COL774

2022EE11152 - 2022CS11621

Pratham Malhotra - Jakharia Rishit

1 Introduction

In this part of the assignment, we explore feature creation and feature selection approaches, along with outlier analysis of the input dataset. The objective is to get the best possible prediction on the unseen test set. We explored:

1. Feature Creation
   (a) One-Hot Encoding
   (b) Target Encoding
   (c) Relational Encoding
   (d) Polynomial Features
2. Feature Selection
   (a) Correlation
   (b) LASSOLARS for feature selection
3. Data Preprocessing
   (a) Outliers removal using Inter Quartile Ranges
   (b) Normalization of input data
4. Model Training
   (a) Linear Regression
   (b) Ridge Regression
   (c) Elastic Net Regression
   (d) LASSO LARS Selection

2 Feature Creation

2.1 One-Hot Encoding

All categorical features were ordinally encoded, which instills a sense of order between the training examples; however, for nominal features in this data, this can be a problem; hence, one-hot encoding was employed for the following columns.

- Hospital Service Area
- Patient Disposition
- Race, Ethnicity, Gender
- Age Group
- APR MDC Code, APR MDC Description
- APR Severity of Illness Code, APR Severity of Illness Description

- APR Risk of Mortality
- APR Medical Surgical Description
- Payment Typology 1, Payment Typology 2, Payment Typology 3
- Type of Admission

2.2 Target Encoding

While the input data has ordinal encoding, we do not know if the order is correct. To ensure this, we implement target encoding

- Hospital County
- Zip Code - 3 digits
- Operating Certificate Number
- Permanent Facility Id
- Facility Name
- CCSR Diagnosis Code
- CCSR Procedure Code
- CCSR Diagnosis Description
- CCSR Procedure Description
- APR DRG Code
- APR DRG Description

2.3 Smooth Target Encoding

Here are some problems with Target encoding as it is, mainly because our dataset has many outliers.

Table 1: Target vs Smooth Target Encoding

Feature	Target Encoding	Smooth Target Encoding
Overfitting	Prone to overfitting	Less prone to overfitting
Rare Categories	Sensitive to rare categories	Handles rare categories better

Thus, here, we have used Smooth target encoding, implemented below,

$$E[Y|X = x_i] = \frac{\sum_{j=1}^{n_i} y_{ij}}{n_i}$$

$$n_i = \text{count of category } x_i$$

$$\hat{y}_i = \frac{n_i E[Y|X = x_i] + \alpha\mu}{n_i + \alpha}$$

Where α is the smoothing factor

2.4 Relational Encoding

Attempted to create Logical features out of the feature given in the dataset, such as:

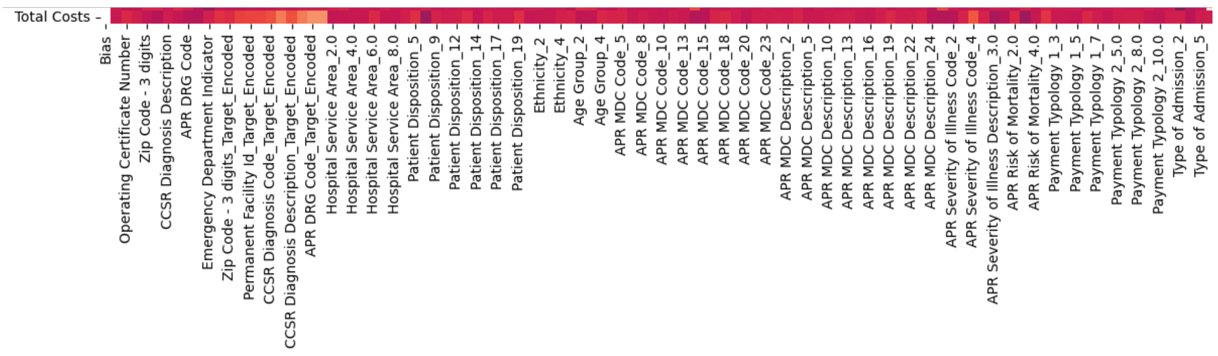
$$House\_Service\_Area * House\_County$$

However, these features had a low correlation value and, thus, were removed

2.5 Polynomial Features

Tested Features like $Hospital\_Service\_Area^2$ and $Hospital\_County^2$ based on the graphs.
However, these features had a low correlation value and, thus, were removed.

3 Feature Selection



3.1 Correlation Analysis

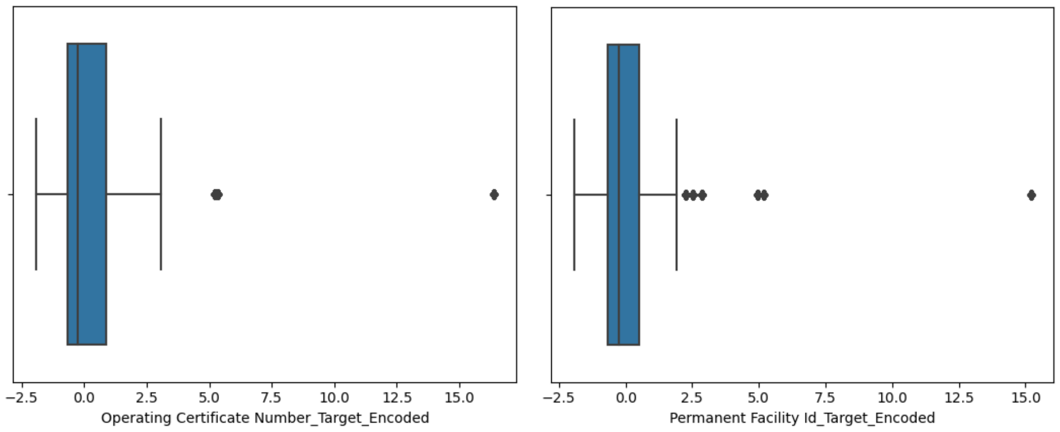
- From the plot, we see that not all features have a good correlation with Total Costs
- Any feature with a correlation value less than 0.01 is removed at this stage

3.2 Lasso Lars Selection

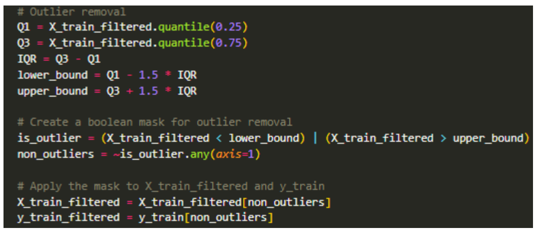
- For features selected after preliminary Correlation Analysis, we select features using Lasso LARS
- Any feature with a coefficient value less than 0 is removed at this stage

4 Data Preprocessing

4.1 Outliers removal using Inter Quartile Ranges



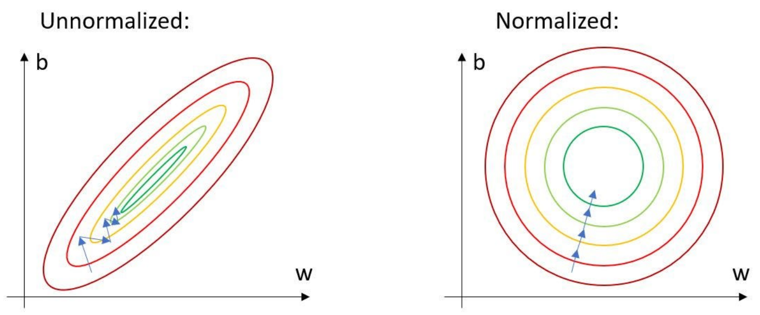
As evident from the above box plots of two selected features, our training set still had outliers. To take care of these, we use Interquartile range analysis to remove the outliers outside the lower or upper bound, using the code below.


```
# Outlier removal
Q1 = X_train_filtered.quantile(0.25)
Q3 = X_train_filtered.quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Create a boolean mask for outlier removal
is_outlier = (X_train_filtered < lower_bound) | (X_train_filtered > upper_bound)
non_outliers = ~is_outlier.any(axis=1)

# Apply the mask to X_train_filtered and y_train
X_train_filtered = X_train_filtered[non_outliers]
y_train_filtered = y_train[non_outliers]
```

4.2 Normalization of the train sets and test sets

The necessity of data normalization arises from the inherent differences in the scales of various features within a dataset. For instance, consider a dataset containing features such as age (0-100) and income (thousands to millions). Larger-scale features can dominate the learning process without normalization, leading to sub-optimal model performance.

$$X_{norm} = \frac{X - \mu}{\sigma + \epsilon}$$



Credit: John Willaert
(Medium: How To Calculate the Mean and Standard Deviation—Normalizing Datasets in Pytorch)

5 Models

The below models were tried in Kaggle's environment, and the following Error values were achieved using the 'grade_c.py' file given with the assignment.

Table 2: Model Comparison

Model	Error
LassoLARSCV	9,910.32
Ridge(alpha=2100)	10,197.84
Linear Regression	3,54,94,61,36,310.22
ElasticNetCV	11,777.49