Predicting Future Warranty Costs

PURDUE UNIVERSITY | The Data Mine

WABASH

Thomas Ashcraft, Joe Dressel, Nandini Pande, Nitya Ramireddy, Nam Tran, Vinayak Vipradas, Ike Wongkaew, Adam Zuckerman

INTRODUCTION

About Wabash:
Wabash is an American industrial manufacturing company of engineered solutions and services for transportation, logistics and distribution industries, as well as North America's largest producer of semi-trailers and liquid transportation systems.

Our goal:
- Build a model to predict future warranty costs for Wabash
- Utilize historical warranty, configuration data, and product specifications to gain meaningful insights to forecast warranty costs
- Investigate the fail codes and the factors that attribute to specific failures to understand warranty trends and root causes
- Analyze the different predictors that will influence warranty costs

Why:
- More efficient planning for future warranty costs
- Improve product quality
- Strengthen value proposition for new products

REDUCING FEATURES

- We found that numerous features had an insignificant number of appearances in claims in the data
- We also found that some features were always present alongside one another so we merged these features to avoid collinearity in the predictive models.

MODEL BUILDING

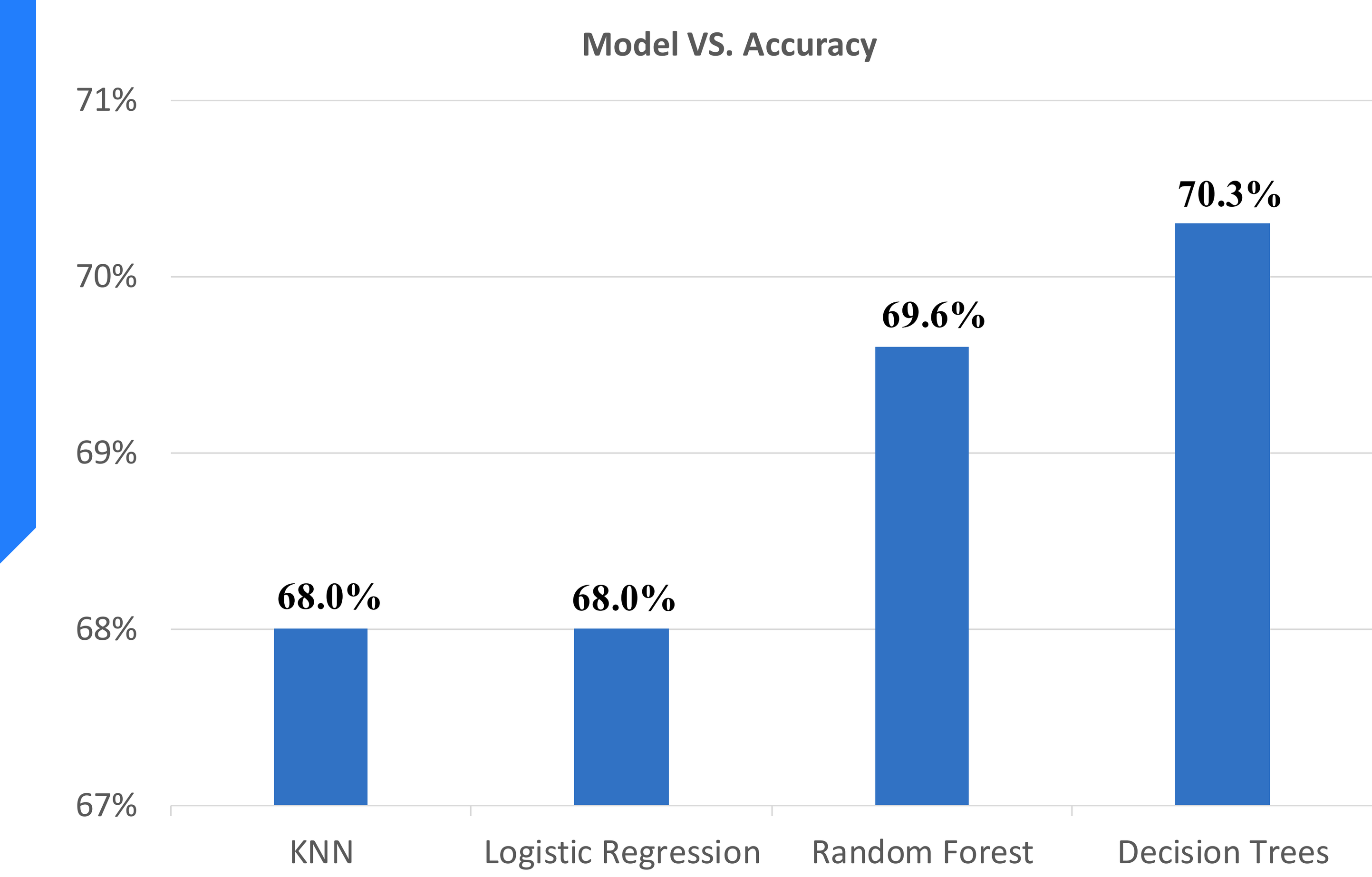
Our models:
- Split the data into training and testing sets and sees how well the test set performs
- Provides an accuracy value based on how well it fits the test set

K-Nearest Neighbors
- Used different values of k in order to optimize accuracy
- Highest accuracy obtained was 68% using k=20

Logistical Regression
- Used configurations that were correlated with fail codes to predict likelihood of a specific fail code occurring
- Coefficients were determined by strength of correlation; likelihood calculated using binary configuration data

Decision Tree
- Test that closely mirrors human decision making and can handle the binary configuration data easily without creating dummy variables

Random Forest
- Same as a decision tree except for a small tweak that decorrelates the trees
- Results did not significantly increase accuracy compared to the decision tree



CHALLENGES/ROADBLOCKS

Model Confusion
- We initially tested a variety of regression models using qualitative predictors (fail code, trailer count, etc.)
- Very low R^2 values (see figure below) indicated poor linear fit, leading to confusion on which models to implement
- Eventually obtained configuration data as new predictors

Data Flaws
- Inconsistent attribute data led to issues with the training model and caused potentially key information to be left out of analysis
- Our dataset contained columns of free text meaning to be identical, which led to misspellings/abbreviations that made the data inaccurate

Predictors for Claim Amount	R^2
Days in Service	0.00
Trailer Count	0.313
Fail Code	0.053
Days in Service, Trailer Count	0.313
Days in Service, Fail Code	0.053
Trailer Count, Fail Code	0.365
Days in Service, Trailer Count, Fail Code	0.365

CONCLUSION/FUTURE PLANS

What's next:
- The models we built can be continuously updated by Wabash to predict their future failures and costs associated with them
- As we approach the end of the year, we plan to improve accuracy of our models by updating code, reducing more configurations, and implementing new tests such as chi-square tests and neural networks
- We also would like to summarize our findings in a Power BI dashboard

EXPLORATORY DATA ANALYSIS (EDA)

Fail Rates - We first calculated how often warranty claims occurred. We also found the fail rates for products, although our further analysis only included vans

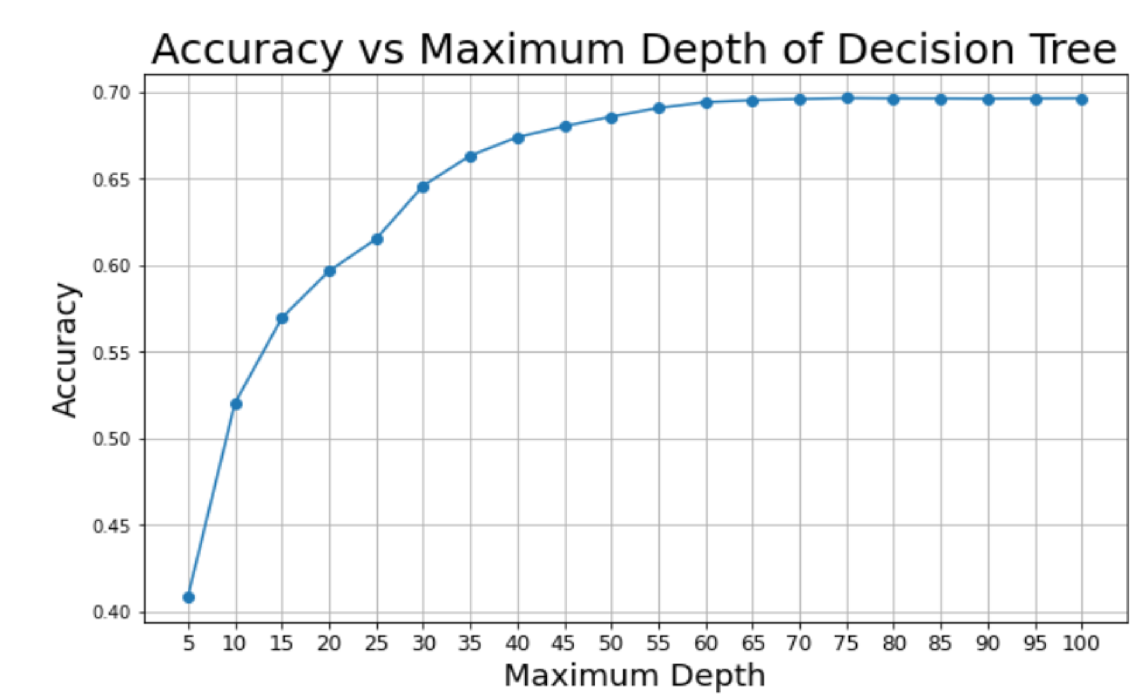
Outlier Analysis - We found claim amounts that were outside 3 standard deviations from the mean and removed them from our cleaned dataset.

Configuration Analysis - We sorted the configuration data by listing the trailers for each individual configuration. We then calculated the percentages of trailers that resulted in each fail code.

Percentage Analysis - We looked at which fail code occurred most often for each configuration and excluded outliers. These filtered percentages eventually helped us with our predictive modeling.

Configuration Frequency Function - We built a function that first looked up the averages of each configuration to see which occurred most often, and then chose upper and lower bounds to set a threshold for averages to be excluded.

Configuration Probability Function - We built a function to calculate the probability of a particular configuration showing up, given another configuration exists within a certain fail code. This was our first predictive model.



ACKNOWLEDGEMENTS

Aaron – IT Director
John – Data Analytics Manager
Evan – Data Analyst
Mike – Senior Manager of Warranty
Mark – Warranty Supervisor
Lauren – Data Mine Corporate Partners Advisor
Kali – Data Mine Associate Research Engineer
Poomin & Darsh – Wabash Project TA

DISCLAIMER: ALL NUMBERS ARE THEORETICAL

The Data Mine Corporate Partners Symposium 2024