MACHINE LEARNING & UNDERWRITING

By: Brian Moritz, Christopher Cao, Daoud Haq, and Luke Natalo

BUSINESS CONTEXT

- Life insurance is a common protective financial tool which manages trillions of dollars in benefits across the U.S.
- Payouts from a life insurance policy tend to be, relatively, much higher than the premiums compared to other forms of insurance.
- Due to the nature of life insurance policies, it is crucial for businesses that sell such insurance to mitigate as much risk as possible.
- One way that this is done, and for a very long time, is through the process of underwriting.

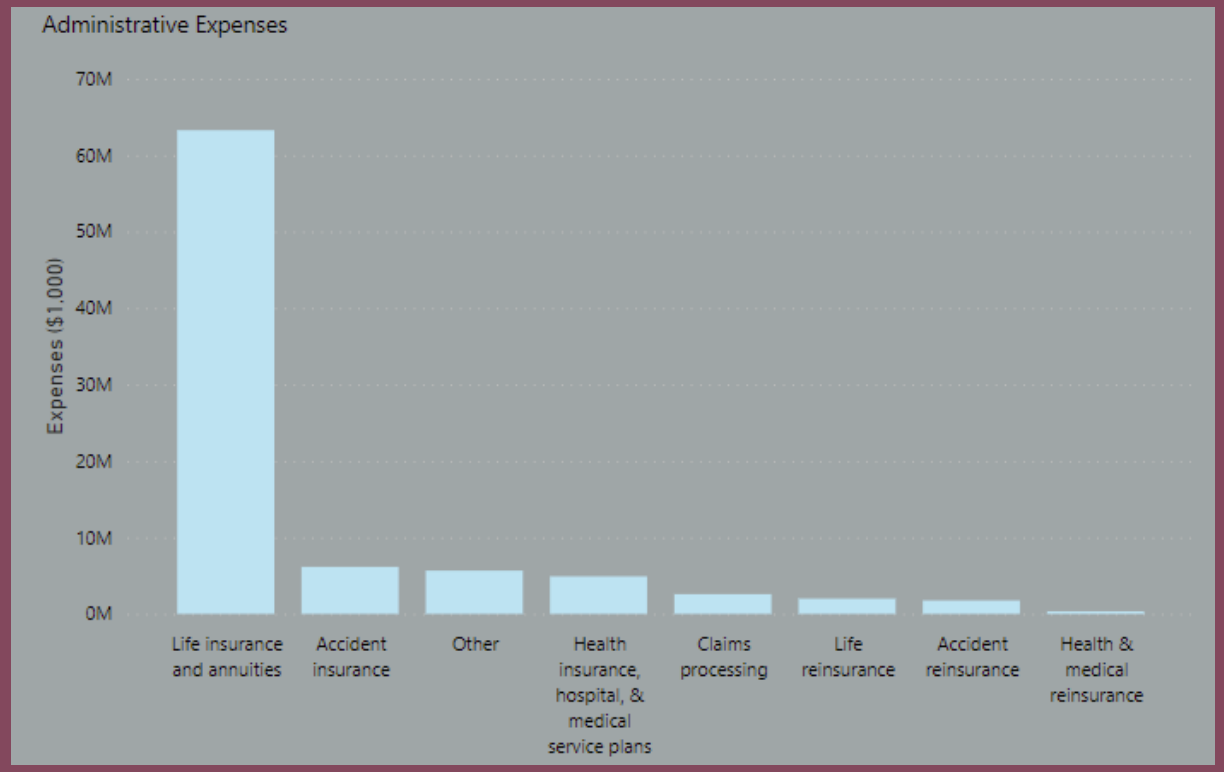
WHAT IS UNDERWRITING?

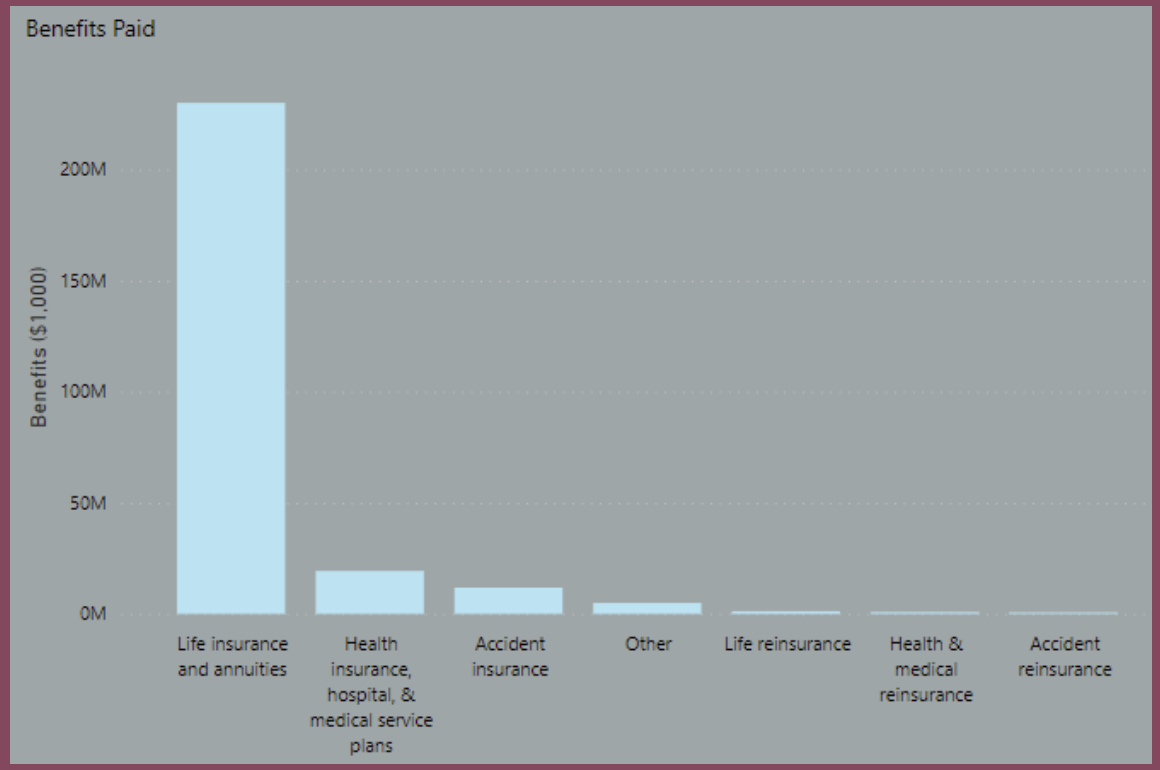
- Underwriting, within insurance, is an old practice which assess the risk of an uninsured party and allows them to pay a premium to relinquish that risk to the insurer.
- Traditionally, this is performed manually using human judgment and point-based systems that consider risk factors independently.
- These methods are sufficient in industry but are coarse and subject to inconsistency.
- As a result, traditional underwriting limits the degree to which an insurer can estimate risk from data and offer efficiently priced products in a timely manner.

INDUSTRY COST ANALYSIS

- When looking at industry wide data on expenditures as well as benefits paid out, we can see that the lion share of these costs are tied to life insurance and annuities.
- With this in mind, we wanted to see how and if ML models could help cut costs on the underwriting process and pass those profits back to either shareholders or policyholders.

Census Data from 2017

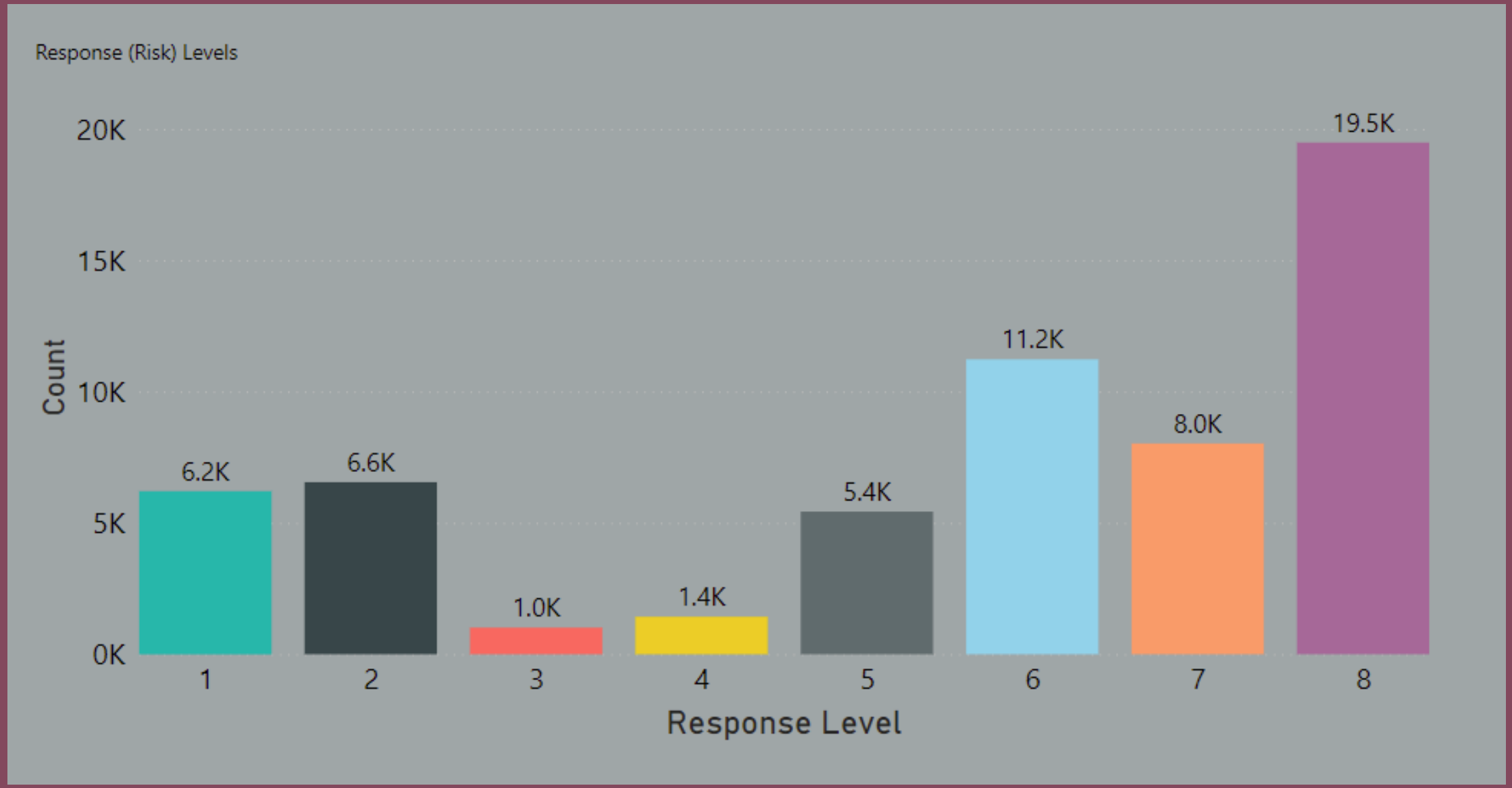




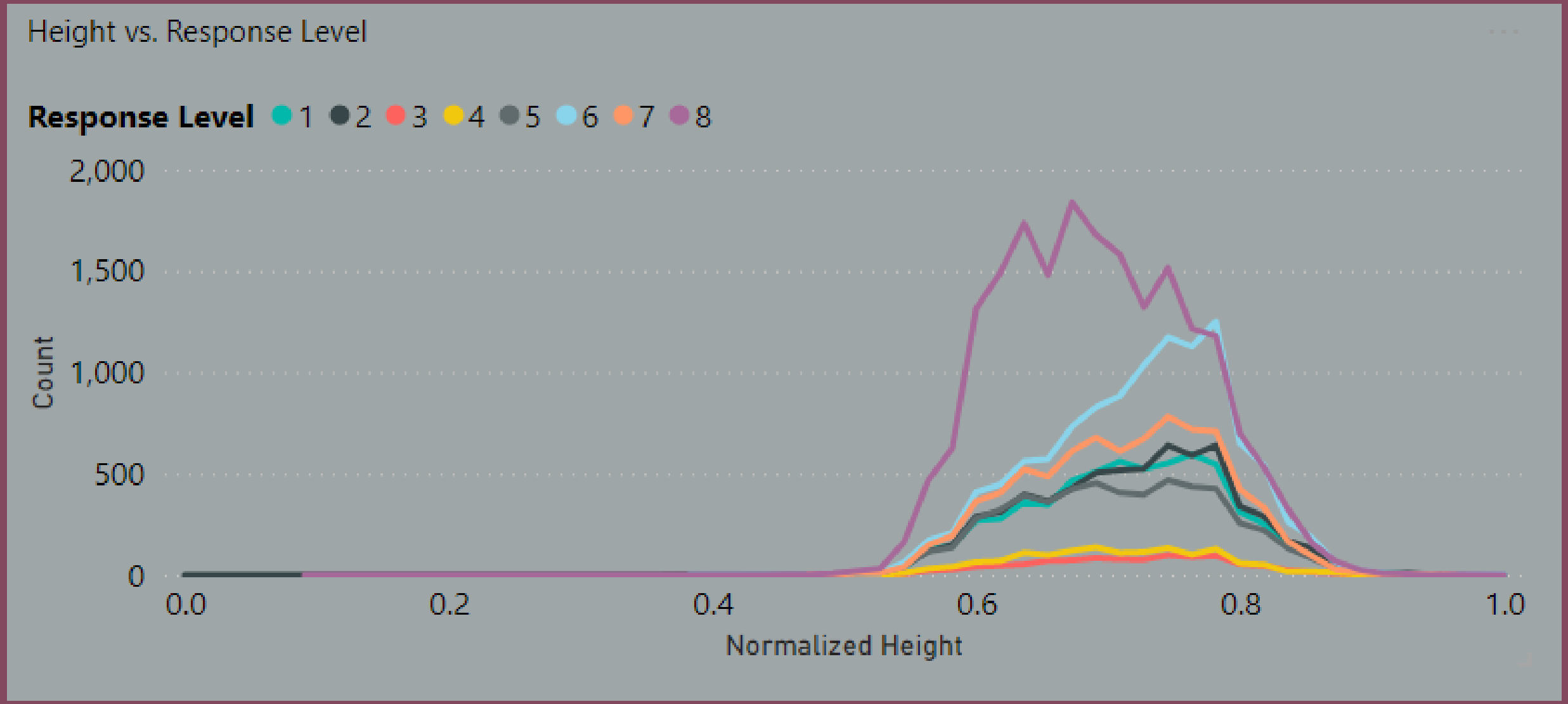
PRUDENTIAL LIFE INSURANCE DATA

- The data we chose to work with was all pre-normalized as well as changed into dummy variables to prevent giving away medical background on the applicants.

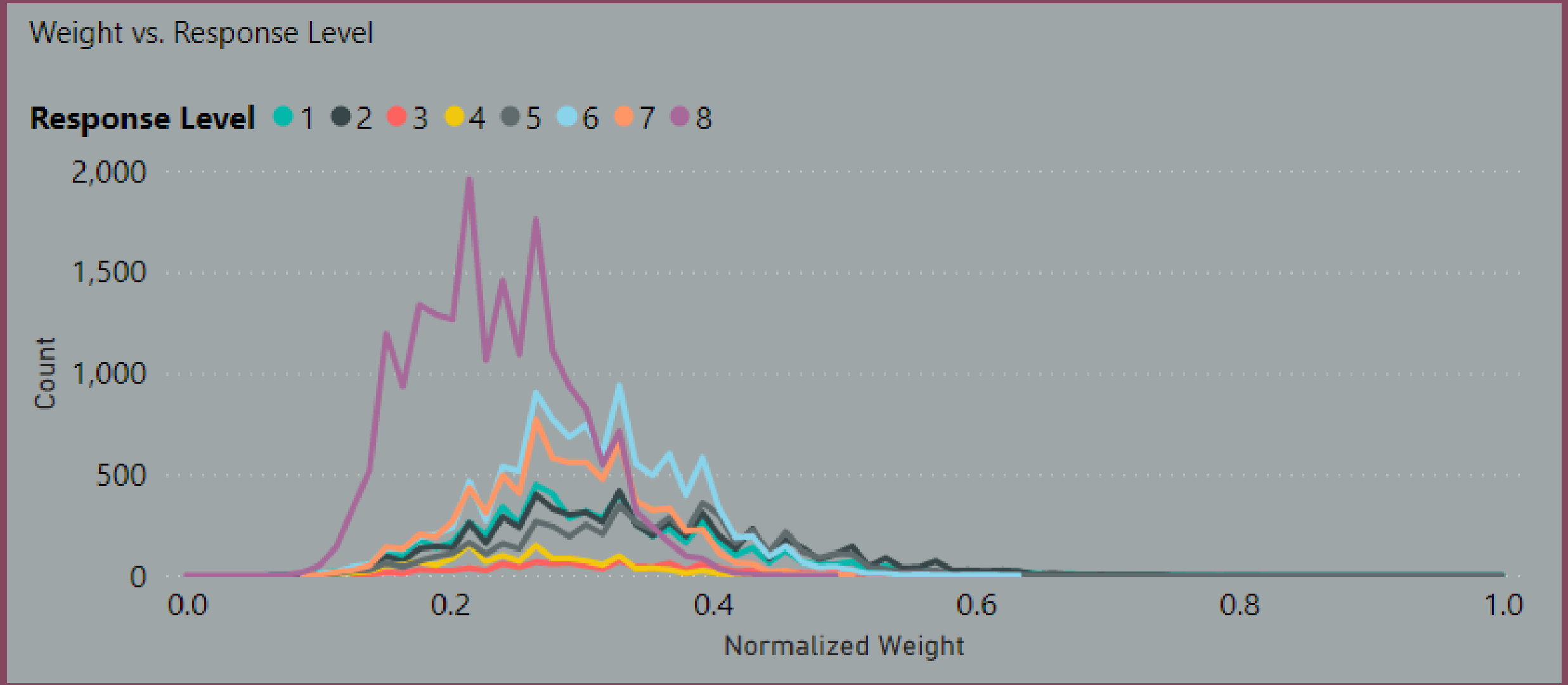
Variable	Description
Id	A unique identifier associated with an application.
Product_Info_1-7	A set of normalized variables relating to the product applied for
Ins_Age	Normalized age of applicant
Ht	Normalized height of applicant
Wt	Normalized weight of applicant
BMI	Normalized BMI of applicant
Employment_Info_1-6	A set of normalized variables relating to the employment history of the applicant.
InsuredInfo_1-6	A set of normalized variables providing information about the applicant.
Insurance_History_1-9	A set of normalized variables relating to the insurance history of the applicant.
Family_Hist_1-5	A set of normalized variables relating to the family history of the applicant.
Medical_History_1-41	A set of normalized variables relating to the medical history of the applicant.
Medical_Keyword_1-48	A set of dummy variables relating to the presence of/absence of a medical keyword being associated with the application.
Response	This is the target variable, an ordinal variable relating to the final decision associated with an application


Response (Risk) Levels
20K
15K
10K
5K
0K
Count
6.2K
6.6K
1.0K
1.4K
5.4K
11.2K
8.0K
19.5K
1
2
3
4
5
6
7
8
Response Level

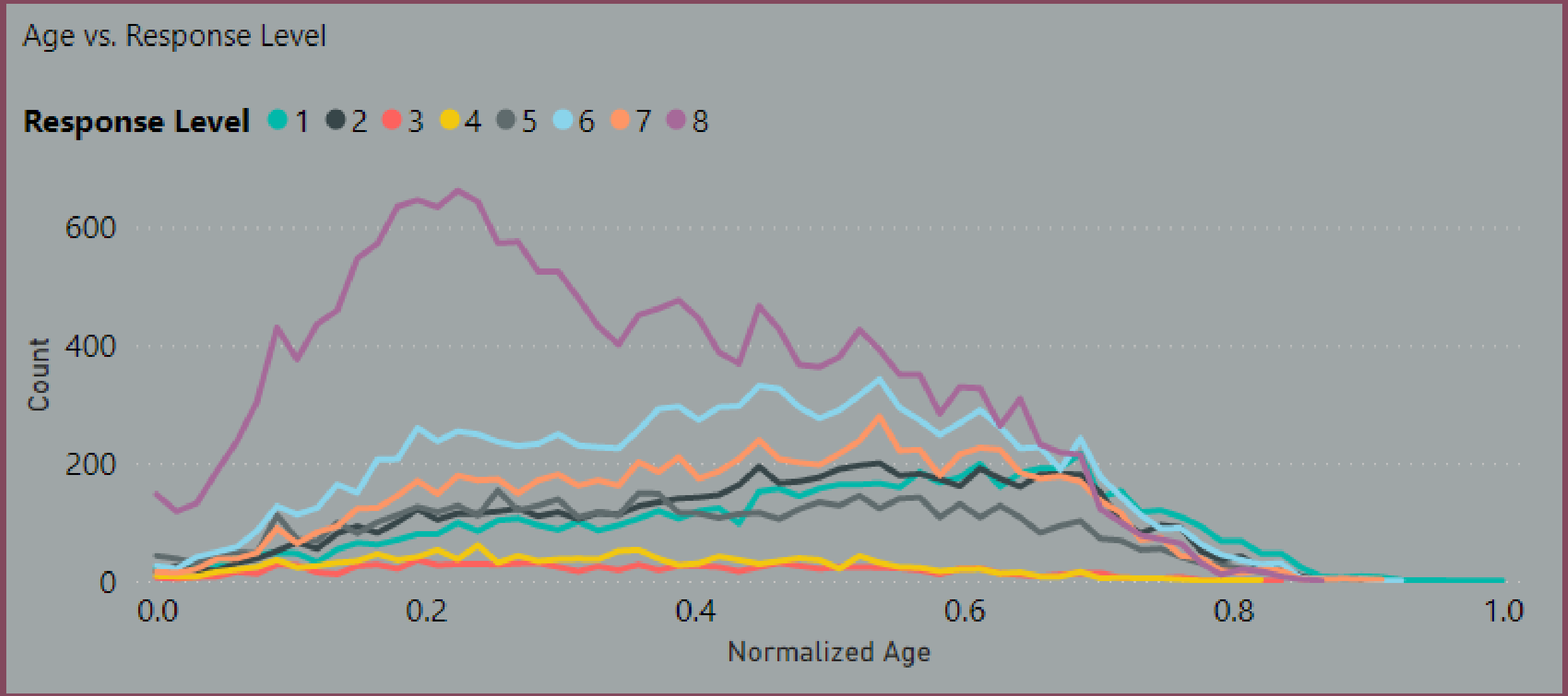
Height Findings



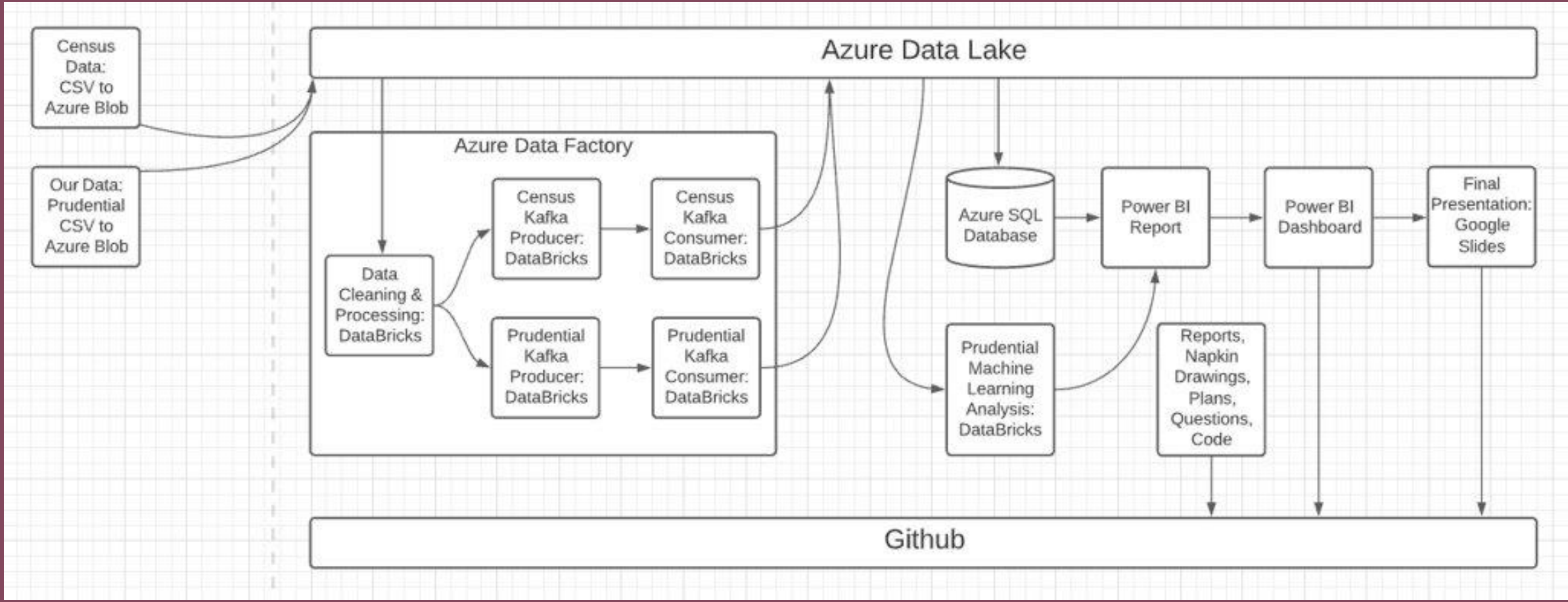
Weight Findings



Age Findings

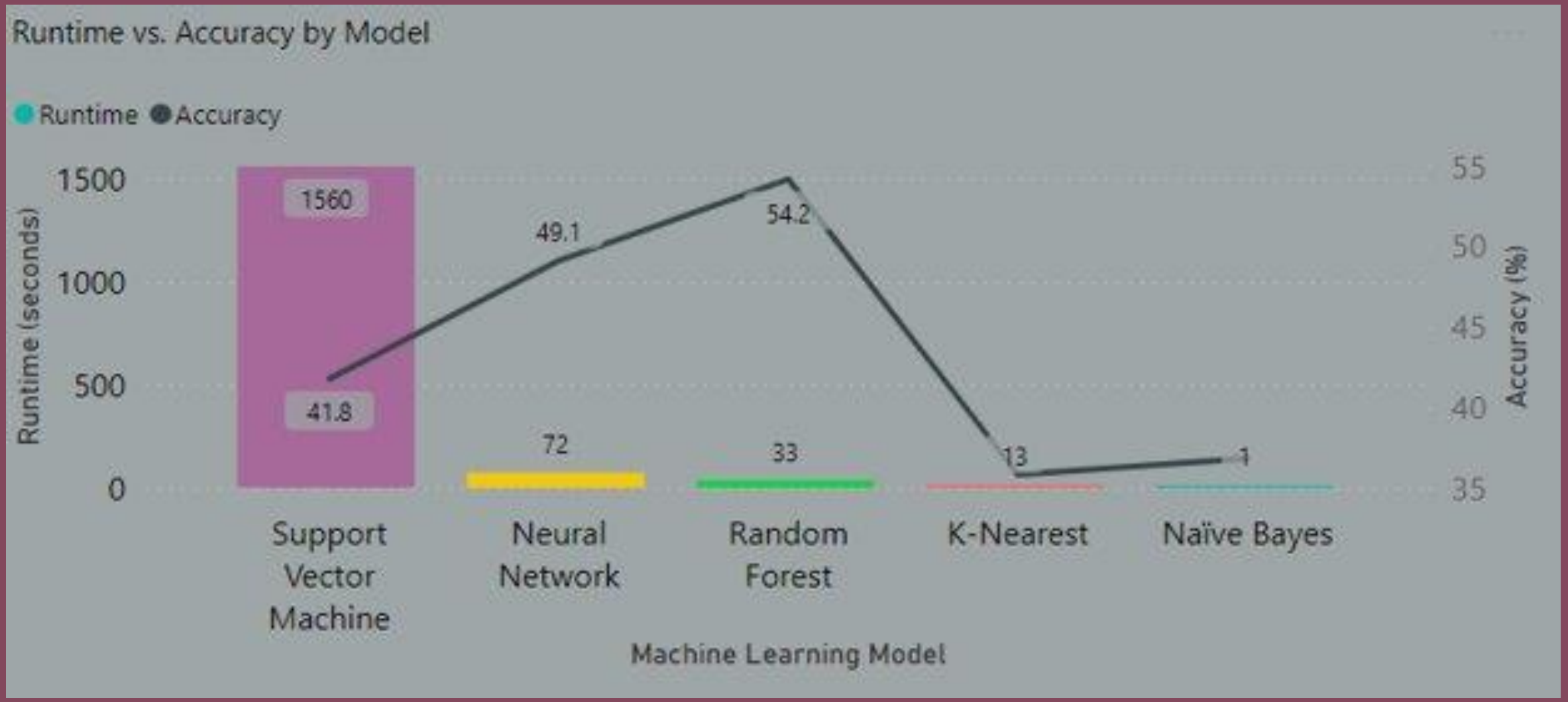


Data Platform Overview



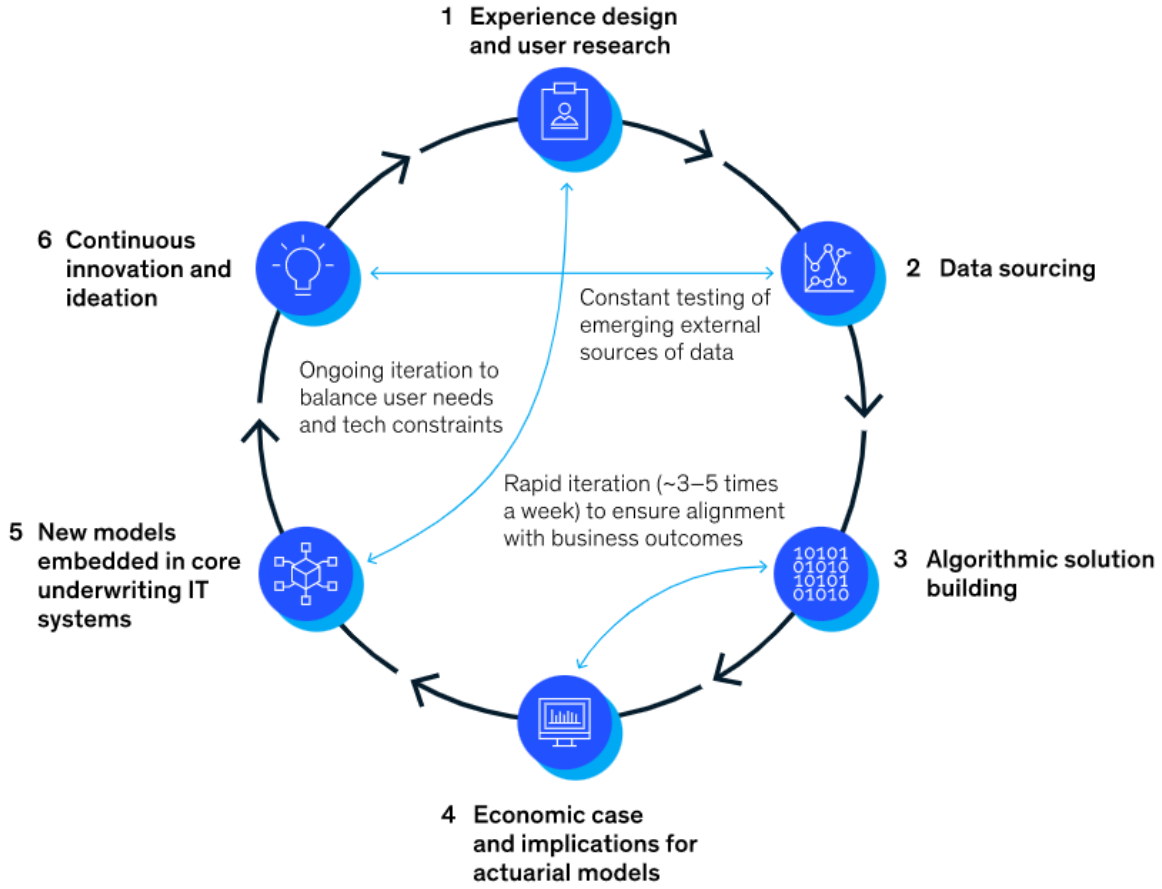
OUR MACHINE LEARNING DECISION PROCESS

- Classification vs. Regression
  - Classification models were chosen as the problem was a classification one to begin with (i.e., classifying data into risk category).
- Model Selection:
  - Support Vector Machine (SVM): Hyperplanes create decision boundaries
  - Neural Networks: Mimics human brain neurons
  - Random Forest Classifier: Ensemble of decision trees. Known for high accuracy with large datasets
  - Naïve-bayes: Probability-based
  - K-Nearest Neighbors (KNN) Classifier: Is a commonly used classification model essentially fitting data by its distance to similar data points. It is simple and does not require too much time to run/implement.


Runtime vs. Accuracy by Model
Runtime
Accuracy
1500
1000
500
0
Runtime (seconds)
55
50
45
40
35
Accuracy (%)
1560
41.8
49.1
54.2
72
33
13
1
Support Vector Machine
Neural Network
Random Forest
K-Nearest
Naïve Bayes
Machine Learning Model

CONCLUSIONS

- Based off our results, the model with the best accuracy score and most reasonable runtime is the Random Forest Classification model.
- However, the accuracy score is at around 54%, hardly accurate enough to classify applicants in a risk category reliably.
- More work should be done in improving the most promising models (i.e., Random Forest and Neural Networks) and some avenues for this would include:
  - Additional feature engineering
  - Additional hyper-tuning of model parameters
  - Making use of more data



Source: McKinsey analysis

RECOMMENDATIONS

- Based off the results of the models shown previously it is clear that while machine learning is a powerful tool it's not foolproof.

- Therefore, going forward, we encourage the use of an underwriting innovation program similar to the ones presented by McKinsey and Co (figure on the left).