MINI-PROJECT 2: YELP BUSINESS RATING PREDICTION USING TENSORFLOW

Version 1.0

CSC215, Fall 2018

Oct 8th, 2018

Prepared by: Chandini Nagendra

Siddharth Chittora

Contents

1. Problem Statement

In this project, we aim to predict a business's stars rating using the reviews of that business and review count based on neural network implementation in Tensorflow. This project is twofold:

Task 1: Consider this problem as a regression problem. Compare the RMSE of the BEST **Tensorflow regression neural network model** you obtained with that of **regression model** you achieved in the last project.

Task 2: Consider this problem as a classification problem. Compare the accuracy of the BEST **Tensorflow classification neural network model** you obtained with that of **each classification model** you achieved in the last project.

2. Methodology

Here we compare Linear and logistic models with Tensor flow models by using early stopping, Model checking and tuning the models with hyperparameters and see how they affect performance.

- we are using data for 10000 businesses.
- For regression problem we worked with linear regression and logistic regression.
- we created tensorflow model with activation function ReLU to compare the best RMSE of the earlier regression models with this model.
- we used Early Stopping and checkpointing with ReLU to see how it affected the model.
- now we tried different optimizers like, adam, SGD, RMSprop, Adagrad, Adamax, Adadelta, Nadam.
- we also experimented with multiple hidden layers and the number of neurons. The experimental results are shown below.
- after learning that the model gave best performance with optimizer Adam and four hidden layers, we then experimented with activation function Sigmoid and Tanh.
- Tanh is also like logistic sigmoid. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped) hence we used the zscore normalized review count for Tanh
- For Classification problem we implemented KNN, SVM, MNB.
- we created tensorflow model for classification model using activation function Softmax.
- we used optimizer adam and four hidden layers as we learnt from our experiment that they give the best result.

2.1. Regression Problem

2.1.1. Linear Regression

Root Mean Squared Error: 0.56

R2 score: 0.70

2.1.2. Logistic Regression:

Root Mean Squared Error: 1.38

R2 score: 0.54

2.1.3. Tensor flow regression neural network models

Activation: ReLu

Optimizer: adam

Without stopping, checkpointing

Root Mean Squared Error: 0.5732024908065796

R2 score: 0.68

With stopping, checkpointing

Root Mean Squared Error: 0.5019386410713196

R2 score: 0.75

With stopping, checkpointing

Optimizer: SGD

Root Mean Squared Error: 0.5619208812713623

R2 score: 0.69

Optimizer: RMSprop

Root Mean Squared Error: 0.5563821196556091

R2 score: 0.70

Optimizer: Adagrad

Root Mean Squared Error: 0.5838866233825684

R2 score: 0.67

Optimizer: Adadelta

Root Mean Squared Error: 0.5558121800422668

R2 score: 0.70

Optimizer: Adamax

Root Mean Squared Error: 0.5647493004798889

R2 score: 0.69

Optimizer: Nadam

Root Mean Squared Error: 0.5831780433654785

R2 score: 0.67

From these results adam optimizer performed the best, so we use that to continue our trials

With 2 hidden layers

Root Mean Squared Error: 0.6047360897064209

R2 score: 0.65

With 3 hidden layers

Root Mean Squared Error: 0.59750896692276

R2 score: 0.65

With 4 hidden layer

Root Mean Squared Error: 0.5962139368057251

R2 score: 0.66

With 5 hidden layers

Root Mean Squared Error: 0.5959509611129761

R2 score: 0.66

Activation: Sigmoid

Optimizer: adam

Without stopping, checkpointing and 4 hidden Layers

Root Mean Squared Error: 0.5019386410713196

R2 score: 0.75

With stopping, checkpointing

Root Mean Squared Error: 0.5323189496994019

R2 score: 0.73

Activation: Tanh

Optimizer: adam

Without stopping, checkpointing and 4 hidden Layers

Root Mean Squared Error: 0.7059175372123718

R2 score: 0.53

With stopping, checkpointing

Root Mean Squared Error: 0.4864327609539032

R2 score: 0.78

2.2. Classification Problem

2.2.1. KNN

Accuracy score: 0.495

Precision score: 0.49186543188663984

Recall score: 0.495

F1 score: 0.48906518574106805

2.2.2. SVM

Accuracy score: 0.4365

Precision score: 0.43072237226007926

Recall score: 0.4365

F1 score: 0.4278230138666609

2.2.3. Naïve Bayes

Accuracy score: 0.3325

Precision score: 0.29977451126707483

Recall score: 0.3325

F1 score: 0.283143370733679

2.2.4. Tensor flow classification neural network models with stopping and checkpointing

Activation: ReLu

Optimizer: adam

Accuracy score: 0.495

Precision score: 0.49186543188663984

Recall score: 0.495

F1 score: 0.48906518574106805

Activation: Sigmoid

Optimizer: adam

Accuracy score: 0.48

Precision score: 0.4761277993991939

Recall score: 0.48

F1 score: 0.46547981723758886

Activation: Tanh

Optimizer: adam

Accuracy score: 0.4845

Precision score: 0.469104623498867

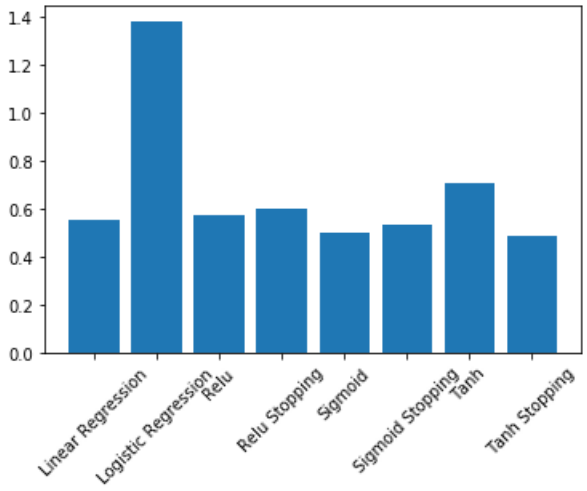
Recall score: 0.4845

F1 score: 0.46874008746379436

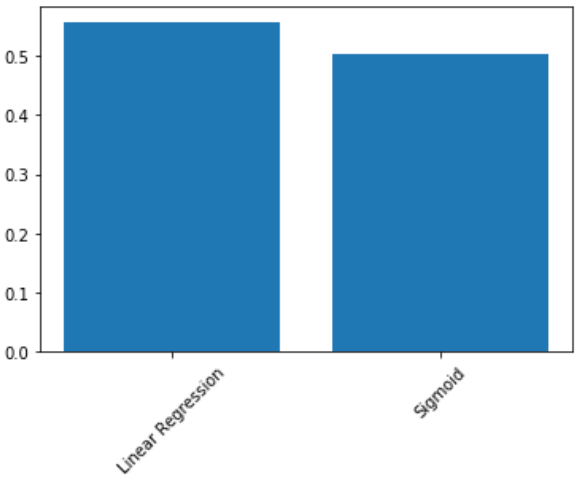
3. Experimental Results and Analysis

3.1. Regression

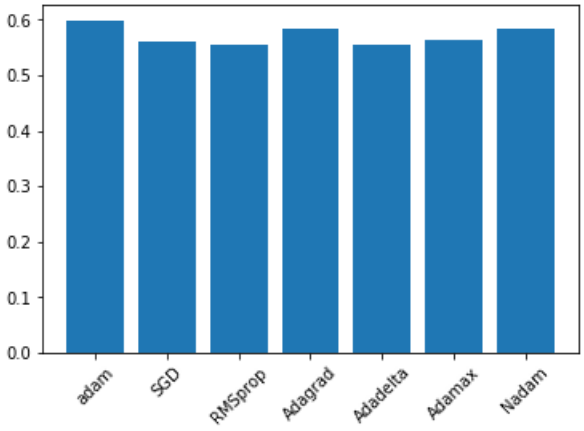
- Comparison of RMSE between all the regression models



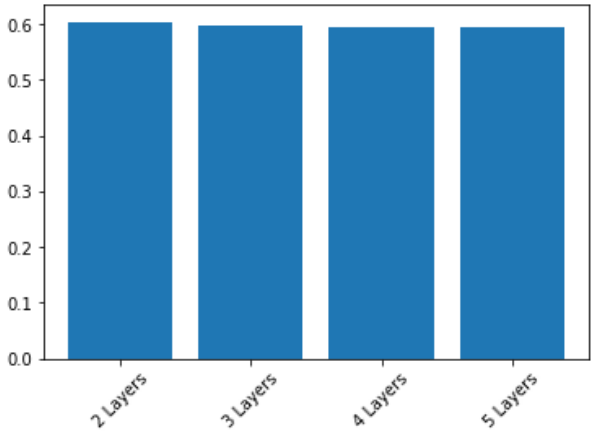
- Comparison of RMSE between the best Tensorflow model with the best Classical Regression model



- Comparison of RMSE between different optimizers



- Comparison of RMSE between models with different number of hidden layers

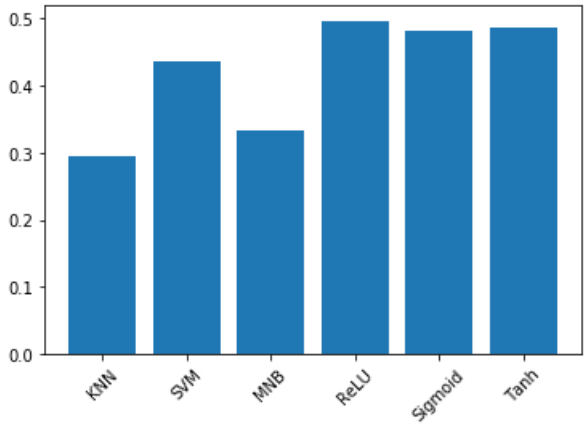


Analysis and Result:

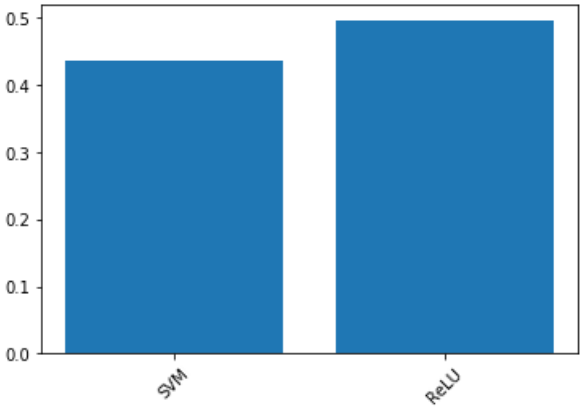
From our experiments for regression problems we observed that Sigmoid with early stopping and checkpointing with optimizer adam and 4 hidden layers had the least RMSE. Linear regression had the next best RMSE score.

3.2. Classification

- Comparison of accuracy between all the classification models



- Comparison of accuracy between best classification models



Analysis and Result:

From our experiments for classification problem we observed that ReLU with early stopping and checkpointing with optimizer adam and 4 hidden layers had the best accuracy. SVM had the next best accuracy.

Conclusion:

From our experiments we observed that regression model best fit the given problem statement.

4. Task Division

4.1. Chandini Nagendra:

- Comparison of Linear regression model with tensorflow model
- Report

4.2. Siddharth Chittora

- Comparison of Classification Model with tensor flow model
- Report

Discussed together on how to improve the model and came up with the solution discussed in the additional features section.

5. Project Reflection

- In Mini project 1, we were extracting only the primary category from each row in the categories column. In this project we are using multilabel binarizer to extract all the categories.
- when using tensorflow for classification, we found that if label encoded stars are used as output directly, throws dimesion error. So we one hot code the label encoded stars and use this as our output.
- If we use the same best weight HDF5 file to save the best weight for multiple models it does not give accurate results hence we have to reinitialize and recreate the file each we run a model.
- Tanh is also like logistic sigmoid. The range of the tanh function is from (-1 to 1). tanh is also sigmoidal (s - shaped) hence we used the zscore normalized review count for Tanh

6. Additional Features

- We implemented the L1/L2 regularization for the original dataset.
- We also implemented Dropout to reduce the effect of overfitting.
- We processed postal code, we performed One hot Coding on it to extract features.
- we processed categories using Multilabel Binarizer to extract features.
- we merged it with the original matrix and used this matrix.
- we used activation function ReLU and Sigmoid on the new matrix we created with postal code and categories.
- we also performed regularization and dropout on the new matrix.

Regularization

Root Mean Squared Error: 0.5475241541862488

R2 score: 0.72

Dropout

Root Mean Squared Error: 0.5154887437820435

R2 score: 0.75

With Postal code and categories

Activation: ReLU

Root Mean Squared Error: 0.5733668208122253

R2 score Sigmoid: 0.69

Activation: Sigmoid

Root Mean Squared Error: 0.5733668208122253

R2 score: 0.69

Regularization

Root Mean Squared Error 0.5948136448860168

R2 score: 0.66

Dropout

Root Mean Squared Error 0.6095702648162842

R2 score: 0.65