Shreeya Badhe

badhes@oregonstate.edu

934558255

Homework 3

1. **What exactly is this RMSLE error? (write the mathematical definition)**

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(x_i+1) - \log(y_i+1))^2}$$

Here
xi = predicted value,
yi = actual value
n = number of observations

2. **What's the difference between RMSLE and RMSE?**
MSLE metric only considers the relative error between and the Predicted and the actual value and the scale of the error is not significant. On the other hand, RMSE value Increases in magnitude if the scale of error increases.

3. **Why does this contest adopt RMSLE rather than RMSE?**
RMSLE is well-suited for predicting housing prices because it treats predictions for both high-priced and low-priced houses fairly. This metric minimizes the disproportionate influence of large price values, ensuring a balanced evaluation across the entire price range.

4. **One of our TAs got an RMSLE score of 0.11 and was ranked 28 in Spring 2018. What does this 0.11 mean intuitively, in terms of housing price prediction error?**
This 0.11 means that on average, TA's prediction of a house prices is within ~11% of actual price.

5. **What are your RMSLE error and ranking if you just submit sample_submission.csv?**

6. **Why do you need to work in the log output space?**
   Working in the log output space ensures that the model treats predictions across the target range more equitably, captures proportional relationships, and improves its ability to generalize while reducing the impact of skewness and outliers.

PART 2

1. **How many features do you get?**
   I got 7226 features

2. **How many features are there for each field?**
   [15, 5, 108, 989, 2, 3, 4, 4, 2, 5, 3, 25, 9, 8, 5, 8, 10, 9, 110, 61, 6, 8, 15,
   16, 4, 305, 4, 5, 6, 5, 5, 5, 7, 601, 7, 131, 730, 686, 6, 4, 2, 6, 721, 390, 21,
   810, 4, 3, 4, 3, 8, 4, 4, 12, 7, 4, 6, 7, 97, 4, 5, 422, 6, 6, 3, 253, 193, 116,
   17, 72, 8, 4, 5, 5, 21, 12, 5, 9, 6]

3. **Train linear regression using sklearn.linear_model.LinearRegression or np.polyfit on my_train.csv and test on my_dev.csv. What's your root mean squared log error (RMSLE) on dev? (Hint: should be ~0.152).**

```
rmsle
```

```
0.151767269829436
```

4. **What are your top 10 most positive and top 10 most negative features? Do they make sense?**

```
top_positive_features
```

```
[('cat__RoofMatl_Membran', 0.6138040931984234),
 ('cat__PoolQC_nan', 0.5857375561380904),
 ('cat__RoofMatl_Metal', 0.4766955136957093),
 ('cat__Condition2_PosA', 0.44914171371966366),
 ('cat__RoofStyle_Shed', 0.3428110429792528),
 ('cat__GarageQual_Ex', 0.3214779372276497),
 ('cat__RoofMatl_Roll', 0.2843175132514184),
 ('cat__RoofMatl_WdShngl', 0.2680008906634073),
 ('cat__RoofMatl_Tar&Grv', 0.2453928459889626),
 ('cat__Condition2_Feedr', 0.21672019951174254)]
```

```
top_negative_features
```

```
[('cat__RoofMatl_ClyTile', -2.2675123648575184),
 ('cat__Condition2_PosN', -0.721626150062488),
 ('cat__Condition2_RRAe', -0.4860716738296085),
 ('cat__MSZoning_C (all)', -0.3322074829929592),
 ('cat__PoolQC_Fa', -0.2956236189339175),
 ('cat__Functional_Sev', -0.22207423471617227),
 ('cat__GarageCond_Ex', -0.22201823858222305),
 ('cat__Functional_Maj2', -0.20296616514256874),
 ('cat__Electrical_Mix', -0.18921839058305762),
 ('cat__Exterior1st_BrkComm', -0.18070478261372394)]
```

Yes they make sense for negative features they indicate undesirable conditions, and for positive features they represent desirable qualities like larger spaces, better condition, or prime location.

5. **Do you need to add the bias dimension (i.e., augmented space) explicitly like in HW2, or does your regression tool automatically handle it for you? What's your feature weight for the bias dimension? Does it make sense?**
   No, we don't need to manually add the bias term since the regression model automatically incorporates it. The weight for the bias feature in my model is 12.17. This value makes sense, as it suggests that the baseline price starts at around $160,000, which aligns with the price of an affordable house.

6. **What's the intuitive meaning (in terms of housing price) of this bias feature weight?**
   The bias weight in housing price prediction represents the baseline price of a house when all other factors are neutral or have no effect. It reflects the starting price, accounting for inherent or unmeasured factors influencing housing prices, such as location appeal, market trends, or fundamental property value, which are not explicitly captured by the model's features.

7. **Now predict on test.csv, and submit your predictions to the kaggle server. What's your score (RMSLE, should be around 0.16) and ranking?**

My Rmsle value is 15.7% and my ranking is 3891

Part 3

1. **What are the drawbacks of naive binarization?**
   The primary drawback is that numerical features like OverallQual, OverallCond, and LotArea have a direct correlation with SalePrice. When these features are converted into one-hot encoded representations through binarization, their intrinsic relationships with SalePrice are simplified and lost.
   Additionally, binarizing these features considerably increases the number of sparse features, primarily consisting of zero values. This abundance of zeros adds little value to the model's accuracy while significantly increasing computational costs.

2. **Now binarize only the categorical features, and keep the numerical features as is. What about the mixed features such as LotFrontage and GarageYrBlt?**
   When mixed features include 'NA' instead of a numerical value, it typically indicates the absence of a specific attribute. To handle this, I replaced the 'NA' values.

3. **Redo the following questions from the naive binarization section. (Hint: the new dev error should be around 0.14, which is much better than naive binarization).**

   (a) How many features are there in total?
   There are 316 features.

   (b) What's the new dev error rate (RMSLE)?

```
rmsle
```

```
0.1293408716697332
```

   (c) What are the top 10 most positive and top 10 most negative features? Are they different from the previous section? (0.5 pts)

```
top_positive_features
```

```
[('cat__RoofMatl_Membran', 0.6138040931984234),
 ('cat__PoolQC_nan', 0.5857375561380904),
 ('cat__RoofMatl_Metal', 0.4766955136957093),
 ('cat__Condition2_PosA', 0.44914171371966366),
 ('cat__RoofStyle_Shed', 0.3428110429792528),
 ('cat__GarageQual_Ex', 0.3214779372276497),
 ('cat__RoofMatl_Roll', 0.2843175132514184),
 ('cat__RoofMatl_WdShngl', 0.2680008906634073),
 ('cat__RoofMatl_Tar&Grv', 0.2453928459889626),
 ('cat__Condition2_Feedr', 0.21672019951174254)]
```

```
top_negative_features
```

```
[('cat__RoofMatl_ClyTile', -2.2675123648575184),
 ('cat__Condition2_PosN', -0.721626150062488),
 ('cat__Condition2_RRAe', -0.4860716738296085),
 ('cat__MSZoning_C (all)', -0.3322074829929592),
 ('cat__PoolQC_Fa', -0.2956236189339175),
 ('cat__Functional_Sev', -0.22207423471617227),
 ('cat__GarageCond_Ex', -0.22201823858222305),
 ('cat__Functional_Maj2', -0.20296616514256874),
 ('cat__Electrical_Mix', -0.18921839058305762),
 ('cat__Exterior1st_BrkComm', -0.18070478261372394)]
```

(d) Now predict on test.csv, and submit your predictions to the kaggle server. What's your score (RMSLE, should be ~0.13) and ranking?

Rmsle=15%, rank 2299

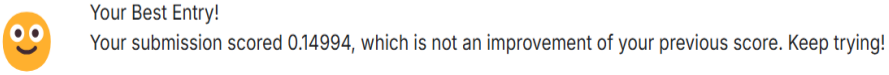
PART 4

1. **Try regularized linear regression (sklearn.linear_model.Ridge). Tune α on dev. Should improve both naive and smart binarization by a little bit.**

Naive binarization

```
rmsle_ridge
```

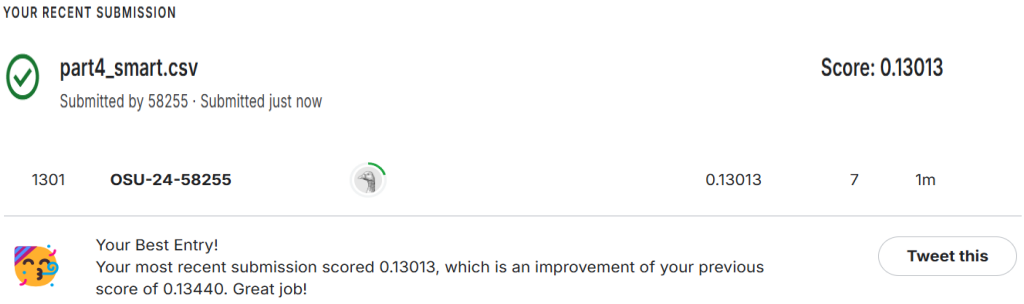
```
0.14144206793125919
```



Smart binarization

```
rmsle_ridge
```

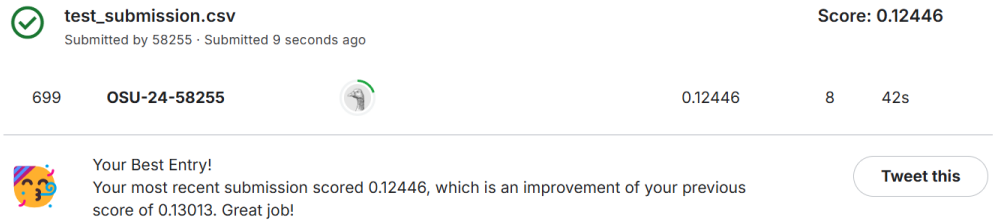
```
0.12564397238573172
```



2. **Try non-linear regression (sklearn.preprocessing.PolynomialFeatures)**

```
rmsle_score
```

```
0.11223079123201221
```



3. **How are these non-linear features (including feature combinations) relate to non-linear features in the perceptron? (think of XOR)**

   Non-linear features, in our context, capture interactions between the original features, enabling the model to learn non-linear relationships between the input variables and the target. This allows the model to better fit complex patterns in the data.

   In perceptrons, non-linearity is introduced through the use of activation functions applied to the weighted inputs. For example, without an activation function, a single-layer perceptron cannot learn the XOR function, highlighting the critical role of non-linearity in addressing complex patterns.

4. **Try anything else that you can think of. You can also find inspirations online, but you have to implement everything yourself (you are not allowed to copy other people's code).**

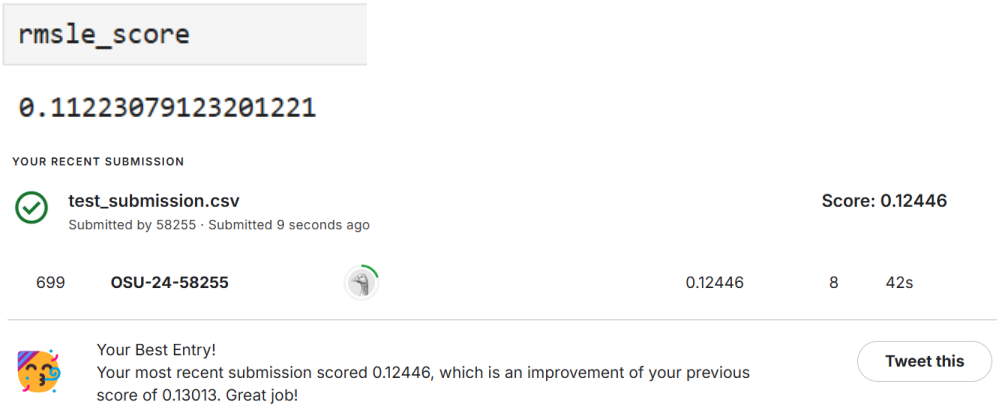
```
Model Performance (RMSLE):
Random Forest: 0.14001
Gradient Boosting: 0.12721
XGBoost: 0.13002
LightGBM: 0.12590
K-Nearest Neighbors: 0.20266
```

These are the regression algorithms with the respective rmsle values that I have tried. Out of these I chose lightgbm to predict test set.

RMSLE=12.9%

5. **What's your best dev error, and what's your best test error and ranking? Take a screen shot of your best test error and ranking, and include your best submission file. (2 pts)**

Best dev error:



PART 5

1. **Approximately how many hours did you spend on this assignment?**
   15-25 hrs.
2. **Would you rate it as easy. Moderate, or difficult?**
   Easy
3. **Did you work mostly alone, or mostly with other people?**
   I mostly work alone.
4. **How deeply do you feel you understand the material it covers (0%-100%)**
   85-90%
5. **Any Comments?**
   Hints are very helpful