Individual Healthcare Cost Prediction

DurianCandy
Sunny Kwong
Shirley Li
Charles Siu
Sean Tey
Andrew Young

Analytic Goal

Building a model for predicting individual healthcare costs ~~and to identify the most important factors that determine a person's insurance rate.~~

Dataset

Health Insurance Marketplace

Source: Kaggle (Centers for Medicare & Medicaid Services)

Shape:

- **Rate:** 24 Cols x 12.7M Obs
- **PlanAttr:** 176 Cols x 77.4K Obs
- **ServiceArea:** 18 Cols x 42.2K Obs
- **BenefitsCostSharing:** 32 Cols x 5M Obs
  - Each observation represents an insurance plan
  - Each column represents an attribute of the insurance plan

Zillow Rental Values

Source: Zillow Research

Shape: 112 Cols x 934 Obs

- Each observation represents an a region
- Each column represents monthly *Zillow Rental Index* (A smoothed measure of the median estimated market rate rent across a given region and housing type.)

US Census Demographic Data

Source: Kaggle (US Census Bureau)

Shape:

2015 County: 37 Cols x 3220 Obs

- Each observation represents an a county in US
- Each column represents a measurement in census

Related Works

- ***Risk prediction in life insurance industry using supervised learning algorithms***
  We use Python instead of R and we focus on price prediction instead of risk prediction.
- ***Customer Clustering in the Health Insurance Industry by Means of Unsupervised Machine Learning***
  We use supervised learning instead of unsupervised and we focus on individual prices instead of the whole health insurance market.
- ***Supervised Learning Methods for Predicting Healthcare Costs: Systematic Literature Review and Empirical Evaluation***
  We see the problem as a regression problem instead of a classification problem. We also focus on finding the best features instead of the best algorithm.
- ***Predicting your casualties – how machine learning is revolutionizing insurance pricing at AXA***
  Our data source is static instead of dynamically streaming. We also use traditional ml models instead of deep learning so that we could interpret the feature importances.
- ***Exploring the use of machine learning for risk adjustment: A comparison of standard and penalized linear regression models in predicting health care costs in older adults***
  The paper tries to predict costs for older adults from an insurance company's P.O.V whereas our project tries to predict insurance price from a consumer P.O.V.

Preprocessing

Join other tables to Rate

Join

1. PlanAttr
2. ServiceArea
3. BenefitCostSharing
4. Rent

By

1. PlanId
2. State

Aggregate and Join Census

Aggregate columns

1. Sum
2. Average
3. Median

Join it with the previous tables

Drop unimportant features

Drop some unimportant features

1. ImportDate
2. IDs

With domain knowledge

Preprocessing

String to float

Transform numerical features from string type to numerical type

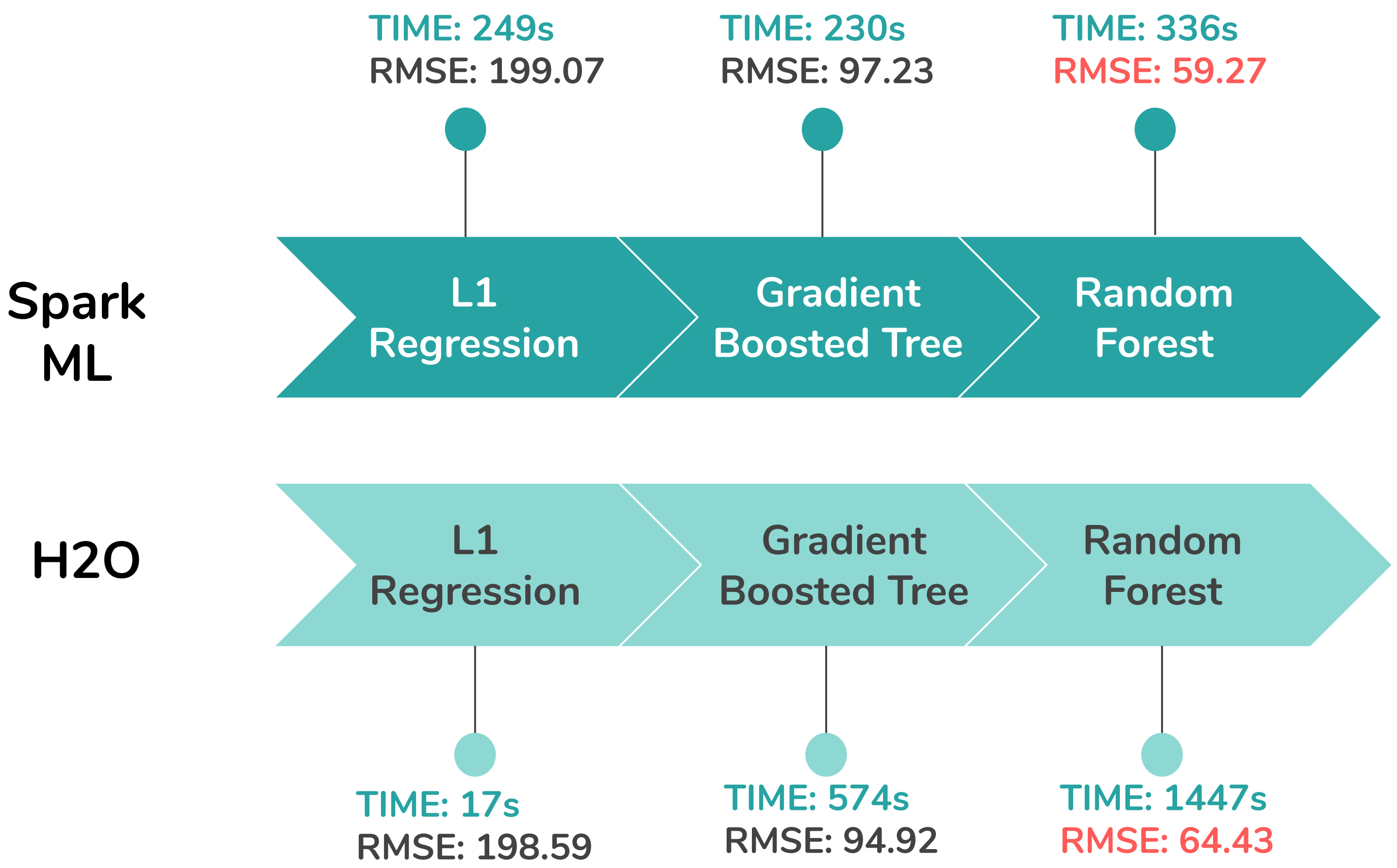
String to Index

Transform the categorical features from string to index

Imputation

Use the mean of the columns to impute missing values

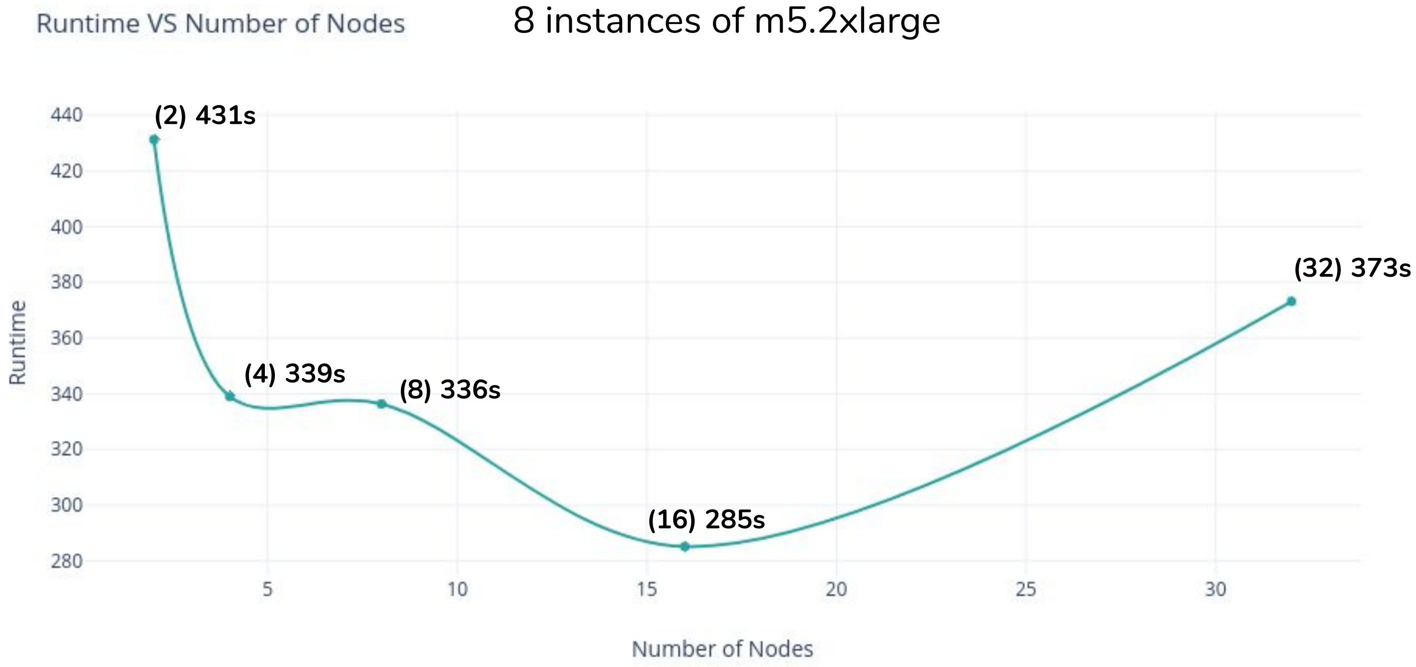
Machine Learning Outcome Comparison on the same cluster specs (M5.2xlarge x 8 instances)



Spark ML
TIME: 249s
RMSE: 199.07
TIME: 230s
RMSE: 97.23
TIME: 336s
RMSE: 59.27
L1 Regression
Gradient Boosted Tree
Random Forest
H2O
L1 Regression
Gradient Boosted Tree
Random Forest
TIME: 17s
RMSE: 198.59
TIME: 574s
RMSE: 94.92
TIME: 1447s
RMSE: 64.43

Visualization

8 instances of m5.2xlarge



https://plot.ly/~shirleygogogo/1/

Conclusion and Lesson Learned

1. Analysis Topic
    - A combination of census, rent, and insurance marketplace data can be used to build a surprisingly good model (RMSE less than $100). But we would need **more time** to investigate further if there was an accidental data leakage in the model.
2. Distributed computing (EMR)
    - For our specific analysis, the sweet spot in terms of number of nodes to use is around **16** m5.2xlarge nodes.

3. Distributed modeling (Spark vs H2O)
    - The H2OContext has an attitude problem on EMR. (It works when it feels like it)
    - H2O models might take more time to train but the code is much more easy to develop with much more options to configure.
    - Spark data frames load significantly faster than H2O data frames and are much easier to work with for pre-processing.