Kayla Johnson
Amanda Konet
Meena Muthusubramanian
Amanda Short

Bridgestone

Team 7

Executive Summary

Objectives	Approach	Key Findings	Next Steps
Determine features relevant to whether a customer will purchase tires Build a model to predict tire purchases in November 2018. Those likely to purchase should be targeted to receive Bridgestone's November 2018 email promotion	Exploratory data analysis and feature engineering on data provided Base model to capture baseline performance More complex models with external features added to improve performance and value added	Most important features for predicting tire purchases are: 1) Days since last service 2) Total sales ($) on services 3) Total tire sales ($) $1.3M estimated model value by targeting "on the fence" customers	Increase model complexity to better capture data nuances Add additional weather and road condition features as predictors of tire sales

Data Overview and Assumptions

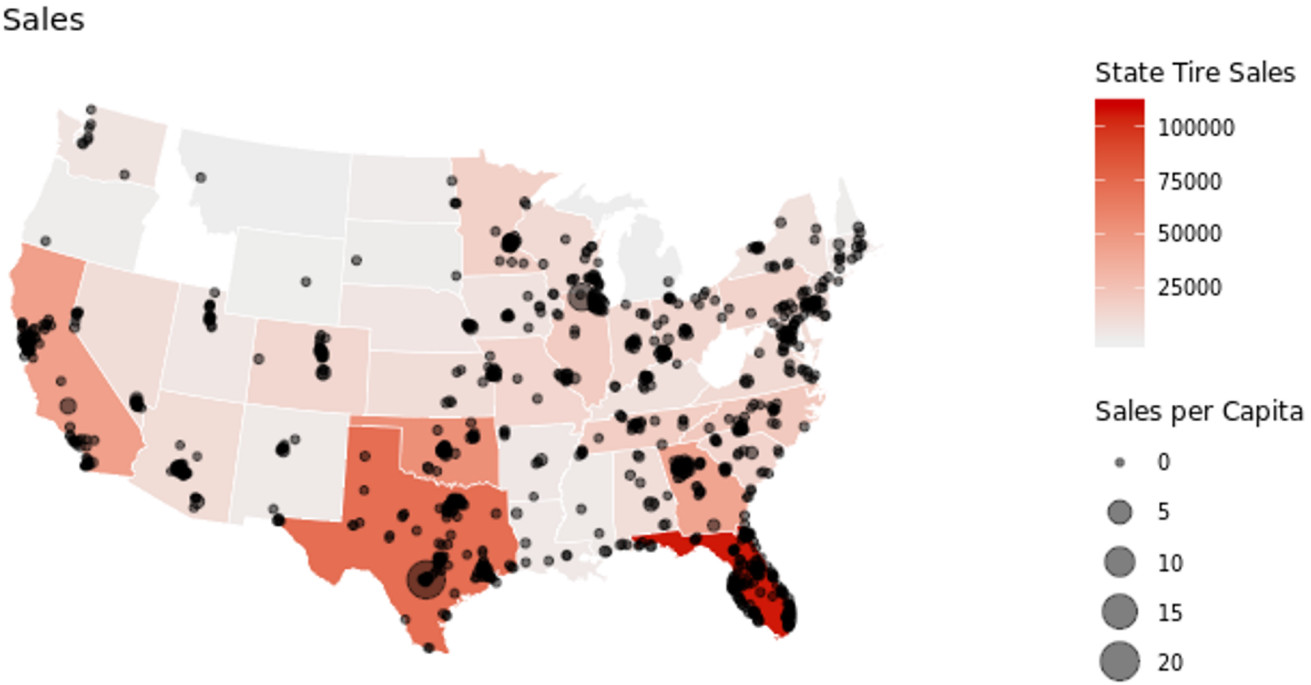
- Process
  - Aggregated data by IDs of eligible customers
  - Joined aggregated tables together to form the base dataset
- EDA was carried out on 1) the complete customer journey for 10% of the eligible customers and 2) the modeling dataset
- **Assumption**: Factors that help predict November 2017 tire sales will also help to predict November 2018 tire sales

1. Data Exploration

Data Exploration

Florida, **Texas**, and **Oklahoma** have the most tire sales.

Universal City (TX), **Rockford** (IL), and **Edwards** (CA) have the highest tire sales per capita.

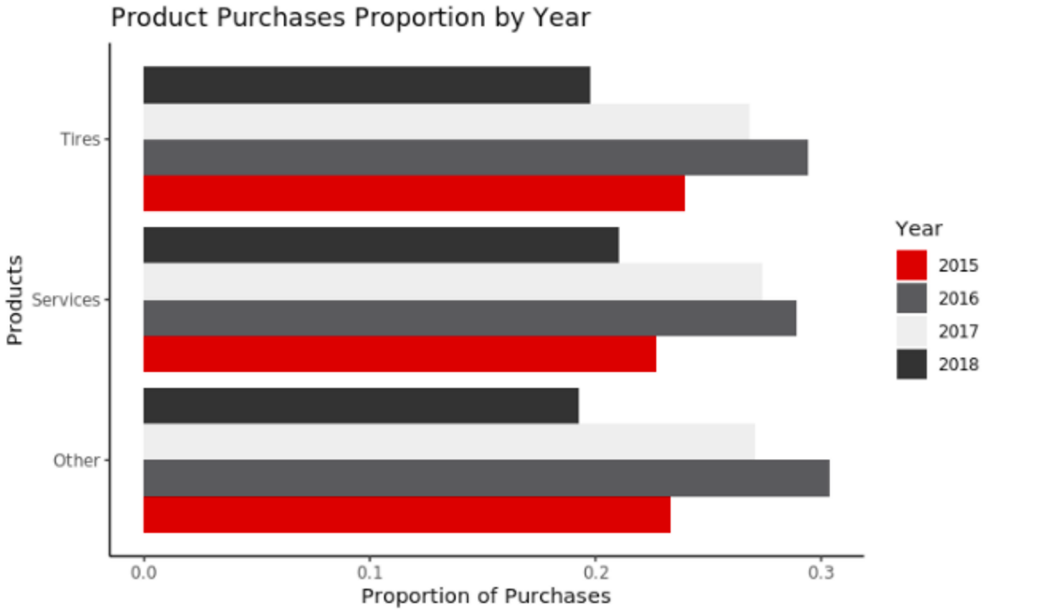


Data Exploration

2016 saw the most sales across all three product categories

There is an even split among people that come for tires, services, or for other reasons

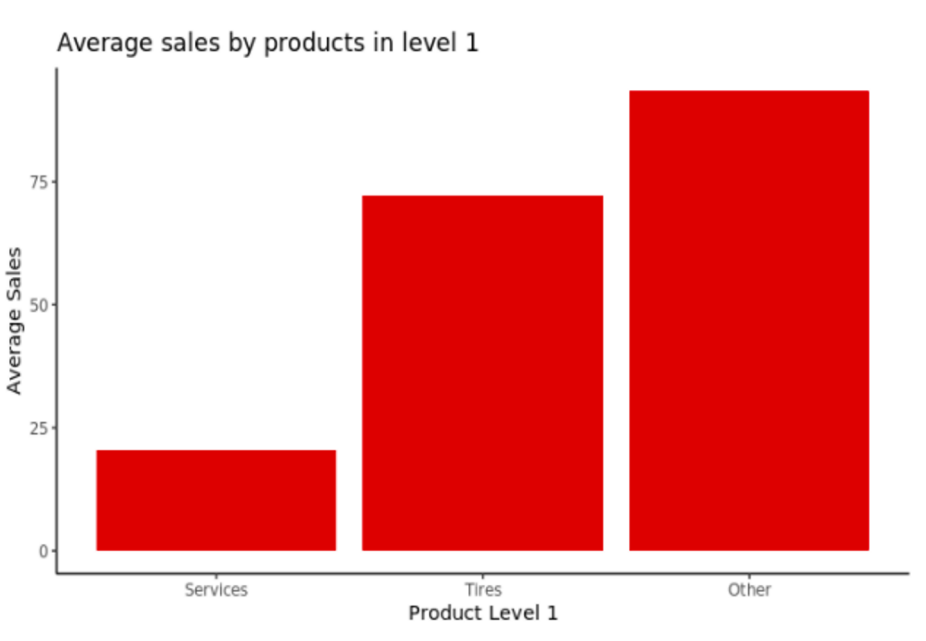
*2018 reflects Jan-Oct purchases



Data Exploration

Customers spend the most, on average, on “other” products at Bridgestone, followed by tires

Customers spend the least on vehicle services



Data Exploration

Customers own 1-2 vehicles that are 9 ± 5 years old, on average

November 2017 purchase	No November 2017 purchase
180 days since last service	324 days since last service
458 days since last tire purchase	667 days since last tire purchase

2. Modeling

Feature Engineering - Existing Data

Time (in days) since tires or services were purchased

Number of transactions by product category and year

Min, max, average vehicle age for every customer

Total revenue from tires, services, other

Feature Engineering - External Data

Number of commuting vehicles by zip code

Zip code population

Average travel time by zip code

Weather data by state: average snowfall from November to January, average & max summer temperatures

Baseline Model

Predict which customers will purchase tires in Nov 2017

- Simple logistic regression model
- Features include the original data provided with minimal data processing, no external data
- Baseline accuracy of 62%

Sale Prediction

Predicting tire sales November 2017

- Random forest model
- 200,000+ customer ID's stratified on tire purchase
- Original features + feature engineering
- Model predicts correctly 67% of the time
- 5% improvement over the baseline
- 9% accuracy on 2018 purchases

Top features to target

Days since last service
Total sales on services
Total tire sales
Days since last tire purchase
Number of commuting vehicles in zip code
Population in zip code
Average travel time in zip code

Estimated Value

- Look at high probability customers (>=65%)
- 75% of those customers did actually purchase tires in our test set
- 25% of customers did not purchase although we assigned them a high probability of doing so based on their features

Average revenue from tires:
$72

Expected value of model:
$1.3M

Recommendations

Use top features to refine targeting strategy:

1. Days since last service
2. Total sales on services
3. Total tire sales

Target customers that have 65% certainty of purchase and above

Next Steps

Test more complex models

- Model interpretability was prioritized, but future models could prioritize performance instead

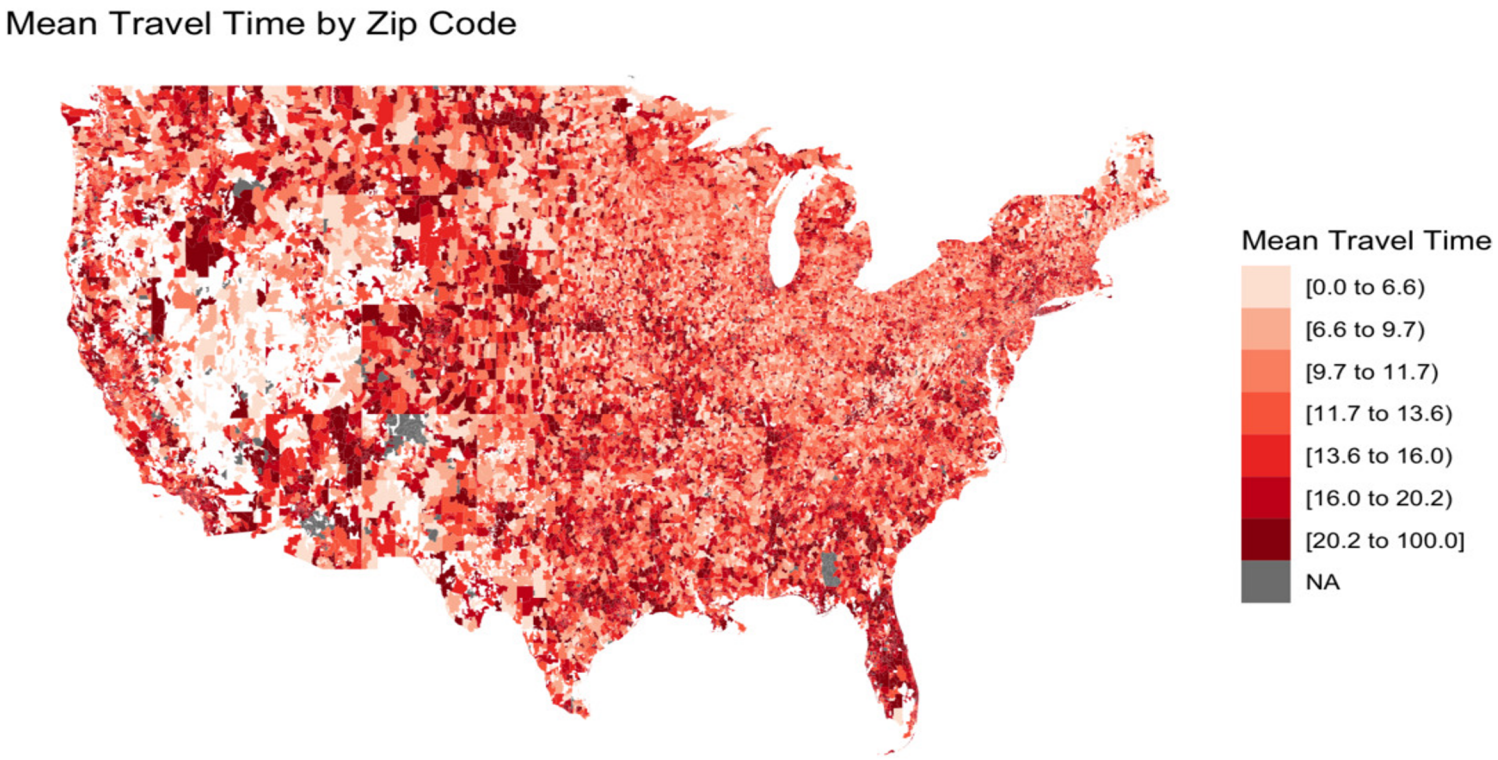
Add additional features

- Weather data for multiple years prior, other precipitation, and road conditions where possible

THANK YOU!

3. Appendix

Data Exploration



Dataset Creation

- Used vaex, a python package built to handle large datasets
- Handled one year of sales data at a time
- Joined sales and product table and grouped by customer ID to calculate:
  - Number of transactions where tires, services, and “other” products were purchased
  - Date of most recent tires, services transaction, then calculated days since
  - Total tire sales
- Joined sales and store tables to find most recent store visited
- Joined sales and vehicle to calculate number of vehicles owned and min, max, and average vehicle age
- Finally, joined all individually aggregated tables by customer ID to get base dataset

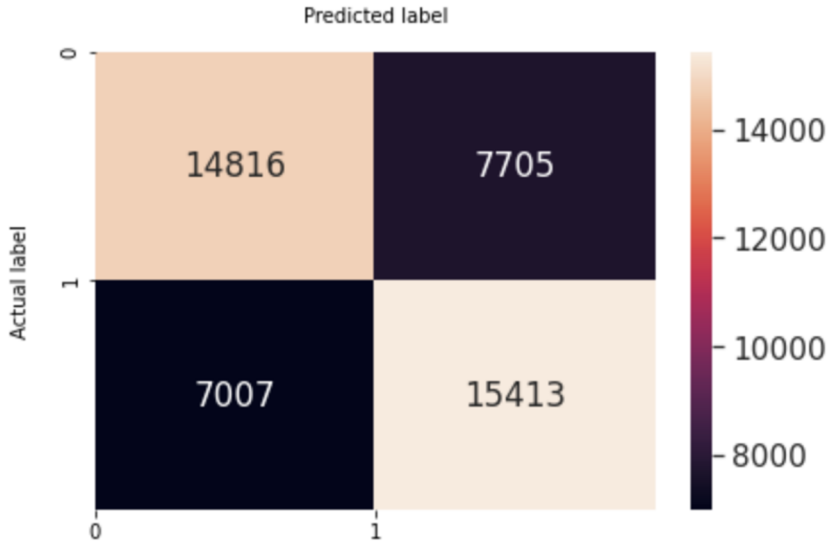
Baseline Model

- Logistic regression
- Accuracy: 62.3%
- F1 Score: 59.2%

Variable	Increase in odds of purchasing tires
num_vehicles	2.26494
max_vehicle_age	1.10299
region_S	1.03251
region_M	1.02078
region_N	0.97663
region_W	0.96054
avg_vehicle_age	0.92729
min_vehicle_age	0.90282

Random Forest Model

- Accuracy: 67%
- F1 score*: 70%
- Precision: 62%
- Recall: 83%



*threshold changed to 40%

Random Forest Model

Parameters:

'bootstrap': True,
'max_depth': 50,
'max_features': 'auto',
'min_samples_leaf': 4,
'min_samples_split': 2,
'n_estimators': 130

179,000 train; 45,000 test

features	coef
days_since_service	0.111736
sales_tire	0.094923
days_since_tire	0.091314
sales_service	0.084220
population	0.062471
commuting_vehicles	0.062286
travel_time	0.060060
num_service_year3	0.057462
avg_vehicle_age	0.045977
num_vehicles	0.045116

Random Forest Model

Model variables:

num_tire_year1
num_service_year1
num_other_year1
num_tire_year2
num_service_year2
num_other_year2
num_tire_year3
num_service_year3
num_other_year3

num_vehicles
min_vehicle_age
max_vehicle_age
avg_vehicle_age
days_since_tire
days_since_service
sales_tire
sales_service
sales_other

commuting_vehicles
travel_time
population
snowfall_nov
snowfall_dec
snowfall_jan
nov_tire_transactions
mean_temp
max_temp

Additional Models

Gradient boosting machines

- Implemented via xgboost and lightgbm in python
- Used grid search and cross validation to tune parameters
- Average accuracy of 66% and F1 score of 67%
- About as accurate as random forest but less interpretable

Expected Value

Filter final results of test set to probability >= .65

Filter only customers that did not purchase

Multiply each of these probabilities by the avg cost of tires ($72) and sum

Get average revenue per customer and scale for larger customer set

Prop of customers that purchased: .75