MODEL DOCUMENTATION REPORT

Zaid Saleh

November 2023

1 Linear Regression Model

Purpose: The purpose of the linear regression model is to predict insurance charges based on features like age, sex, BMI, etc. The features (denoted as X) and the target variable (denoted as Y) are taken from a dataframe.

1.1 Training the Model:

The model training involves the fit function which calculates the optimal weights (w) using the normal equation method. This method involves matrix operations such as transpose, multiplication, and inversion to derive a closed-form solution that minimizes the sum of squared residuals between the observed and predicted values.

1.2 Mathematical Formulation:

The model uses the standard linear regression equation given by the following loss function:

$$J(w) = \sum_{i=1}^{N} (y_i - w^T x_i)^2$$

In matrix form, for the entire model, the loss function can be written as:

$$J(w) = (y - Xw)^T (y - Xw)$$

Where:

- y is the vector of observed values.
- X is the matrix of input features with a column of ones added to include the bias term.
- w is the vector of weights corresponding to the input features and the bias term.
- N is the number of observations in the dataset.

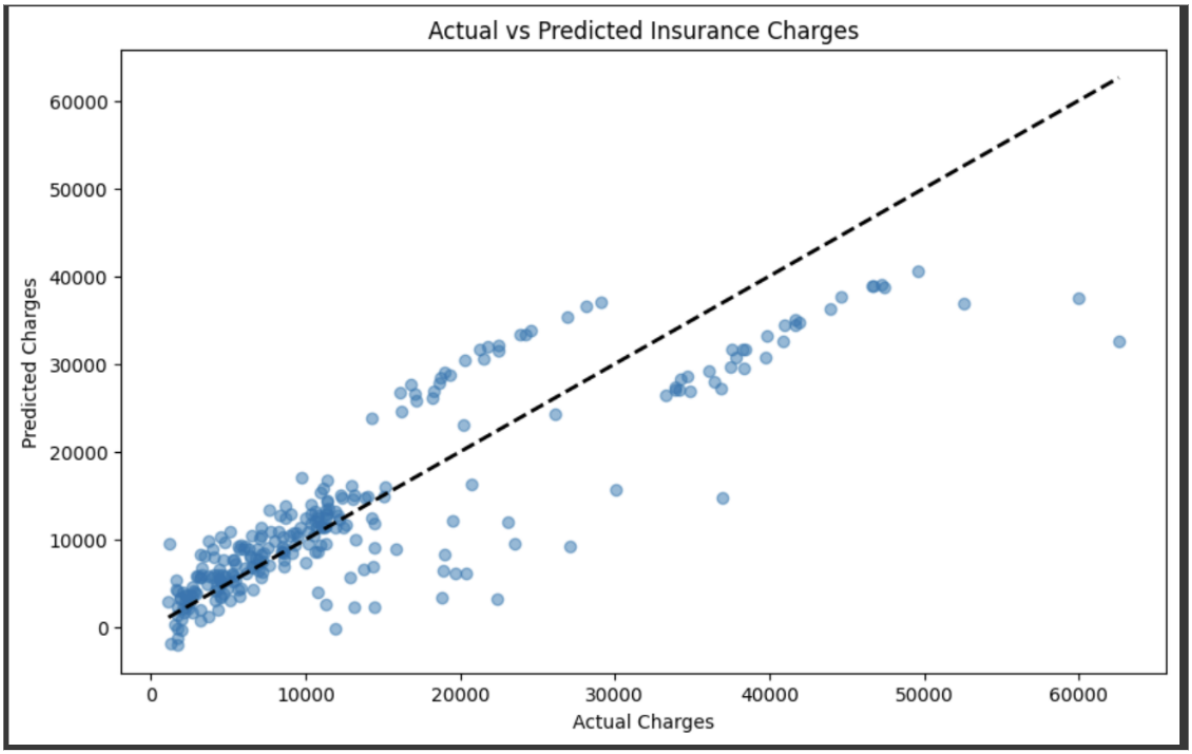
1.3 Making Predictions:

The trained weights obtained from the fit function are used to make predictions on new data. This is done by multiplying the new input features (X_{test}) with the trained weights (w).

1.4 Model Evaluation:

The performance of the model is evaluated using the Root Mean Squared Error (RMSE), which measures the average magnitude of the errors between the predicted and actual values. The RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - w^T x_i)^2}$$



1.5 User Input Prediction:

The model also includes a function for predicting charges based on new user inputs. This function applies the trained weights to the input features provided by the user to predict the charges.

2 Ridge Regression Model

Purpose: To enhance the Linear Regression model by incorporating L2 regularization to address potential overfitting.

2.1 Mathematical Formulation:

The Ridge Regression model adds a regularization term to the cost function, defined as:

$$J(w) = \sum_{i=1}^{N}(y_i - w^T x_i)^2 + \lambda \|w\|^2$$

where $\|w\|^2$ denotes the L2 norm of the weight vector, also known as the Ridge penalty. The regularization term λ controls the magnitude of this penalty.

The L2 norm is given by:

$$\|w\|^2 = \sum_{j=1}^{p} w_j^2$$

where p is the number of predictors.

The OLS (Ordinary Least Squares) equation for Ridge regression is estimated as:

$$w = (X^T X + \lambda I)^{-1} X^T y$$

which is the closed-form solution, where I is the identity matrix.

2.2 Model Training

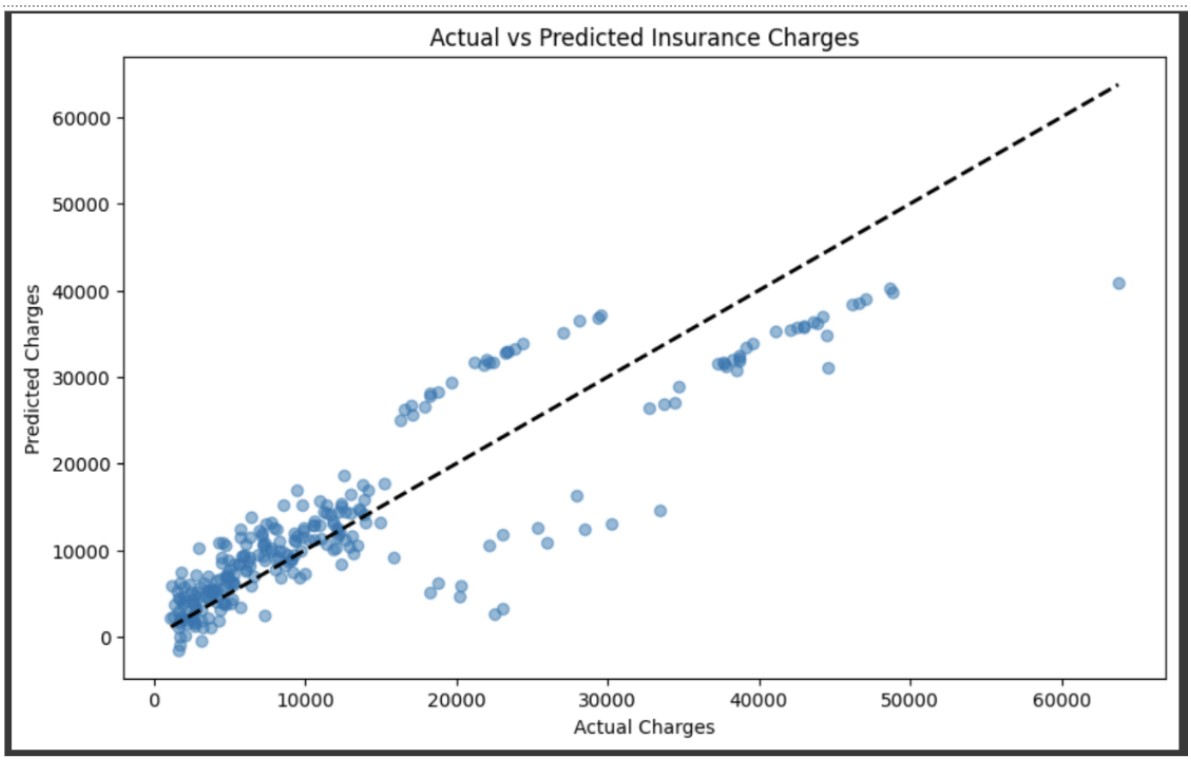
- Split data into train and test sets.
- Standardize numerical features like BMI, age.
- One-hot encode categorical features like gender.
- Fit the ridge regression model on the training set.
- Tune regularization hyperparameter λ via cross-validation.

2.3 Cross-validation for Hyperparameter Tuning:

The optimal value of λ is found using cross-validation, often k-fold cross-validation, where the dataset is split into k smaller sets. The performance metric (RMSE) is computed for each fold, and the λ that results in the lowest average RMSE across all folds is selected.

2.4 Model Evaluation

- Evaluate the model on the test set using RMSE.
- RMSE measures deviation of predictions from true charges.
- Lower RMSE indicates better generalization performance.
- Compare to baseline model like linear regression without regularization.
- Monitor model performance across random train/test splits.



2.5 Gradient Descent Optimization:

In scenarios where the closed-form solution is computationally expensive due to a large number of features, Ridge Regression can be optimized using gradient descent:

$$\nabla_w J(w) = -2X^T(y - Xw) + 2\lambda w$$

where $\nabla_w J(w)$ is the gradient of the cost function with respect to the weights w. Gradient descent iteratively adjusts the weights by moving in the opposite direction of the gradient.

2.6 Making Predictions

- Accept new patient data as input.
- Preprocess inputs per pipeline (standardization, encoding).
- Pass inputs through trained ridge regression model.
- Model generates predicted healthcare charges as output.
- Can make predictions for new patients to estimate charges.

Additional Information:

- Ridge regression is especially useful when there is multicollinearity in the data, as the regularization term shrinks the coefficients of correlated predictors and helps in reducing overfitting.
- The choice of λ is critical in ridge regression. It is often selected via cross-validation, where different values of λ are tried, and the one that results in the lowest cross-validated RMSE is chosen.

3 Neural Network Model:

Neural networks model is a model that is created inspired by the structure of the human brain, designed to recognize patterns. Using the model I tried to predict the medical charges of individuals over the year. The basic structure includes input layers, hidden layers, and an output layer, with each layer consisting of nodes (neurons) connected with weights. The mathematical representation involves:

- Weighted sum of inputs: $z = \sum w_i x_i + b$
- Activation function: $a = f(z)$

Where:

- x_i are inputs.
- w_i are weights.
- b is bias.
- f is an activation function like ReLU, Sigmoid, or Tanh.

Neural networks learn the optimal weights and biases through backpropagation and gradient descent algorithms, minimizing a loss function like Mean Squared Error or Cross-Entropy Loss.

3.1 Loss Function

- Mean Squared Error (MSE) for regression:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Cross-Entropy Loss for classification:

$$Cross - Entropy = -\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

3.2 Activation Functions

- ReLU: $ReLU(x) = \max(0, x)$
- Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$
- Tanh: $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

3.3 Backpropagation and Gradient Descent

- Update rule for weights: $w_{new} = w_{old} - \eta \frac{\partial Loss}{\partial w}$
- Update rule for biases: $b_{new} = b_{old} - \eta \frac{\partial Loss}{\partial b}$

3.4 Regularization

- L2 Regularization (Ridge): $RegularizationTerm = \lambda \sum w_i^2$
- Dropout: Technique to prevent overfitting by dropping randomly selected neurons or nodes during training.

3.5 Optimization Algorithms

- Stochastic Gradient Descent (SGD)
- Adam (Adaptive Moment Estimation)
- RMSprop (Root Mean Square Propagation)

3.6 Performance Metrics

- For regression: RMSE (Root Mean Squared Error)

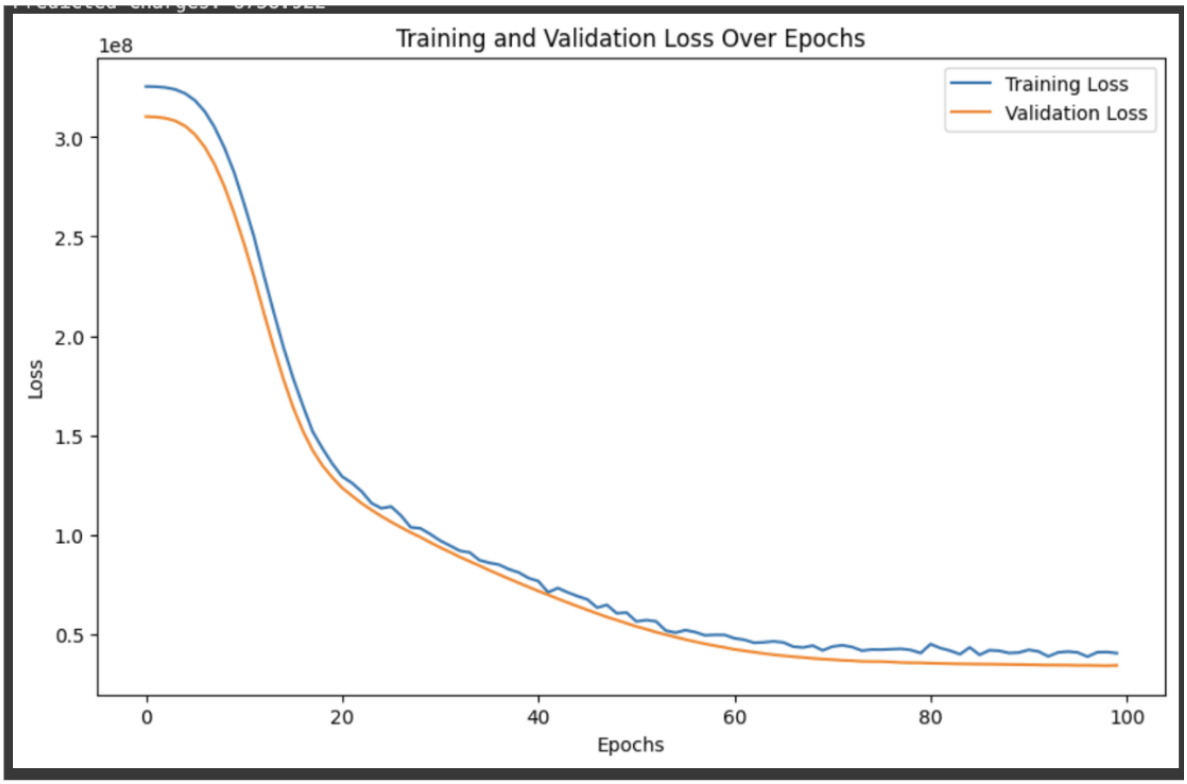


FIGURE: graphical representation of the training and validation loss of a neural network model over the course of its training epochs.