Oregon State University

AI534 - MACHINE LEARNING

HW3 - Kaggle Competition for Housing Price Prediction

AUTHORS

Melek Derman - 934382167

November 19, 2024

Contents

1 Part 1: Understanding the Evaluation Metric 1

2 Part 2: Naive data processing: binarizing all fields 3

3 Part 3: Smarter binarization: Only binarizing categorical features 6

4 Part 4: Experimentation 8

5 Debriefing 10

1 Part 1: Understanding the Evaluation Metric

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

RMSLE (Root Mean Squared Logarithmic Error) RMSLE is defined as the square root of the average of the squared differences between the logarithmic values of the predicted $\hat{y}_i$ and the actual y_i values:

$$\text{RMSLE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log_e(\hat{y}_i + 1) - \log_e(y_i + 1)\right)^2} \tag{1}$$

1.2: What's the difference between RMSLE and RMSE?

RMSE (Root Mean Squared Error) is defined as the square root of the mean of the squared differences between the actual value y_i and the predicted value $\hat{y}_i$:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{2}$$

Since RMSLE includes a logarithmic formulation, it is more useful for larger values, while RMSE, being more sensitive to smaller differences, is better suited for predictions involving smaller numbers.

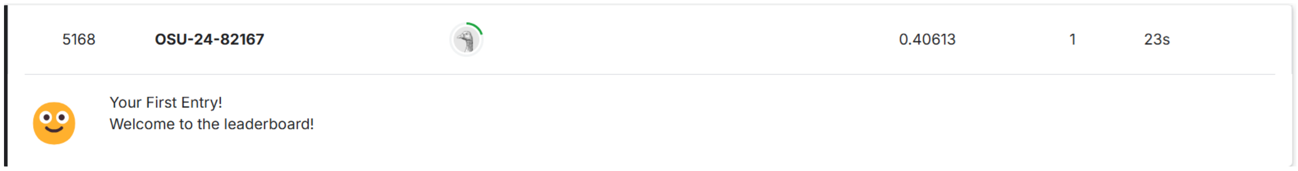
1.3: Why does this contest adopt RMSLE rather than RMSE?

Since this Kaggle competition is focused on house prices, and house prices are typically in the range of thousands of dollars, larger differences are more important than smaller ones. Therefore, using RMSLE is more appropriate for this competition.

1.4: 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?

An RMSLE score of 0.11 indicates that the model's predictions deviate from the actual housing prices by a relatively small margin, on average. This score shows that the predicted prices are quite close to the actual prices, with an average difference of 0.11 in their logarithmic values. In other words, the model provides relatively accurate results without significantly overestimating or underestimating the true values.

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



5168 OSU-24-82167 0.40613 1 23s

Your First Entry!
Welcome to the leaderboard!

Figure 1: The RMSLE error and ranking of my `sample_submission.csv` file submission

1.6 Why do you need to work in the log output space?

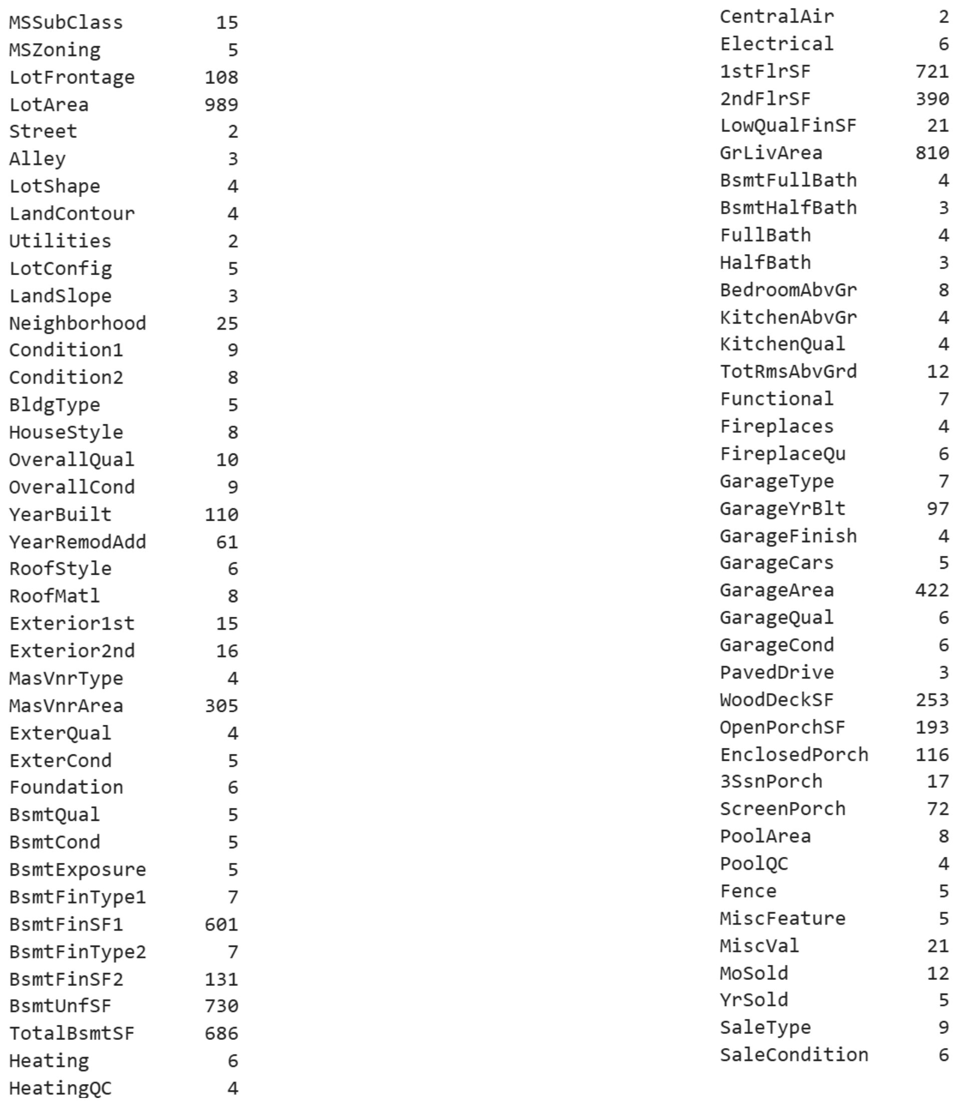
Sale Price values are typically very large, often in the hundreds of thousands of dollars, and this transformation allows the model to operate more stably during computations. It also helps to better evaluate large differences between data points. Additionally, by working on a logarithmic scale, the model can capture proportional changes, reducing the importance of small variations.

2 Part 2: Naive data processing: binarizing all fields

2.1: How many features do you get? (Hint: around 7,230).

I got 7226 features.

2.2: How many features are there for each field?



```
MSSubClass        15
MSZoning           5
LotFrontage      108
LotArea          989
Street             2
Alley              3
LotShape           4
LandContour        4
Utilities          2
LotConfig          5
LandSlope          3
Neighborhood      25
Condition1         9
Condition2         8
BldgType           5
HouseStyle         8
OverallQual       10
OverallCond        9
YearBuilt        110
YearRemodAdd      61
RoofStyle          6
RoofMatl           8
Exterior1st       15
Exterior2nd       16
MasVnrType         4
MasVnrArea       305
ExterQual          4
ExterCond          5
Foundation         6
BsmtQual           5
BsmtCond           5
BsmtExposure       5
BsmtFinType1       7
BsmtFinSF1       601
BsmtFinType2       7
BsmtFinSF2       131
BsmtUnfSF        730
TotalBsmtSF      686
Heating            6
HeatingQC          4
```

```
CentralAir         2
Electrical         6
1stFlrSF         721
2ndFlrSF         390
LowQualFinSF      21
GrLivArea        810
BsmtFullBath       4
BsmtHalfBath       3
FullBath           4
HalfBath           3
BedroomAbvGr       8
KitchenAbvGr       4
KitchenQual        4
TotRmsAbvGrd      12
Functional         7
Fireplaces         4
FireplaceQu        6
GarageType         7
GarageYrBlt       97
GarageFinish       4
GarageCars         5
GarageArea       422
GarageQual         6
GarageCond         6
PavedDrive         3
WoodDeckSF       253
OpenPorchSF      193
EnclosedPorch    116
3SsnPorch         17
ScreenPorch       72
PoolArea           8
PoolQC             4
Fence              5
MiscFeature        5
MiscVal           21
MoSold            12
YrSold             5
SaleType           9
SaleCondition      6
```

Figure 2: Number of Features for Each Field

2.3: 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 $\sim$ 0.152).

Dev RMSLE: 0.1529202024730971

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

Top 10 Most Positive Features	Top 10 Most Negative Features
OverallQual_9.0	MSZoning_C (all)
FullBath_3.0	GrLivArea_968.0
Neighborhood_StoneBr	EnclosedPorch_236.0
2ndFlrSF_472.0	OverallQual_3.0
OverallQual_8.0	BsmtFinSF2_311.0
GarageCars_3.0	LotArea_8281.0
GrLivArea_1192.0	OverallCond_3.0
RoofMatl_WdShngl	OverallQual_1.0
Neighborhood_NoRidge	LotArea_5000.0
LotArea_8029.0	YearRemodAdd_1958.0

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

The `LinearRegression` tool from `sklearn` automatically includes the bias term, so we do not need to add it explicitly. The intercept term represents the weight for the bias dimension, while `coef_` provides the weights for the other features.

Log Bias Feature Weight (Intercept):	12.216644872455392
Bias Feature Weight:	202125.6620768801

The intercept value of this model is calculated from the result of logarithmic target values; therefore, the log intercept value alone does not hold much significance. When we examine this value after applying an exponential transformation, the resulting value of 202125, although somewhat large, seems to be within reasonable limits (it is not significantly larger or smaller than the true values).

2.6: What's the intuitive meaning (in terms of housing price) of this bias feature weight?

The bias feature weight represents the model's prediction for a house when all other feature values are set to zero. In other words, it indicates the predicted value for a house that has none of the specified features (while this is not practically possible, it is a theoretical concept for this problem).

2.7: 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?

Without `astype(float)` *for numerical features:*

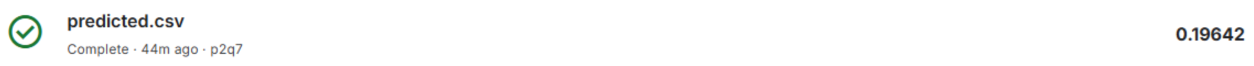

predicted.csv
Complete · 44m ago · p2q7
0.19642

Figure 3: Naive Data Processing

With `astype(float)` *for numerical features:*

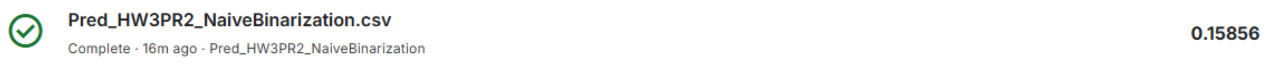

Pred_HW3PR2_NaiveBinarization.csv
Complete · 16m ago · Pred_HW3PR2_NaiveBinarization
0.15856

Figure 4: Naive Data Processing

3 Part 3: Smarter binarization: Only binarizing categorical features

3.1: What are the drawbacks of naive binarization? (Hint: data sparseness, etc.)

The drawbacks include an increase in matrix size (data sparsity), loss of information in numerical features, and an increase in computational time and model complexity.

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

The `LinearRegression` tool cannot handle missing values (NA) on its own, so the data must be preprocessed. To address this, I used `sklearn.impute`'s `SimpleImputer` for handling missing values. Specifically, I applied `SimpleImputer(strategy="most_frequent")` for categorical values and `SimpleImputer(strategy="mean")` for numerical values. However, more advanced imputation methods could be applied. For instance, for the most challenging feature, `LotFrontage`, which has one of the highest numbers of missing values, we could identify the most correlated feature (e.g., `1stFlrSF` with a correlation of 0.411183 in this dataset) and perform imputation based on that relationship to potentially improve model accuracy.

3.3: 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).
3.3a: How many features are there in total?

With smart binarization, I had 286 features.

3.3b: What's the new dev error rate (RMSLE)? (should be $\sim$0.12)

My new dev error rate was 0.12393728873166943.

3.3c: What are the top 10 most positive and top 10 most negative features? Are they different from the previous section?

Top 10 Most Positive Features	Top 10 Most Negative Features
RoofMatl_Membran	RoofMatl_ClyTile
LotArea	Condition2_PosN
GrLivArea	Condition2_RRAe
1stFlrSF	MSZoning_C (all)
RoofMatl_Metal	GarageCond_Ex
Condition2_PosA	Functional_Sev
TotalBsmtSF	Functional_Maj2
BsmtFinSF1	MiscVal
OverallQual	Exterior1st_BrkComm
RoofStyle_Shed	MiscFeature_TenC

3.3d: Now predict on test.csv, and submit your predictions to the kaggle server. What's your score (RMSLE, should be ∼0.13) and ranking?

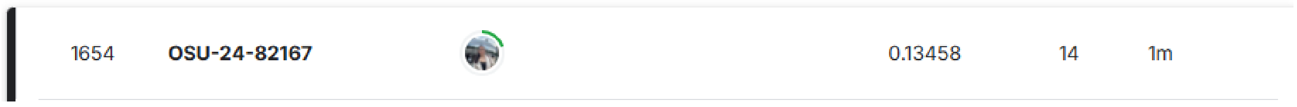

1654 OSU-24-82167 0.13458 14 1m

Figure 5: Ranking on the public leaderboard and public error of my results.

4 Part 4: Experimentation

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.

Method	Train RMSLE	Dev RMSLE
Ridge with Naive binarization	0.06369850143378249	0.1402085239568543
Ridge with Smart binarization	0.11136467152842583	0.12841107483259093

4.2:Try non-linear regression (sklearn.preprocessing.PolynomialFeatures)

There was an improvement with naive binarization, but a deterioration with smart binarization (from 0.1234 to 0.1284). I could not fully understand the reason for this. Most likely, if there are strong non-linear relationships between the features, Ridge might have reduced their importance.

Method	Train RMSLE	Dev RMSLE
Non-linear Regression (PolynomialFeatures)	0.11642560320769127	0.13568981704563762

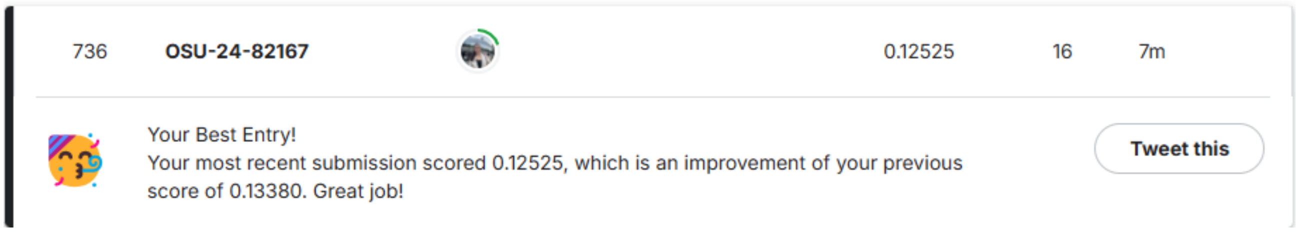
4.3: How are these non-linear features (including feature combinations) relate to non-linear features in the perceptron? (think of XOR)

According to the Perceptron Convergence Theorem, the perceptron can only work with linearly separable datasets. However, since real-world data is often not linearly separable, non-linear feature mapping is applied. To make the data linearly separable, it is transformed into a higher-dimensional space where the data becomes separable by a hyperplane. Similarly, in polynomial features, higher-order features are generated to make the data linearly separable.

4.4: 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).

I used ensemble predictions for this purpose. The final prediction was calculated as a combination of `Ridge`, `LGBMRegressor`, `XGBRegressor`, and `GradientBoostingRegressor`.

4.5: 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.



736 **OSU-24-82167** 0.12525 16 7m

Your Best Entry!
Your most recent submission scored 0.12525, which is an improvement of your previous score of 0.13380. Great job!

Figure 6: Ranking on the public leaderboard and public error of my results.

5 Debriefing

5.1: Approximately how many hours did you spend on this assignment?
I spent approximately 30 hours on this assignment.

5.2: Would you rate it as easy, moderate, or difficult?
The assignment was challenging for me. Some details in the homework explanations were insufficient for me, which made it a bit confusing. However, in the end, it was a very helpful and enjoyable experience.

5.3: Did you work on it mostly alone, or mostly with other people?
I worked on this assignment entirely on my own.

5.4: How deeply do you feel you understand the material it covers (0% - 100%)?

85%

5.5: Any other comments?

Thank you!