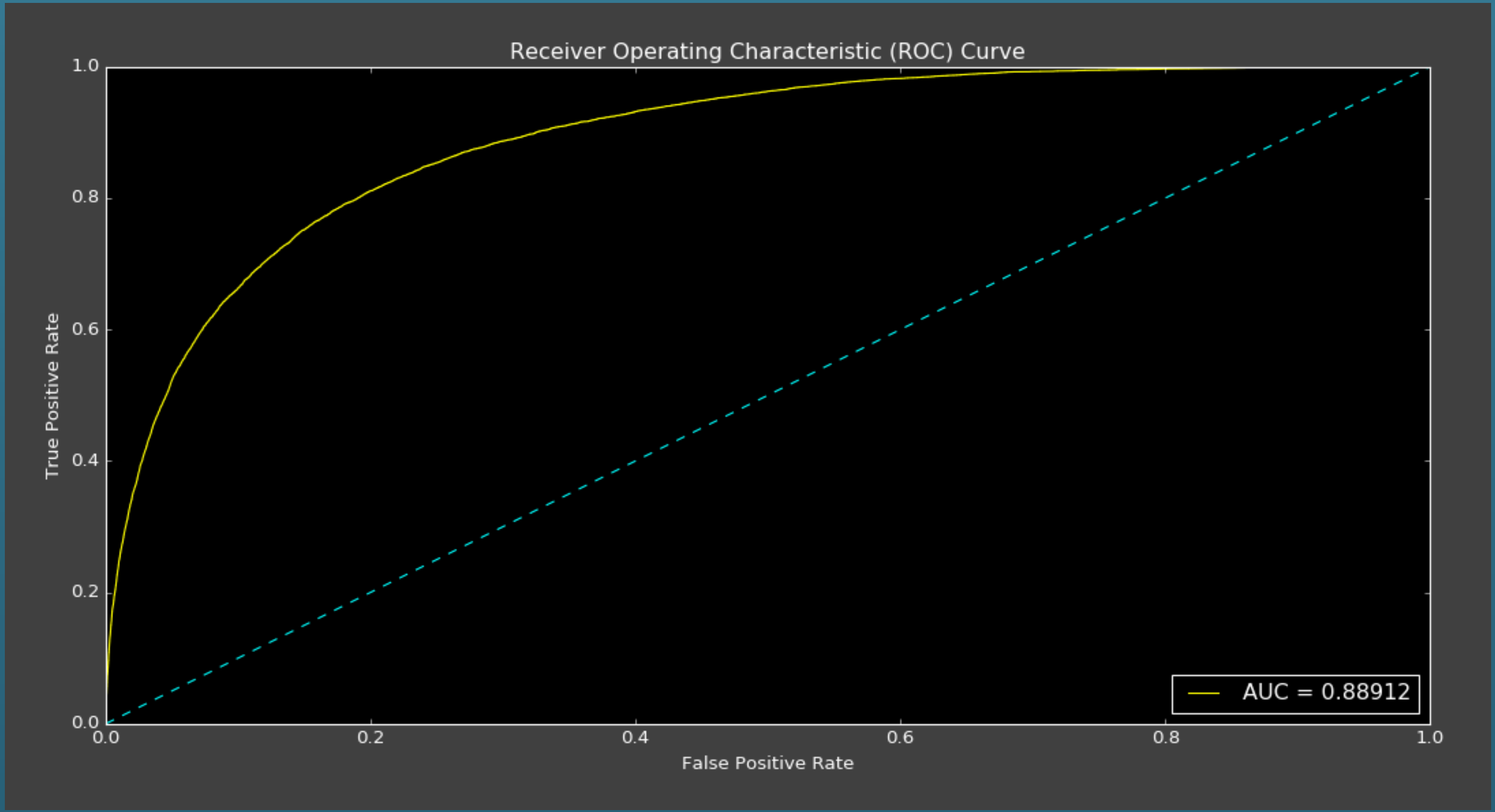
INTENSIVE CROSS VALIDATION PROCESS TO ENSURE SETTINGS PRODUCE CONSISTENT HIGH SCORES WITHOUT OVERFITTING, PERFORM FINE-GRAINED MANUAL TUNING

SIX

USE BEST CROSS-VALIDATION RESULTS TO PRODUCE SUBMISSION FILE FOR KAGGLE SCORING, THEN REPEAT CYCLE UNTIL TOP SCORE ACHIEVED

FINAL KAGGLE

AUC ROC CURVE



Tradeoff between sensitivity (TPR) vs 1-specificity (FPR). At AUC = 0.88912, it is quite good (almost borderline excellent). Curve follows the left-hand border and top border of the ROC closely, indicating a fairly good accuracy test.

RESULTS & FINDINGS

- The models seemed to **perform well** without the missing values being imputed than when imputing the missing values
- Stacking and Voting and the **combination** of the two models generally tend to have very high predictive power compared to plain Ensemble models
- Feature Engineering improved the AUC score for single models (Naive Bayes and Logistic Regression) from ~0.7 to ~0.85 but did not have much impact on the Tree based methods
- The incremental increase in the predictive accuracy **(AUC) is of the order of 0.0001** as we move towards the top of the Kaggle leaderboard (top 2%) and the tuning gets a lot harder

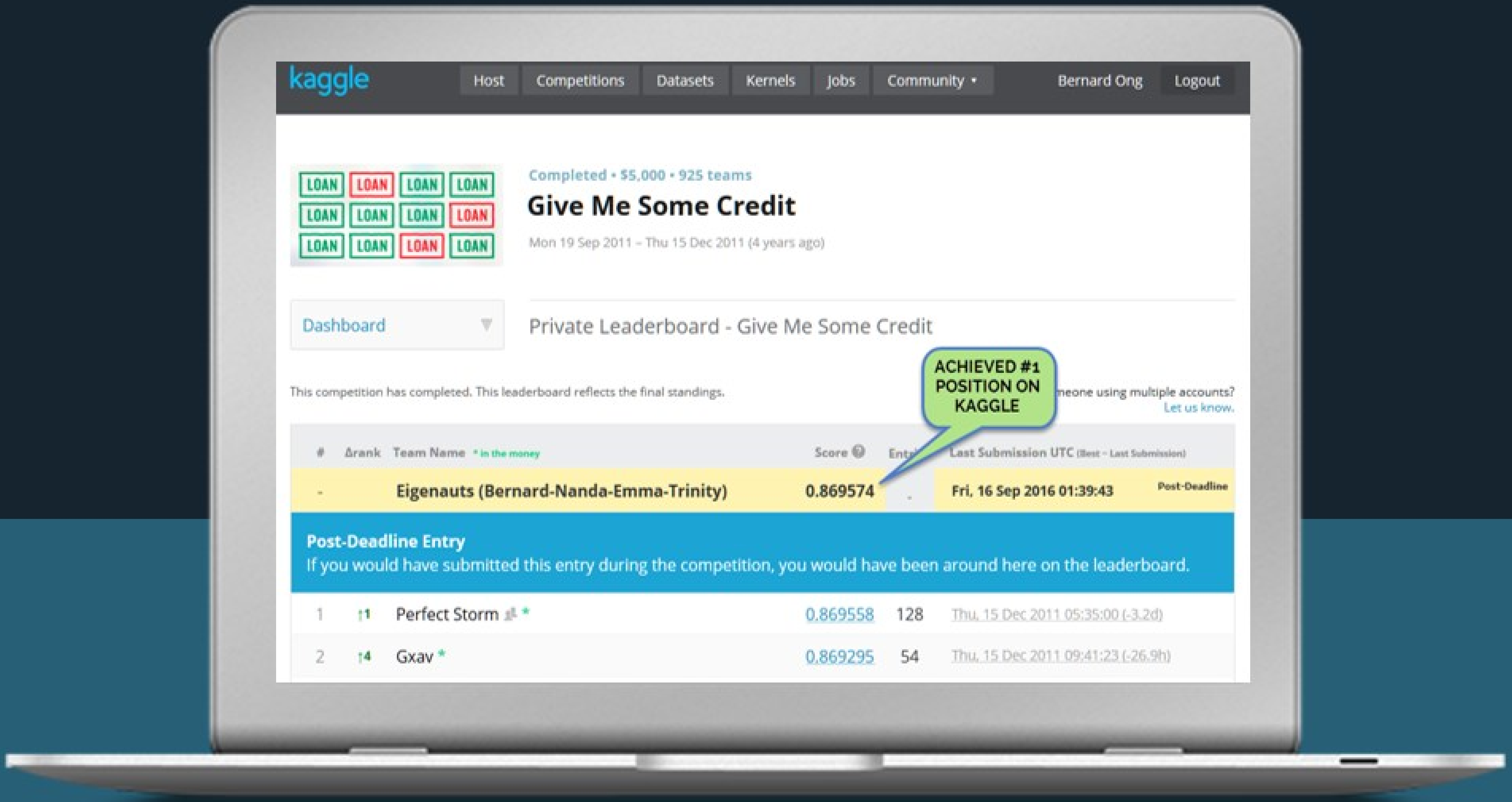
LESSONS & INSIGHTS

- Hyperparameter tuning is a very **time consuming process** and it is better to have the team split this effort and work in parallel
- Cross Validation is **very critical** and it is worth spending time testing the impact of various folds on the model accuracy
- The model needs to be tuned at a much more **granular level** as the dataset gets smaller in size (both in terms of number of features and observations)
- Following an **Agile parallel process** has continued to be a proven factor for maximizing success
- Hyperopt works better when tuned **one parameter at a time** than multiple parameters being tuned simultaneously. The best combination of optimal parameters were obtained using the above approach

FUTURE STEPS

Tune the parameters for Deep Learning using Theano / Keras and compare the predictive accuracy and performance against Stacking / Voting models.

Explore the possibility of adding new polynomial and transformed features, and evaluate the predictive accuracy.



TOP SCORE ACHIEVED

We not only surpassed our original goal to get on the Top 5% position, the team actually beat the high AUC ranking and achieved the #1 spot on the Kaggle challenge in 2 weeks.

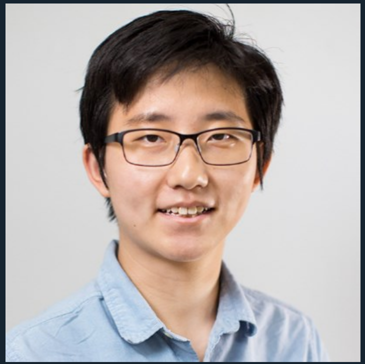
MEET TEAM EIGENAUTS

BERNARD ONG

EMMA (JIELEI) ZHU

NANDA RAJARATHINAM

TRINITY (MIAOZHI) YU



A PASSION FOR DATA

A dynamic team of Data Scientists that work very well together, ready to take on whatever challenges that come their way. The team is composed of an eclectic mix of a seasoned Executive, a Management Consultant, a creative problem solver and a theory-oriented problem solver. They do what it takes to excel in their chosen fields and are relentless in their pursuit of taking on innovative projects that exemplify their passion in working with data.

NYC Data Science Academy – Fall 2016 Team

THANK YOU