

COMP5212 Report

1. Introduction

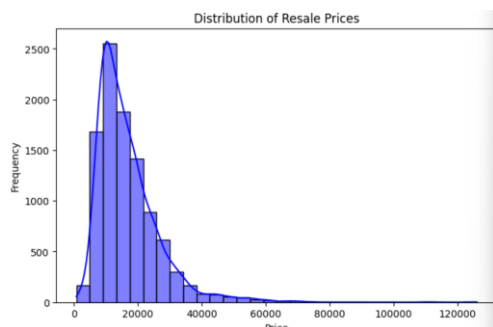
In this project, the goal was to develop a regression model to predict the resale price of agricultural equipment. The dataset contains information on various equipment listings, including features such as manufacturer, model, engine capacity, operating hours, and fuel efficiency. The primary challenge is to estimate the resale price based on these attributes, while ensuring the model generalizes well for unseen data. The project's performance is evaluated using the Root Mean Squared Error (RMSE) metric on the Kaggle leaderboard.

2. Data Preprocessing

2.1 Data Exploration

The dataset includes a mix of categorical (e.g., manufacturer, model, gearbox type) and numerical features (e.g., operating hours, engine capacity, and efficiency). Key findings from the exploratory analysis include:

1. **Target variable distribution:** Resale prices exhibit left-skewed distributions.



2. **Feature variability:** Operating hours and efficiency have high variability, with extreme outliers.

Statistical Summary (Numerical Features) - Train Data:					Statistical Summary (Numerical Features) - Test Data:				
	year	operating_hours	registration_fees	efficiency \		year	operating_hours	registration_fees	efficiency \
count	10000.000000	10000.000000	10000.000000	10000.000000	count	2000.000000	2000.000000	2000.000000	2000.000000
mean	2017.008600	22860.031100	120.922500	55.042980	mean	2017.05650	23389.791500	117.172500	55.252650
std	2.138272	20902.669316	62.246036	14.911425	std	2.13321	21671.896307	61.571408	12.719357
min	1970.000000	1.000000	0.000000	2.800000	min	2000.00000	1.000000	0.000000	19.300000
25%	2016.000000	7315.000000	125.000000	47.100000	25%	2016.00000	7441.250000	125.000000	47.100000
50%	2017.000000	17259.000000	145.000000	54.300000	50%	2017.00000	17608.000000	145.000000	55.400000
75%	2019.000000	32405.750000	145.000000	62.800000	75%	2019.00000	32765.000000	145.000000	62.800000
max	2020.000000	323000.000000	570.000000	470.800000	max	2020.00000	149000.000000	570.000000	217.300000

engine_capacity		price		engine_capacity	
count	10000.000000	count	10000.000000	count	2000.000000
mean	1.661030	mean	16849.075000	mean	1.645650
std	0.549049	std	9847.186966	std	0.544444
min	0.000000	min	795.000000	min	0.000000
25%	1.200000	25%	10000.000000	25%	1.200000
50%	1.500000	50%	14498.000000	50%	1.500000
75%	2.000000	75%	20900.000000	75%	2.000000
max	6.200000	max	126000.000000	max	5.500000

3. The distribution of the test data and train data is similar but there are many outliers for efficiency and operating hours, particularly for the higher value.
4. A correlation matrix and scatterplots are plotted. They revealed non-linear relationships between features and resale price, indicating the potential benefit of non-linear models.

- The correlation matrix is plotted.



- The distribution of each feature as x axis with the price is plotted in the code.

2.2 Data Splitting

The dataset was divided into an 80% training set and a 20% validation set. A fixed random seed (42) ensured consistency. Additionally, 5-fold cross-validation was used to estimate test performance.

3. Model Development

3.1 Initial Experiments

Before feature engineering, I want to evaluate the performance of different machine learning models. The categorical features are processed by one-hot encoding and numerical features remain the same.

1. SVR with RBF kernel has around 9500 training and testing loss
2. GLM with Poisson distribution: training loss 5000 and testing loss 4500
3. Random forest: training loss 1300 and testing loss 3400
4. CatBoost: training loss 2800 and testing loss 3000

Some ML models perform quite well even before data preprocessing and some may not be suitable for this task, like SVR.

After I get the basic idea of the data, I try to remove some features that are not seems to related to the price (model and other categorical features) and do normalization. I tested random forest, AdaBoost, XGBoost, and Ridge regression with kernel which we have done in homework2, but the training and validation RMSE is fluctuated around 2700 while the test RMSE is around 2900. The best performance is achieved by Ridge regression with 2500 validation RMSE and 2900 testing RMSE. This indicate that the models have a high bias due to the remove of “model” and testing performance is affected by unremoved outliers.

3.2 Feature Engineering

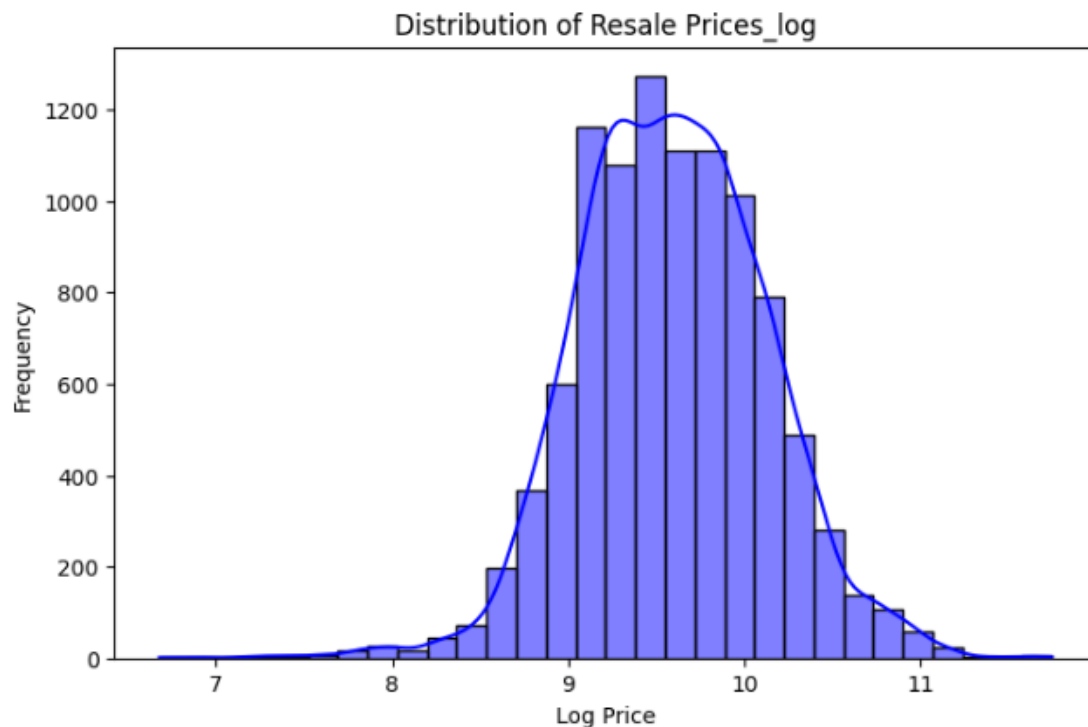
Feature engineering is crucial in improving model performance. In part 2.2 and 2.3, I have already explored the data distribution and some important interactive terms were found.

1. Registration fees and operating hours
2. Year and operating hours
3. Operating hours**2
4. Efficiency capacity

Interactive terms with all combination using Polynomial Features functions are also tested and the above selected terms have a better result.

Log transform

In part 2.1, a left skewed distribution was observed for the response variable price. To transform the skewed distribution to normal, log transform is applied to the price. The transformed distribution is shown below:



This allows the model to have better prediction on the price.

From the previous session, we have seen that the outliers are the key difference between the training and the testing data. I have tried different outliers handling method including isolation forest, IQR and Z-score. Within these three techniques, isolation forest and Z-score have similar performance but for the efficiency concern, I have chosen Z-score as the final method. Data point with Z-score greater than 4.2 are removed from the training set.

After the above feature engineering, one-hot encoding and standard scaling were performed for the categorical and numerical features. The data split is still the same as in 2.2.

3.4 Model and hyperparameters

From the previous part, I have tested some tree-based models, GLM, SVR. In addition, KNN and LDA are also tested to have a high bias. MLP with different layers are tested but the performance is still close to tree-based models. Among all the machine learning and deep learning models, only tree-based models demonstrated overfitting with relatively fast running time. The other models are either slow or have

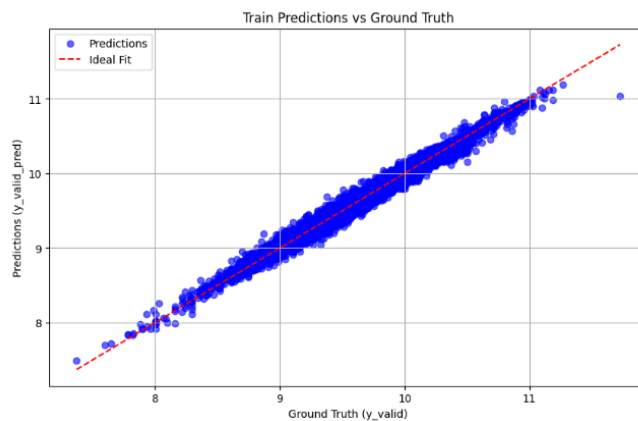
a high bias even after hyperparameter tuning by GridSearchCV meaning their model is not flexible enough. To facilitate model hyperparameter tuning and testing, I have chosen a faster model, XGBoost as the final model. GridSearchCV was performed with the parameters `n_estimators`, `learning_rate`, `max_depth`, `subsample`, `colsample_bytree`, `reg_lambda`. Some are to make sure a low bias and some are regularizations.

4. Result

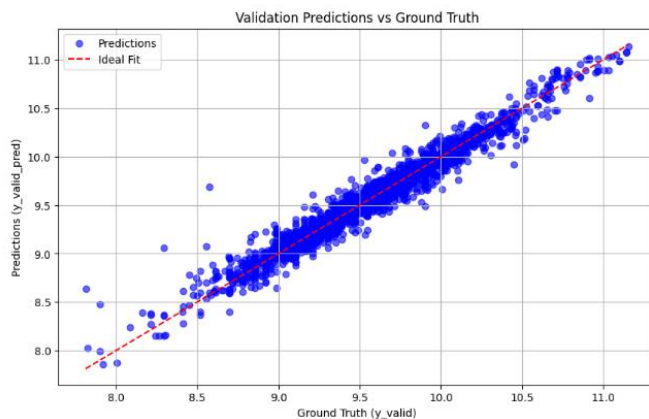
The final parameter is:

```
# Define Model and Pipeline
xgb_model = XGBRegressor(
    random_state=42,
    n_estimators=600,
    learning_rate=0.05,
    max_depth=7,
    subsample=0.8,
    colsample_bytree=0.8,
    reg_lambda=1 # Use reg_lambda instead of lambda_
)
model_pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', xgb_model)
])
```

The final prediction results are:




Training RMSE:1451



Validation RMSE: 1968

The public and private leaderboard testing loss are:

Public:

45	tianyu209		2203.1	50	7h
----	-----------	---	--------	----	----

Private (From the second released ranking):

9 tianyu209

2257.4

The losses show a slightly overfit on the training data, however, based on the result on the public and private leaderboard, the model seems to be a good fit for the overall testing data.

5. Discussion

From 2.1, the Statistical Summary of the training and testing look like the same, except the extreme outliers from operating hours and efficiency. However, the huge difference between the train/validate loss and the actual two testing loss suggested that there is some variant between their distribution. The poor performance of SVR and other linear model (especially when “model” is in the training set) suggested that including the feature “model” cause the data highly non linearly separable. In conclusion, the machine learning models are efficient to have a good estimation of the data distribution. For some complex data like this project, tree-based models have a better prediction.

6. Conclusion

This project demonstrated the utility of systematic feature engineering and model tuning in predicting resale prices of agricultural equipment. Despite the challenges posed by outliers and non-linear feature relationships, the final model achieved

competitive RMSE scores on both validation and leaderboard tests. Overall, the project highlighted the importance of iterative refinement, model analysis, and the strategic use of feature engineering in developing robust machine learning solutions.