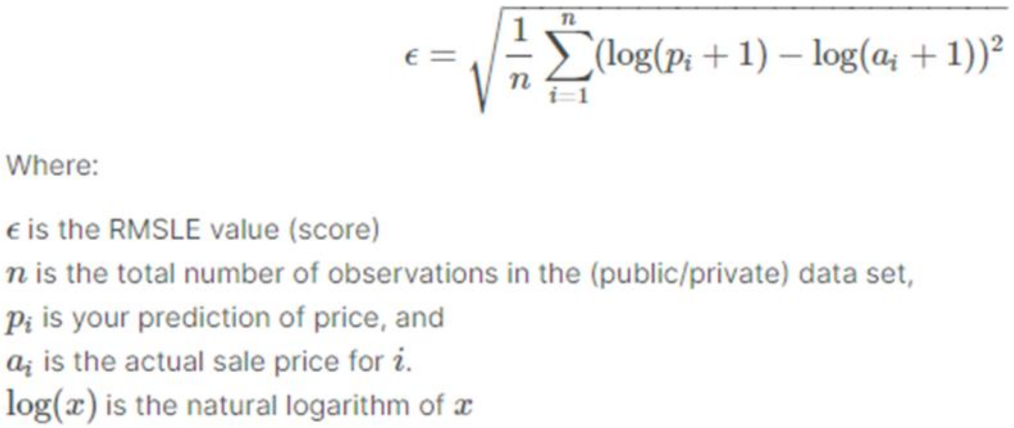
<u>Understanding the Evaluation Metric</u>

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

Answer:

The RMSLE serves as a metric for assessing how a model predicts outcomes accurately when the target variable has undergone a logarithmic transformation. This metric has great importance because it can be used for a wide range of values in a target variable. The RMSLE is calculated as follows:



$$\epsilon = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(p_i + 1) - \log(a_i + 1))^2}$$

Where:

ϵ is the RMSLE value (score)

n is the total number of observations in the (public/private) data set,

p_i is your prediction of price, and

a_i is the actual sale price for i.

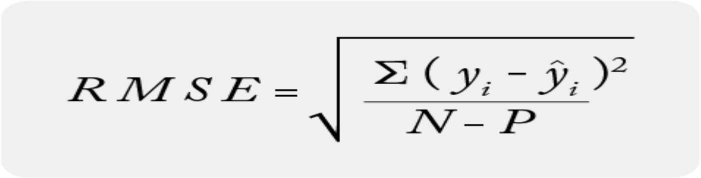
$\log(x)$ is the natural logarithm of x

2. Q: What's the difference between RMSLE and RMSE?

Answer:

The Root Mean Squared Logarithmic Error (RMSLE) and the Root Mean Squared Error (RMSE) are distinct metrics utilized for assessing regression model performance. Their dissimilarity lies in how they handle the differences between predicted and actual values. RMSLE evaluates the logarithmic differences between predicted and actual values, while RMSE measures the square root of the average squared differences between predicted and actual values.

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(\hat{y}_i + 1)\text{–} log(y_i + 1))^2}$$



$$RMSE = \sqrt{\frac{\Sigma (y_i - \hat{y}_i)^2}{N - P}}$$

3. Q: Why does this contest adopt RMSLE rather than RMSE?

Answer:

In predictive modelling the concept of “taking logs” suggests that when dealing with house price errors, predictions in estimating both expensive and inexpensive houses impact the outcome equally. The RMSLE places a higher penalty on underestimating house prices compared to overestimating them. This function of RMSLE is specifically important during the predication of property values, here avoiding undervaluing high-worth houses turns into something of great value. The emphasis of RMSLE on measuring the relative differences between predicted and actual values is in line with this consideration, emphasizing the need for precise estimations, especially for expensive properties.

4. Q: 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?

Answer:

The meaning of a RMSLE score of 0.11 is that on an average the logarithmic differences between predicted and actual values are 0.11.

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

Answer:

RMSLE - 0.40613

Ranking - 4380

5. Q: What is your “Team name” on kaggle (note this HW should be done individually)?

HaseebAB

6. Q: Hint: in machine learning, you should always use np.log() instead of math.log() because the former works with vectors and matrices (same for exp(), sum(), etc.) Extra credit question (+1 pt): Why do you need to do this?

Answer:

You should always use np.log() instead of math.log() because the functions of NumPy are used for the optimization of array operations. The tailored funtions efficiently handle the arrays, allowing for them to be streamlined and compute faster, when working with extensive datasets.By employing NumPy functions, operations can be performed on entire arrays at once through vectorization, eliminating the need for manual looping. Additionally, using NumPy functions ensures seamless compatibility with NumPy arrays, which are extensively employed in machine learning workflows.

<u>Naive data processing: binarizing all fields</u>

Q. How many features do you get?

Answer:

7227

Q. How many features are there for each field?

Answer:

(15, 5, 108, 989, 2, 3, 4, 4, 2, 5, 3, 25, 9, 8, 5, 8, 10, 9, 110, 61, 6, 8, 15, 16, 5, 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)

MSSubClass has 15 unique features

MSZoning has 5 unique features

LotFrontage has 369 unique features

LotArea has 1073 unique features

Street has 2 unique features

Alley has 3 unique features

LotShape has 4 unique features

LandContour has 4 unique features

Utilities has 2 unique features

LotConfig has 5 unique features

```
LandSlope has 3 unique features
Neighborhood has 25 unique features
Condition1 has 9 unique features
Condition2 has 8 unique features
BldgType has 5 unique features
HouseStyle has 8 unique features
OverallQual has 10 unique features
OverallCond has 9 unique features
YearBuilt has 112 unique features
YearRemodAdd has 61 unique features
RoofStyle has 6 unique features
RoofMatl has 8 unique features
Exterior1st has 15 unique features
Exterior2nd has 16 unique features
MasVnrType has 5 unique features
MasVnrArea has 335 unique features
ExterQual has 4 unique features
ExterCond has 5 unique features
Foundation has 6 unique features
BsmtQual has 5 unique features
BsmtCond has 5 unique features
BsmtExposure has 5 unique features
BsmtFinType1 has 7 unique features
BsmtFinSF1 has 637 unique features
BsmtFinType2 has 7 unique features
BsmtFinSF2 has 144 unique features
BsmtUnfSF has 780 unique features
TotalBsmtSF has 721 unique features
Heating has 6 unique features
HeatingQC has 5 unique features
CentralAir has 2 unique features
```

```
Electrical has 6 unique features

1stFlrSF has 753 unique features

2ndFlrSF has 417 unique features

LowQualFinSF has 24 unique features

GrLivArea has 861 unique features

BsmtFullBath has 4 unique features

BsmtHalfBath has 3 unique features

FullBath has 4 unique features

HalfBath has 3 unique features

BedroomAbvGr has 8 unique features

KitchenAbvGr has 4 unique features

KitchenQual has 4 unique features

TotRmsAbvGrd has 12 unique features

Functional has 7 unique features

Fireplaces has 4 unique features

FireplaceQu has 6 unique features

GarageType has 7 unique features

GarageYrBlt has 178 unique features

GarageFinish has 4 unique features

GarageCars has 5 unique features

GarageArea has 441 unique features

GarageQual has 6 unique features

GarageCond has 6 unique features

PavedDrive has 3 unique features

WoodDeckSF has 274 unique features

OpenPorchSF has 202 unique features

EnclosedPorch has 120 unique features

3SsnPorch has 20 unique features

ScreenPorch has 76 unique features

PoolArea has 8 unique features

PoolQC has 4 unique features
```

Fence has 5 unique features

MiscFeature has 5 unique features

MiscVal has 21 unique features

MoSold has 12 unique features

YrSold has 5 unique features

SaleType has 9 unique features

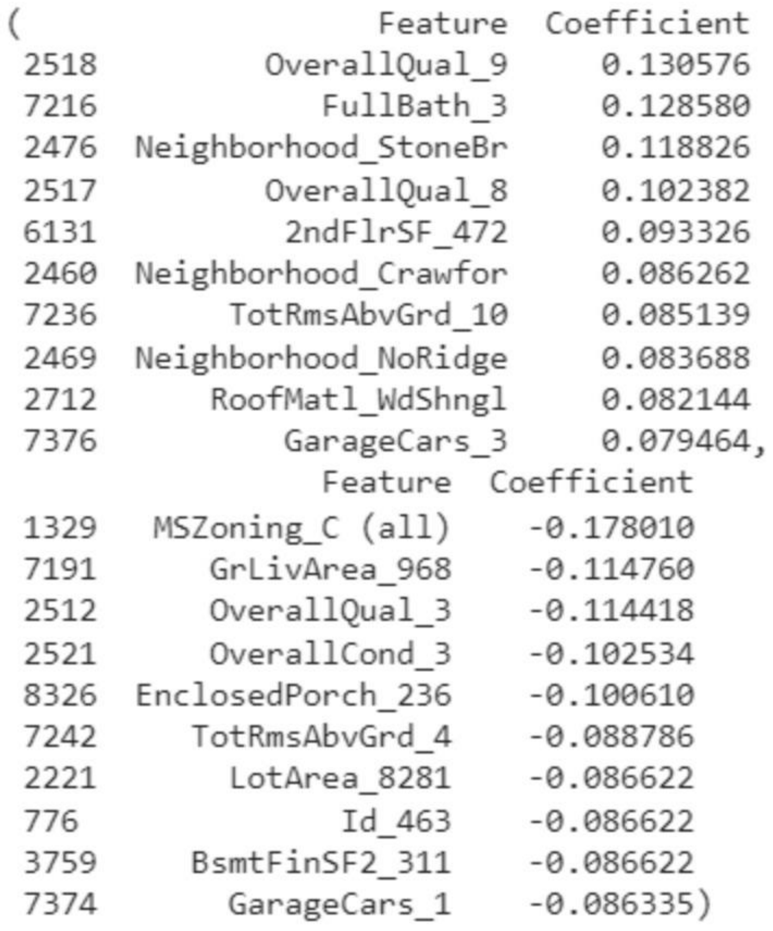
SaleCondition has 6 unique features

Q: 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?

Answer:

~ 0.153

Q: What are your top 10 most positive and top 10 most negative features? Do they make sense?



```
(                    Feature  Coefficient
 2518           OverallQual_9     0.130576
 7216              FullBath_3     0.128580
 2476    Neighborhood_StoneBr     0.118826
 2517           OverallQual_8     0.102382
 6131             2ndFlrSF_472     0.093326
 2460    Neighborhood_Crawfor     0.086262
 7236        TotRmsAbvGrd_10     0.085139
 2469    Neighborhood_NoRidge     0.083688
 2712        RoofMatl_WdShngl     0.082144
 7376            GarageCars_3     0.079464,
                   Feature  Coefficient
 1329     MSZoning_C (all)    -0.178010
 7191        GrLivArea_968    -0.114760
 2512        OverallQual_3    -0.114418
 2521        OverallCond_3    -0.102534
 8326   EnclosedPorch_236    -0.100610
 7242       TotRmsAbvGrd_4    -0.088786
 2221        LotArea_8281    -0.086622
 776                Id_463    -0.086622
 3759       BsmtFinSF2_311    -0.086622
 7374         GarageCars_1    -0.086335)
```

Positively correlated features such as OverallQual, FullBath, and GarageCars are intuitively linked to higher house prices, indicating that as these attributes improve or increase, the house's value tends to rise. On the other hand, the top 10 negatively correlated features, like MSZoning and GrLivArea, suggest characteristics associated with lower house prices. When these attributes exhibit specific qualities or values, the house tends to be priced lower, reflecting potential factors that might detract from its overall value or desirability.

Q: 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? Hint: coef_ and intercept_. What's your feature weight for the bias dimension? Does it make sense?

Answer:

Most of the ML libraries already include a bias term by default. It is generally added internally when the training process is going on, The bias term in a linear equation signifies the starting point or the y-intercept on a graph. It denotes the value that the predicted outcome would have even if all the predictor variables (features) were zero. Essentially, it accounts for the inherent offset or baseline value before considering the influence of the other variables in the linear equation. Bias term (intercept) ~ 12.02

Yes it makes sense, as a baseline prediction is given when all features are zero its not valid. This makes the model flexible and interpretable. It also makes sure the realistic predictions are given during the absence of specific features.

Q: Extra credit question (+1 pt): What's the intuitive meaning (in terms of housing price) of this bias feature weight?

Answer:

In a linear regression model predicting housing prices, the bias feature weight, often termed the intercept, symbolizes the predicted average house price when all other features lack relevance or hold a value of zero. It encapsulates the fundamental worth of a house, accounting for aspects not explicitly captured by other features. Essentially, the bias term establishes a foundational reference point for making predictions, capturing the intrinsic value of a house irrespective of the specific details represented by the remaining features.

Q: 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?

Answer:

RMSLE - 0.16406

Ranking - 3264

Smarter binarization: Only binarizing categorical features

Q. What are the drawbacks of naive binarization?

Answer:

For machine learning and data analysis purposes, it's generally advisable to maintain numerical attributes like LotArea and YrSold in their original continuous state. Employing techniques such as regression models, decision trees, or feature scaling when required allows the utilization of these continuous variables effectively. Binarization, commonly applied to categorical variables, is less suited for continuous data. Instead, it's more beneficial to reserve binarization for transforming categorical attributes into a suitable format, especially for algorithms like logistic regression. This approach helps retain the inherent nature of continuous variables and maximizes their utility in predictive models, enhancing their ability to capture nuances and patterns within the data.

Q. Now binarize only the categorical features, and keep the numerical features as is. What about the mixed features such as LotFrontage and GarageYrBlt?

Answer:

The mixed features LotFrontage and GarageYrBlt are both treatedas numerical fields using an imputer.

Q. 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).

How many features are there in total?

Answer:

244

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

Answer:

0.146

What are the top 10 most positive and top 10 most negative features? Are they different from the previous section?

Answer:

Top 10 positive:

- YearBuilt
- GarageYrBlt
- YearRemodAdd
- OverallQual
- GrLivArea
- GarageCars
- TotalBsmtSF

- GarageArea
- 1stELESF
- FullBath

Top 10 negative:

- MSSubclass
- YrSold
- OverallCond
- BsmtHalfBath
- EnclosedPorch
- KitchenAbvGr
- LowQualFinSF
- BsmtFinSE2
- 3SsnPorch
- MiscVal

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

Answer:

RMSLE ~ 0.139, Ranking - 1750

<u>Experimentation</u>

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

Improved the RMSLE ~ [0.1401 - 0.12]

Naïve ~ 0.144

Smart ~ 0.12

Q. Try non-linear regression (sklearn.preprocessing.PolynomialFeatures)

Answer:

sklearn.preprocessing.PolynomialFeatures(degree=2).

RMSLE(test/dev) -[0.12 - 0.13]

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

Answer:

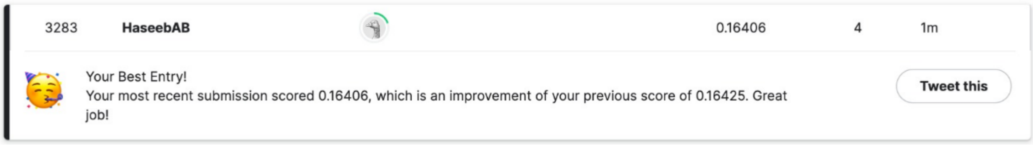
Non-linear regression, facilitated by PolynomialFeatures in sklearn.preprocessing, induces non-linear connections by generating polynomial combinations of initial features. Conversely, non-linear attributes within a perceptron originate from applying non-linear activation functions to linearly combined input features. This distinction enables models to grasp intricate patterns. The XOR problem exemplifies linear model limitations, like perceptrons, incapable of handling non-linearly separable data. Perceptrons, being linear classifiers, struggle with problems like XOR, which lacks a separable line for positive and negative instances. To resolve this, non-linear activation functions such as sigmoid, tanh, or ReLU in neural networks introduce complexities, empowering them to discern and mimic non-linear associations between predictors and outcomes.

Q. 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).

Answer:

Implemented using SVM xgboost.

Q. 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.



3283 HaseebAB 0.16406 4 1m

Your Best Entry!
Your most recent submission scored 0.16406, which is an improvement of your previous score of 0.16425. Great job!

best dev error ~ (0.16 - 0.17]

best test error - 0.16406

ranking - 3283

Debrief

1. **Approximately how many hours did you spend on this assignment?**
   Answer:

   I gave 1 hours every day to the course and on the assignment.

2. **Would you rate it as easy, moderate, or difficult?**
   Answer:

   Moderate

3. **Did you work on it mostly alone, or mostly with other people?**
   Answer:

   Mostly with other people

4. **How deeply do you feel you understand the material it covers (0%–100%)?**
   Answer:

   75%

5. Any other comments?

Not at the present.