Red Hat
People Analytics

2021 SIOP Machine Learning Competition:

RHDS (Red Hat Data Science)

Brian Costello (Senior Data Scientist, People Analytics)

bcostell@redhat.com

April 2021

Agenda

1. Introduction
2. Task
3. Data Preprocessing
4. Modeling
5. Optimization and Results
6. Conclusions

Introduction

Are **Diversity and Inclusion** (D&I) and **Machine-Learning** (ML) on a collision course?

Introduction

- As of 2018 (Grissom), despite comprising 48.9% of the total workforce, women only account for 20.2% of Fortune 500 board of directors (BODs)
  - People of color only account for 14.4% of BODs
- Within 25 years, people of color are projected to comprise a majority of the US population (Cross & Brasell, 2019)
- US workforce is projected to become even more diverse over the next decade (Bureau of Labor Statistics, 2015)
- As a result, many organizations have made large investments in **Diversity and Inclusion** (D&I), emphasizing recruiting of diverse candidates and creating more fairness in selection systems (Flory et al., 2018)

Introduction

- **Machine-Learning** (ML) can be defined broadly, but for this context I define ML as predictive algorithms trained on data with a known outcome to create predictions on new data with an unknown outcome
- ML has become more pervasive across all industries and society more broadly (Chard, 2020)
- ML techniques and algorithms have been shown to have the propensity for bias across a variety of applications (Johnson, 2020)
- While there has been some documented use of ML in Human Resource (HR), overall usage is still low and there are many opportunities to employ ML techniques in an HR environment (Garg et al., 2021)

Introduction

Are **Diversity and Inclusion** (D&I) and **Machine-Learning** (ML) on a collision course?

The question becomes:

How can we as practitioners responsibly employ ML techniques in HR generally, and specifically with regards to selection, to reap the benefits of ML while simultaneously avoiding bias and any "impending collisions"?

Task

Task

- Using applicant selection item data for **hired associates** where it is known if an associate ***stayed***, was a ***high performer*** (HP), and/or is a member of ***protected class*** (PC) create hiring recommendations for new applicants with scores on the same selection criteria
- Data is anonymized from Walmart hourly associates and applicants
- Final rankings based on ***Hiring Accuracy***
- ***Hiring Accuracy*** is comprised of two parts:
  - **Selection Accuracy**: proportion of hired applicants that stay and are high performers
  - **Adverse Impact Ratio**: points lost for selection unfairness (differing selection rates for protected and non-protected class applicants)
- General approach was to predict both outcomes independently and create weight-optimized final hiring "score" by simulating hires on training data

Hired Associates

ID	Item1	Item2	Item3	PC	HP	Stay
1	1	2	4	No	No	No
2	5	4	5	Yes	Yes	Yes
3	4	4	3	No	No	Yes

ID	Item1	Item2	Item3	Hire?
4	3	3	4	
5	2	2	1	

New Applicants

Data Preprocessing

Categorical Variables

- Converted Situational Judgement Test (SJT) and Biodata items to characters (i.e., treated them as categorical)
- Dummy coded categorical fields
  - One-hot-encoded fields (i.e., one column for each category)
  - Removed values occurring less than 5%
  - Removed highly/perfectly intercorrelated categories
  - Thus, referent group becomes the grouping of any categories removed by this process
  - Initially missing values treated as their own category, but no variables had more than 5% missing data

ID	F/M
1	M
2	M
3	F
4	F

ID	F	M
1	0	1
2	0	1
3	1	0
4	1	0

One-Hot Encoding

ID	F
1	0
2	0
3	1
4	1

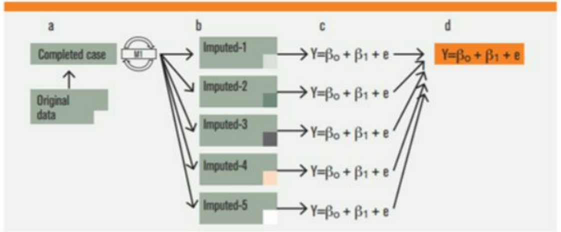
Remove Perfectly Correlated Categories

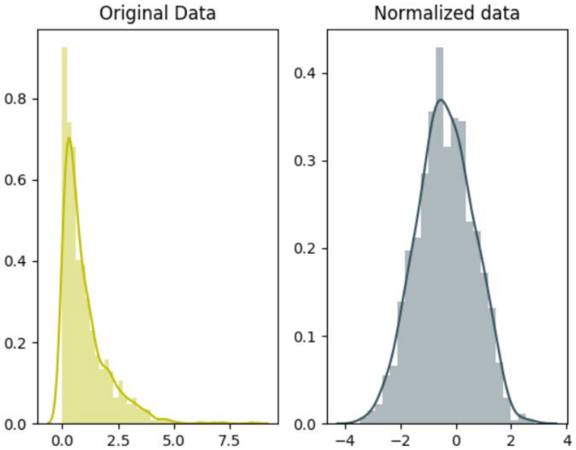
Data Preprocessing

Numeric/Interval/Ratio Variables

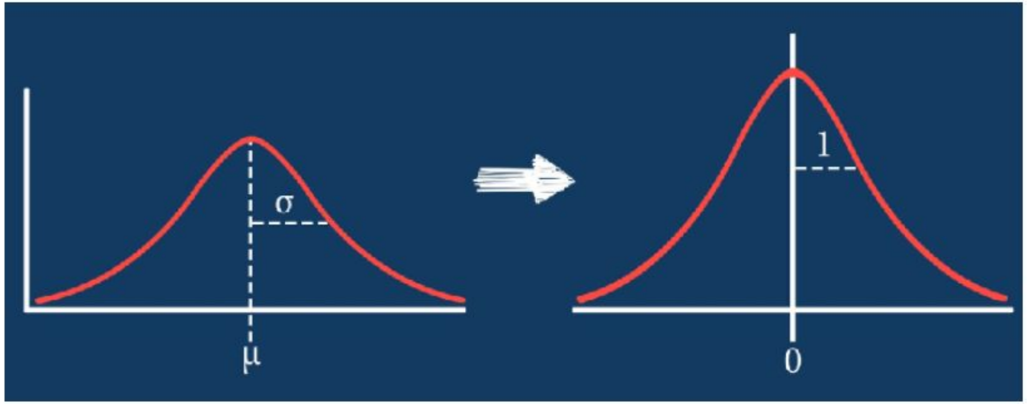
- Personality, Scenario, and Time items
- Centered and scaled
- Multivariate imputation for missing data (mice)
- Experimented with Principal Components Analysis (PCA)
  - Not used in winning solution
  - Minimal dimension reduction (i.e., items appear to have solid discriminant validity)

Data Imputation





Centering



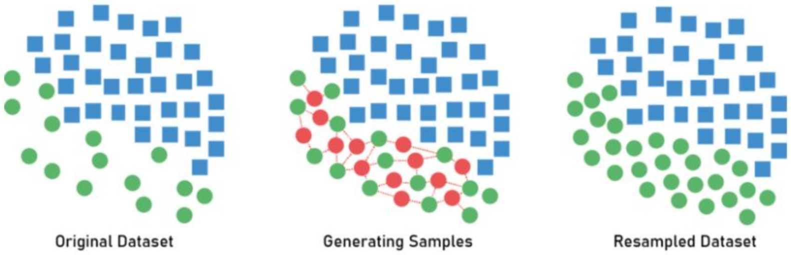
Scaling

Data Preprocessing

Outcome Variables

- Due to **high performer** imbalance (~40% high performers), experimented with SMOTE resampling
  - Not used in winning solution
  - Decreased test data sensitivity (true positive rate) and increased specificity (true negative rate) making prediction bias more extreme

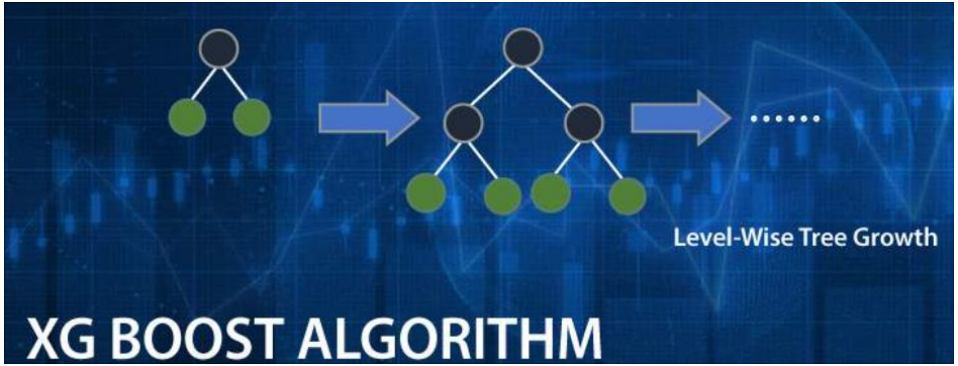
Synthetic Minority Oversampling Technique



SMOTE Algorithm

Modeling

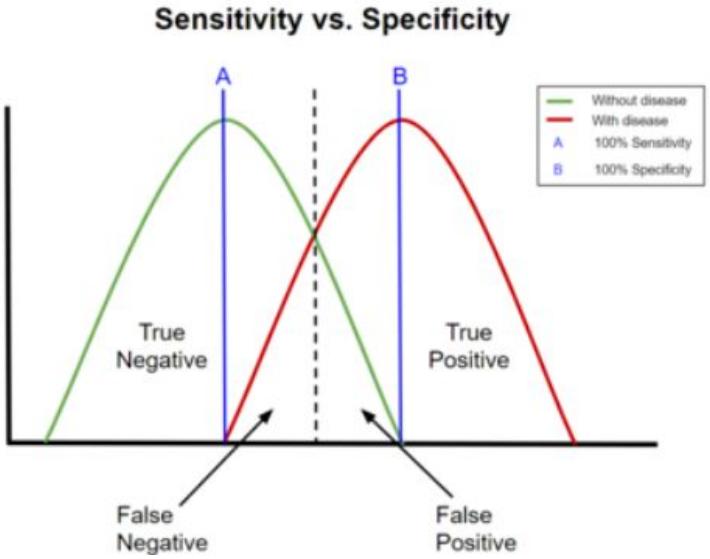
- Predicted each outcome independently
- Used all possible predictors
    - Initially tried without time fields, as they showed some potential protected class bias, but excluding them substantially reduced prediction quality
- XGBoost (surprise, surprise)
- Random grid search for cross-validation hyperparameter tuning
    - 50 iterations
    - Interestingly, I used more iterations (100 and 250) for submissions that did not perform as well
- Average of top 5 models based on cross-validation hyperparameter tuning
- 95/5 train-test split
- Shoutout to O.G. tictoc (function) here (used for calculating process time)



Weight Optimization and Results

Part 1: standard train-test validation to assess general accuracy of each model

- Used all data available for each outcome (i.e., “Retained” had more training data than “Top Performer”)
- Assessed AUC, sensitivity, and specificity on test (hold-out) data



Weight Optimization and Results

Part 2: Weight Optimization

- Each outcome produces a "score" (prediction) between 0 and 1
- Approach was to create a single, aggregated final "score" by combining the predicted "scores" for each metric
- Simulated hiring accuracy on training data by testing all weight combinations to create the final "score"
  - Possible combinations were between 0 and 1 in 0.01 increments for each predictor such that the weight total added up to 1
- Ranked by final score and "hired" top half
- Experimented with weight optimization on different subcomponents of the train data (i.e., train data used to train the model and train data used as test holdout)
  - Unique IDs appearing in holdout data for both outcomes
    - Due to using random outcome-based partition sampling Unique IDs can appear in data used to train the model for one outcome, and holdout data for the other; thus, true holdout data (i.e., unique IDs appearing in holdout data for both outcomes) resulted in a fairly low sample size (≈ 300)
  - Unique IDs appearing model training data for both outcomes
  - All unique IDs for all train data (i.e., model training and holdout combined) across both outcomes

Weight Optimization and Results

Example 1: Retained Weight = 1, Performance Weight = 0

ID	Retained Prediction	Performance Prediction	Final Score	Rank	Hired?
1	0.9	0.8	0.9*1 + 0.8*0 = **0.9**	1	Yes
2	0.6	0.85	0.6*1 + 0.85*0 = **0.6**	3	No
3	0.7	0.65	0.7*1 + 0.65*0 = **0.7**	2	Yes
4	0.3	0.4	0.3*1 + 0.4*0 = **0.3**	4	No

Weight Optimization and Results

Example 2: Retained Weight = 0.3, Performance Weight = 0.7

ID	Retained Prediction	Performance Prediction	Final Score	Rank	Hired?
1	0.9	0.8	0.9*0.3 + 0.8*0.7 = **0.83**	1	Yes
2	0.6	0.85	0.6*0.3 + 0.85*0.7 = **0.775**	2	Yes
3	0.7	0.65	0.7*0.3 + 0.65*0.7 = **0.665**	3	No
4	0.3	0.4	0.3*0.3 + 0.4*0.7 = **0.37**	4	No

Weight Optimization and Results

Final weight optimization:

- Used all training data (i.e., train + test holdout) to generate weight optimization
- Retained = 0.02
- Top Performer = 0.98
- Train Score = 70.50
- Dev Score = 59.94
- Test (Final) Score = 61.09

Conclusions

- Despite varied approaches, none of the winning solutions violated the 4/5ths rule!
- That said, the task was defined in a way that heavily penalized adverse impact
  - It is imperative that we as practitioners are always mindful of ensuring fairness in any models we build
  - We must strive to be better than minimum legal requirements
- In practice, there are a number of additional steps that should be taken (e.g., individual item analysis)
- In my opinion, ML techniques and their responsible deployment (with coding in R and/or Python) must become part of IO psych graduate programs, as we are already competing against other fields for these types of roles

Final Thought

Are **Diversity and Inclusion** (D&I) and **Machine-Learning** (ML) on a collision course?

Final Thought

Are **Diversity and Inclusion** (D&I) and **Machine-Learning** (ML) on a collision course?

Not necessarily, and we have a unique opportunity as experts and practitioners in HR to emerge as leaders that drive the fusion of D&I and ML into the future in a responsible and fair way!