ECE219 Large-Scale Data Mining: Models and Algorithms **(Due: Fri, 03/06/20)**

Project 4: Regression Analysis

Instructor: Vwani Roychowdhury Zhaoxi Yu(005432230) & Yuhao Yin(104880239)

1 Introduction

Regression analysis is a statistical procedure for estimating the relationship between a target variable and a set of describing features. In this project, we explore common practices for best performance of regression. We will conduct different experiments and identify the significance of practices that suggested below.

2 Datasets

You should take steps in Section 3 on the following datasets.

2.1 Bike Sharing Dataset

Bike sharing dataset provides count number of rental bikes based on some timing and environmental conditions. You can find feature descriptions on the website.

We will perform data analysis based on three different labels which are:

- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

We will do 3.1.1 data inspection for all three targets and continue project with total count (third item).

2.2 Video Transcoding Time Dataset

Video transcoding time dataset includes input and output video characteristics along with their time taken for different valid transcodings. The dataset has 68784 data points in total, and each instance has 19 features as described below:

- duration: duration of video in second;
- codec: coding standard used for the input video (e.g. `flv, h264, mpeg4, vp8`);
- height, width: height and width of video in pixels;
- bitrate: bits that are conveyed or processed per unit of time for input video;
- framerate: input video frame rate(fps);
- i: number of i frames in the video, where i frames are the least compressible but don't require other video frames to decode;
- p: number of p frames in the video, where p frames can use data from previous frames to decompress and are more compressible than I-frames;
- b: number of b frames in the video, where b frames can use both previous and forward frames for data reference to get the highest amount of data compression;
- frames: number of frames in video;
- i-size, p-size, b-size: total size in byte of i, p, b frames;
- size: total size of video in byte;
- o-codec: output codec used for transcoding;
- o-bitrate: output bitrate used for transcoding;
- o-framerate: output framerate used for transcoding;

- o-width, o-height: output width and height in pixel used for transcoding.

There are two files in the downloaded folder. Only use `transcoding-mesurment.tsv` for your project. Please notice that this file contains 19 features above as well as following two attributes:

- umem: total codec allocated memory for transcoding;
- utime: total transcoding time for transcoding.

Note that the target variable is transcoding time, which is the last attribute "utime" in the data file.

3 Required Steps

In this section, we describe the setup you need to follow. Take these steps on the datasets in section 2. (Take whichever steps that may apply to each dataset.).

3.1 Before Training

Before training an algorithm, it's always essential to inspect data and understand how it looks like. Also, raw data might need some preprocessing. In this section we will address these steps.

3.1.1 Data Inspection

The first step for data analysis is to take a close look at the dataset.

QUESTION 1. Plot a heatmap of Pearson correlation matrix of dataset columns. Report which features have the highest absolute correlation with the target variable and what that implies.

Remark. For bike sharing dataset, the heatmap is shown in Figure 1. Features `temp` and `atemp` have the highest absolute correlation with the target variables `casual` and `cnt`, and feature `yr` has the highest absolute correlation with the target variable `registered`. It demonstrates that the year is most informative in terms of predicting the number of registered users, while the temperature and feeling temperature is most informative to predict the number of casual users and total rental bikes including both casual and registered.

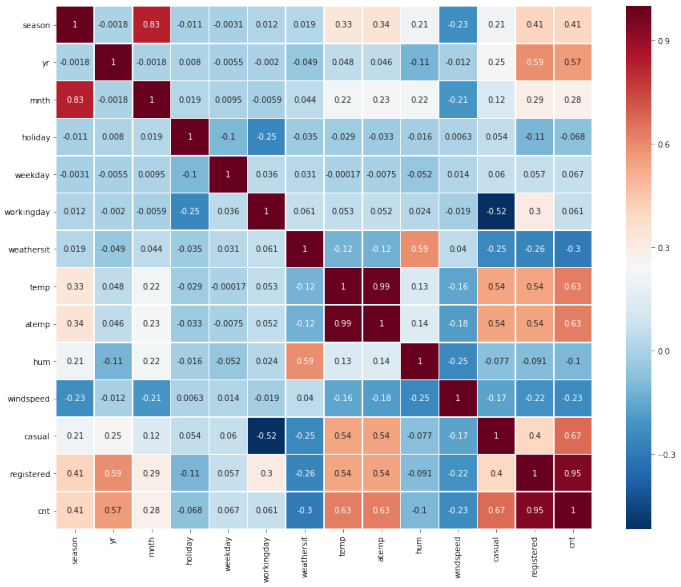

Figure 1: Heatmap for Bike Sharing Dataset

Remark 2. For video transcoding time dataset, the heatmap is shown in Figure 2. Features `o_width` and `o_height` have the highest absolute correlation with the target variable `utime`, which implies output video width and height are most informative in terms of predicting the video transcoding time.

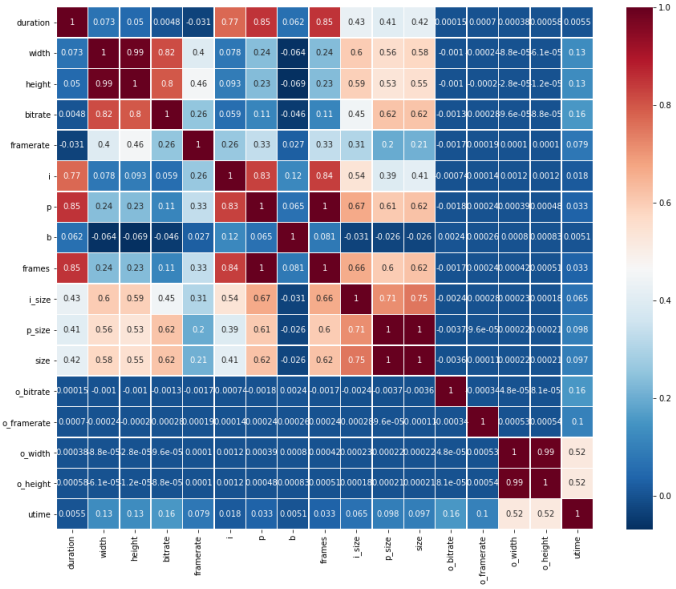

Figure 2: Heatmap for Video Transcoding Time Dataset

QUESTION 2. Plot the histogram of numerical features. What preprocessing can be done if the distribution of a feature has high skewness?

Remark. For bike sharing dataset,histograms of all numerical features are shown in Table 1. Observed from the figures, features `hum` and `windspeed` have high skewness. Preprocessing step like standardization has to be performed on those highly skewed features ahead of model fitting, otherwise extreme values will cause great harm to model robustness.

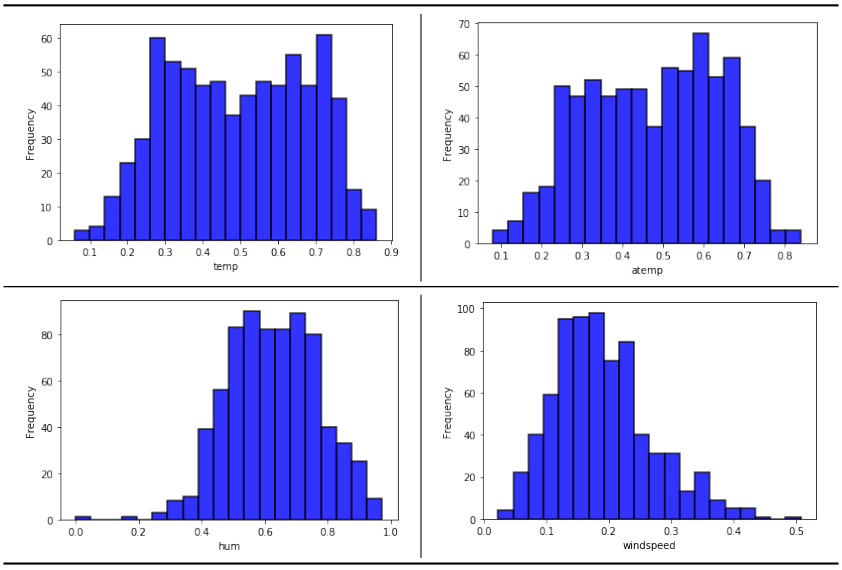

Table 1: Histograms of Numerical Features for Bike Sharing Dataset

Remark 2. For video transcoding time dataset, histograms of all numerical features are shown in Table 2. Obviously, there exists a bunch of numerical features with high skewness, including `duration, bitrate, frames, size`, etc. Preprocessing step like standardization has to be performed on those highly skewed features ahead of model fitting, otherwise extreme values will cause great harm to model robustness.

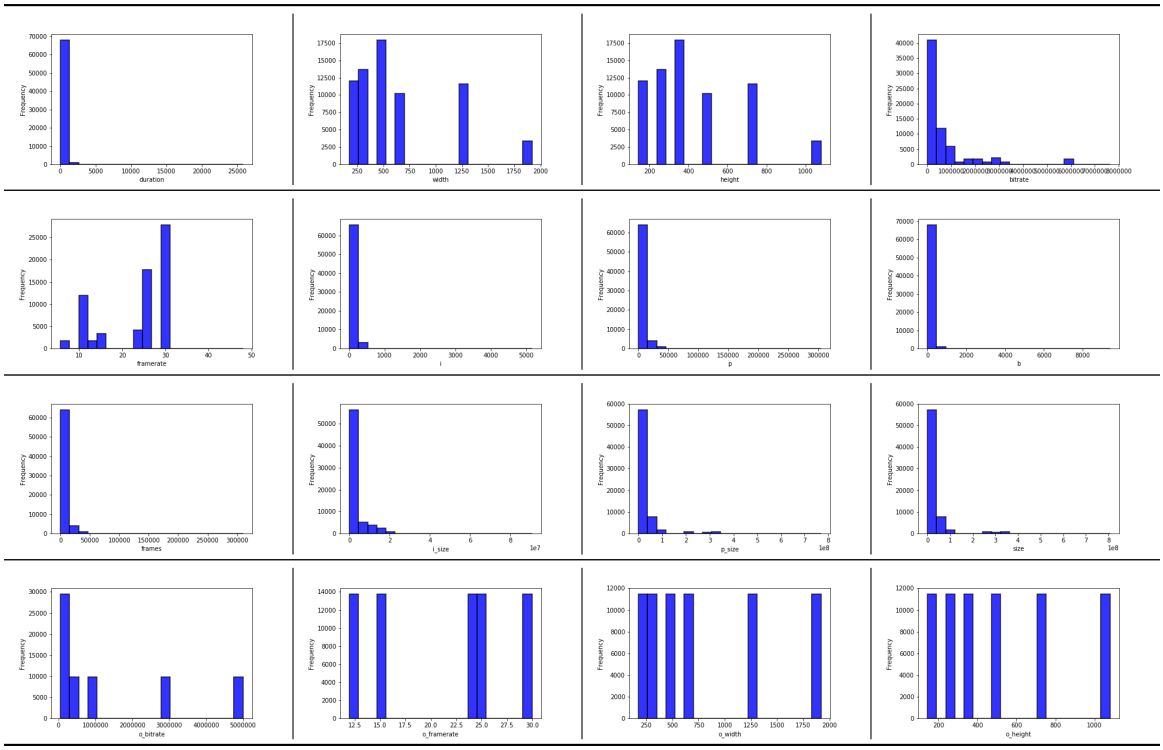

Table 2: Histograms of Numerical Features for Video Transcoding Time Dataset

QUESTION 3. Inspect box plot of categorical features vs target variable. What intuition do you get?

Remark. For bike sharing dataset, there are 7 categorical features, namely `season`, `yr`, `mnth`, `holiday`, `weekday`, `workingday` and `weathersit`. Their respective boxplots are shown in Table 3. We can get the intuition that users are more willing to rent bikes during summer and fall. There are more bike-sharing users in 2012 than in 2011. Casual users are more likely to use bike sharing system in weekend and holidays rather than working days, while registered users tend to use it during working days. Weather condition also has effect on number of users, fewer bike-sharing users in harsher weather.

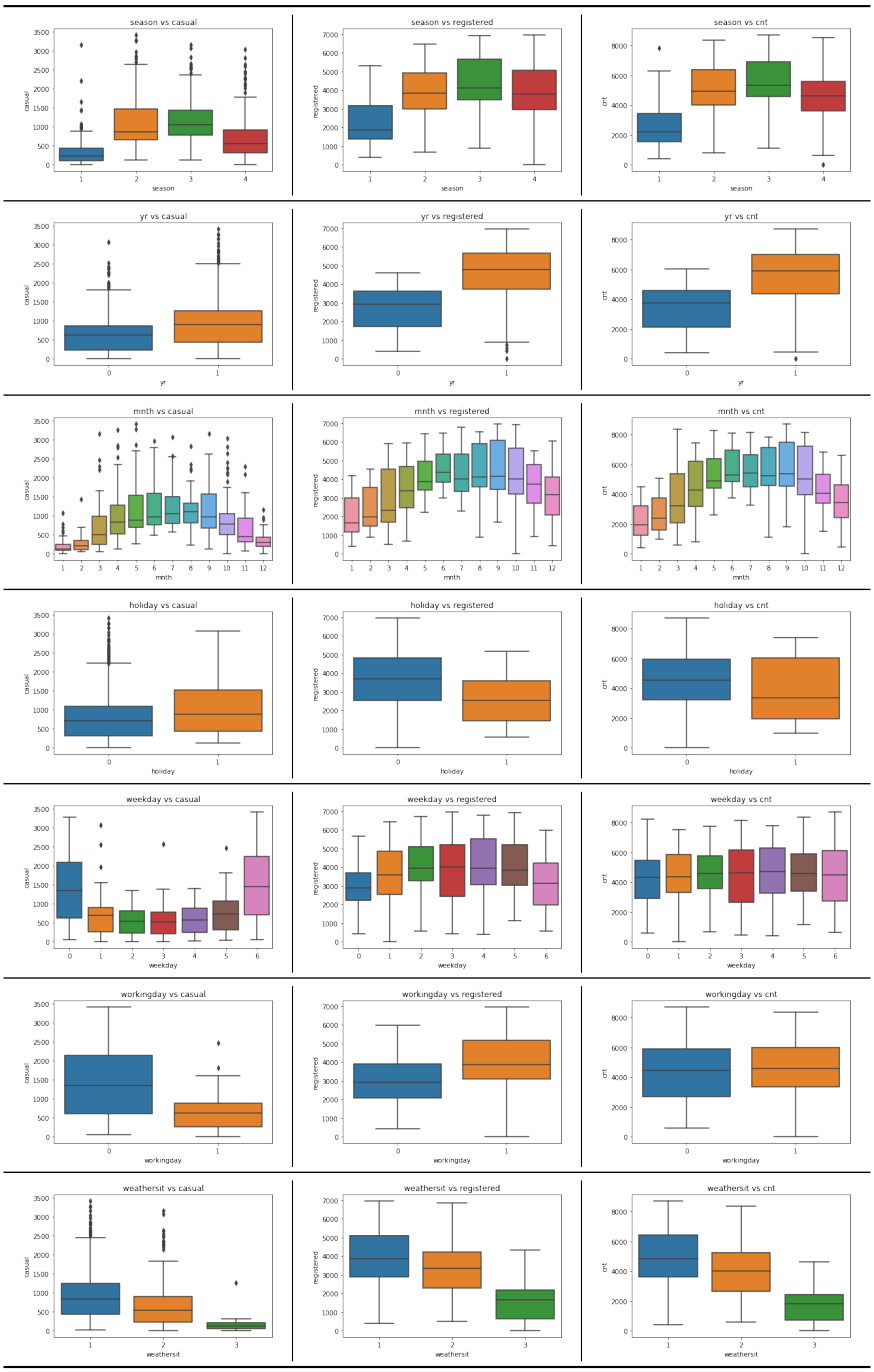

Table 3: Boxplots of Categorical Features for Bike Sharing Dataset

Remark 2. For video transcoding time dataset, there are two categorical features, namely `codec` and `o_codec`. Their respective boxplots are shown in Figure 3. Conclusions derived may include that categorical feature `codec` is not a good predictor for the target `utime`, since corresponding boxplots for four different categorical levels within share no significant differences. While for the other variable, `o_codec`, boxplots for levels `mpeg4` and `h264` are greatly different from the other two, indicating this feature can be effective in terms of predicting the target.

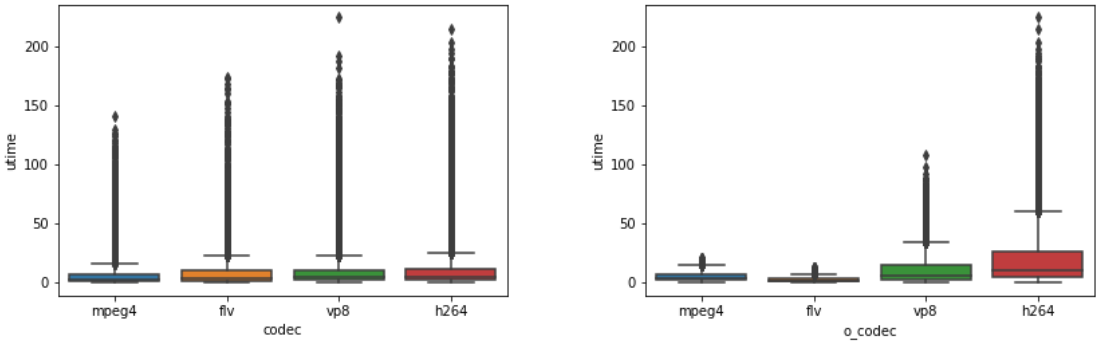

Figure 3: Boxplots of Categorical Features for Video Transcoding Time Dataset

QUESTION 4. For bike sharing dataset, plot the count number per day for a few months. Can you identify any repeating patterns in every month?

Remark. The bar charts of count number per day for first six months are shown in Table 4. The number of bike-sharing users is significant around the 20th of each month.

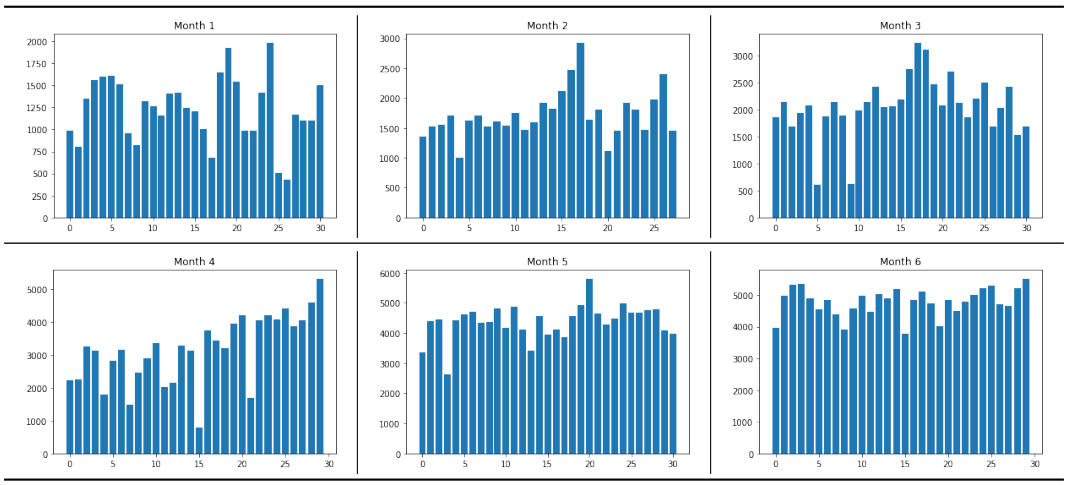

Table 4: Bar Charts of Count Number per Day for Bike Sharing Dataset

QUESTION 5. For video transcoding time dataset, plot the distribution of video transcoding times, what can you observe? Report mean and median transcoding times.

Remark. The distribution of video transcoding times is shown in Figure 4. From there, we observe that the distribution is skewed to the left, indicating there exists few training data whose video transcoding times is extremely large. This conclusion can also be justified by comparing the mean and median statistics of transcoding times, which are **9.9964** and **4.4080** respectively.

3.1.2 Handling Categorical Features

A categorical features is a feature that can take on one of a limited number of possible values. A preprocessing step is to convert categorical variables into numbers and thus prepared for training.

One method for numerical encoding of categorical features is to assign a scalar. For instance, if we have a "Quality" feature with values {`Poor, Fair, Typical, Good, Excellent`} we might replace them with numbers 1 through 5. If there is no numerical meaning behind categorical features (e.g. {`Cat, Dog`}) one has to perform "one-hot encoding" instead.

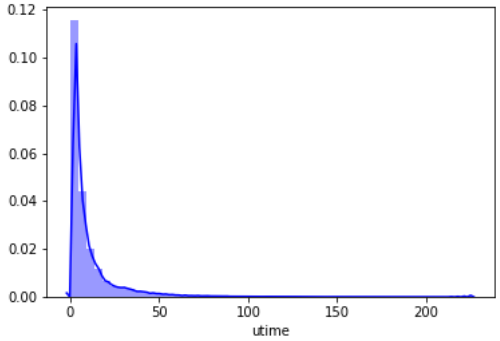

Figure 4: Distribution of Target for Video Transcoding Time Dataset

QUESTION 6. For some other cases, e.g. when encoding time stamps such as {`Mon, ..., Sun`} or {`Jan, ..., Dec`} it might make sense to perform either one. In those cases the learning algorithm of choice and numerical results can lead our way. Can you explain a trade-off here? (Hint: let us assume we perform linear regression, what information does one-hot encoding discard, and what assumption should hold strongly if we perform the scalar encoding instead?)

Remark. For one-hot encoding, we implicitly make the assumption that different categorical levels share the same weight, and thus discard ordering information between levels. Therefore, if the target possesses some seasonality patterns, one-hot encoding can be a good choice.

Remark 2. On the other hand, for the scalar encoding, there must exist ordering relationship between different categorical levels, and furthermore, distances between two consecutive levels should be almost identical as well.

3.1.3 Standardization

Standardization of datasets is a common requirement for many machine learning estimators; they might behave badly if the individual features do not more-or-less look like standard normally distributed data: Gaussian with zero mean and unit variance. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

QUESTION 7. Standardize feature columns and prepare them for training.

Remark. Note that when performing standardization, we ought to standardize the training data only, and apply the calculated mean and standard deviation from training data to normalize validation dataset or incoming test dataset. The logic here is that we always assume the validation dataset is not available until model fitting process is finished. Therefore, when it comes to calculating statistics including both mean and standard deviation, validation dataset shouldn't contribute to it.

Remark 2. For linear regression including Ridge and Lasso, we will compare the model performances with and without standardization preprocessing step. For polynomial regression, neural networks, as well as random forest, we will always standardize all input features before fitting respective models, in order to stabilize the model training.

3.1.4 Feature Selection

- `sklearn.feature_selection.mutual_info_regression` function returns estimated mutual information between each feature and the label. Mutual information (MI) between two random variables is a non-negative value which measures the depen- dency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency.
- `sklearn.feature_selection.f_regression` function provides F scores, which is a way of comparing the significance of the improvement of a model, with respect to the addition of new variables.

QUESTION 8. You may use these functions to select most important features. How does this step affect the performance of your models in terms of test RMSE?

Remark. Especially for linear models, selecting a portion of input features sometimes will boost testing model performance. This can be interpreted as a way of avoiding overfitting problems by getting rid of some less relevant features with respect to the target.

3.2 Training

Once the data is prepared, we would like to train multiple algorithms and compare their performance using both training and testing RMSE based on cross-validation.

3.2.1 Linear Regression

What is the objective function? Train ordinary least squares (linear regression without regularization), as well as Lasso and Ridge regression, and compare their performances. Answer the following questions.

QUESTION 9. Explain how each regularization scheme affects the learned hypotheses.

Remark. The objective functions for ordinary least squares, Lasso and Ridge can be formulated respectively as,

$$\begin{aligned} \text{OLS}: &\min_{\beta} \ \|Y - X\beta\|_2^2 \\ \text{Lasso}: &\min_{\beta} \ \|Y - X\beta\|_2^2 + \lambda\|\beta\|_1 \\ \text{Ridge}: &\min_{\beta} \ \|Y - X\beta\|_2^2 + \lambda\|\beta\|_2 \end{aligned} \tag{1}$$

where $\|\cdot\|_1, \|\cdot\|_2$ represent L_1 and L_2 norm of the vector respectively, and λ is the penalty term for regularization.

Remark 2. For L_1 regularization, the learned hypotheses is that only a fraction of features are active in the linear model. Hence, the L_1 regularization term is added for screening purpose. While for L_2 regularization, the learned hypotheses is that all features are active but overfitting the training data. Hence, L_2 regularization term serves for shrinkage purpose.

QUESTION 10. Report your choice of the best regularization scheme along with the optimal penalty parameter and briefly explain how it can be computed.

Remark. For bike sharing dataset, we manually choose five most significant features based on both mutual information and F scores criteria, which are `atemp`, `temp`,`yr`, `season_1` and `mnth_1`. The first two features possess the highest absolute correlation with target `cnt`, as is reported in Question 1.

In order to search for the best regularization scheme systematically, we construct pipelines and perform grid search over $\{10^x | x = -3, -2, \cdots, 2, 3\}$ in order to optimize the regularization term for both Lasso and Ridge regression, and report the optimal linear model that achieves the least testing RMSE among all those candidate models.

The final results are summarized in Table 5. For simplicity, we only report the top 10 linear models with highest negative root mean squared error.

Remark 2. For video transcoding time dataset, we manually choose five most significant features based on both mutual information and F scores criteria and compare corresponding model performances using 10-fold cross-validation.

In order to search for the best regularization scheme systematically, we construct pipelines and perform grid search over $\{10^x | x = -3, -2, \cdots, 2, 3\}$ in order to optimize the regularization term for both Lasso and Ridge regression, and report the optimal linear model that achieves the least testing RMSE among all those candidate models.

The final results are summarized in Table 6. For simplicity, we only report the top 10 linear models with highest negative root mean squared error.

As we can tell, both training RMSE and validation RMSE are pretty close to each other for all listed linear models with appropriate regularization. In terms of feature selection methods, `f_regression` achieves universally better performance compared to `mutual_info_regression`. Five features selected

mean test score	mean train score	param_model	param_model__alpha	standardize
-989.841732	-990.567256	Ridge	10	False
-992.981214	-988.288295	Ridge	100	True
-998.532721	-986.356699	Lasso	100	True
-1000.667897	-977.409466	Lasso	10	False
-1002.730614	-975.108501	Ridge	1	False
-1004.473824	-974.447499	Ridge	10	True
-1006.881704	-974.018581	Lasso	10	True
-1009.178698	-973.923033	Lasso	1	False
-1009.859563	-973.968975	Ridge	0.1	False
-1013.215210	-973.694229	Ridge	1	True

Table 5: Top 10 Linear Regression Models with Highest Mean Test Score for Bike Sharing Dataset

mean test score	mean train score	param_model	param_model__alpha	standardize	feature selection
-11.851441	-11.856209	Ridge	100	False	F scores
-11.851460	-11.856118	Lasso	0.01	False	F scores
-11.851514	-11.856085	Ridge	10	False	F scores
-11.851524	-11.856085	Lasso	0.001	False	F scores
-11.851533	-11.856084	Ridge	1	False	F scores
-11.851534	-11.856084	Ridge	1	True	F scores
-11.851535	-11.856096	Ridge	10	True	F scores
-11.851535	-11.856084	Ridge	0.1	False	F scores
-11.851535	-11.856084	Ridge	0.1	True	F scores
-11.851535	-11.856084	Ridge	0.01	False	F scores

Table 6: Top 10 Linear Regression Models with Highest Mean Test Score for Video Transcoding Time Dataset

include `o_width, o_height, o_codec_h264, o_codec_mpeg4, o_bitrate`. The first two features possess the highest absolute correlation with the target, as is reported in Question 1, while the importance of the next two features is discussed in Question 3 through boxplot.

QUESTION 11. Does feature scaling play any role (in the cases with and without regularization)? Justify your answer.

Remark. For linear models without regularization, feature scaling won't make any difference to the model, since normalizing one feature will only incur changes to the corresponding coefficient and intercept, but won't affect the fitted value of target at all. We can easily justify our answer by summarizing the model performances of ordinary linear regression:

mean test score	mean train score	param_model	standardize
-1017.715043	-973.625571	LinearRegression	False
-1017.715043	-973.625571	LinearRegression	True

Table 7: Effect of Standardization on Ordinary Linear Regression for Bike Sharing Dataset

Remark 2. While for linear models with regularization, things become a bit more complicated. Standardizing the features or not will have an influence on the regularization, since normalization will cause changes to the estimated coefficients. Therefore, same penalty term λ means different for raw features and scaled features. This conclusion can also be justified by looking at the model performances of both Lasso and Ridge with different regularization terms:

mean test score	mean train score	param_model	standardize	feature selection
-11.851535	-11.856084	LinearRegression	False	F scores
-11.851535	-11.856084	LinearRegression	True	F scores
-15.879762	-15.903696	LinearRegression	False	Mutual information
-15.879762	-15.903696	LinearRegression	True	Mutual information

Table 8: Effect of Standardization on Ordinary Linear Regression for Video Transcoding Time Dataset

mean test score	mean train score	param_model	param_model__alpha	standardize
-989.841732	-990.567256	Ridge	10	False
-1004.473824	-974.447499	Ridge	10	True
-1238.015626	-1148.483372	Lasso	100	False
-998.532721	-986.356699	Lasso	100	True

Table 9: Effect of Standardization on Regularized Linear Regression for Bike Sharing Dataset

QUESTION 12. Some linear regression packages return p-values for different features. What is the meaning of them and how can you infer the most significant features?

Remark. P-values in the linear regression model measures the probability of feature coefficients being equal to zero. Hence, if the p-value for some feature is very close to 0, we will have the confidence to say that particular feature is significant in the linear model.

Remark 2. We use python package `statsmodels` to compute p-values for all features conveniently. For bike sharing dataset, the most significant feature is `yr`, which is also in top 5 features with highest F scores and mutual information. For video transcoding time dataset, the most significant features with very small p-value include `o_codec_h264, o_framerate, o_bitrate, o_codec_vp8, o_width, bitrate, o_codec_mpeg4`, which overlaps top 5 features with highest F scores.

3.2.2 Polynomial Regression

Perform polynomial regression by crafting products of raw features up to a certain degree and applying linear regression on the compound features. You can use `scikit-learn` library to build such features. Avoid overfitting by proper regularization. Answer the following:

QUESTION 13. Look up for the most salient features and interpret them.

Remark. The most salient features are those with greatest absolute coefficients. For bike sharing dataset, the most salient raw features are `temp` and `atemp` which also have the highest absolute correlation with the target. This conclusion is reasonable, since people are more willing to rent bike in warm days and avoid using bike in freezing days. Thus, the total number of users depends largely on the temperature.

For video transcoding time dataset, the most salient raw features are `o_width` and `o_height`, and those two features also have the highest absolute correaltion with the target. This conclusion is interpretable, since the transcoding time definitely depends on the output video size, which also depends on the output width and height in pixel used for transcoding.

QUESTION 14. What degree of polynomial is best? What reasons would stop us from too much increase of the polynomial degree? How do you choose that?

Remark. For bike sharing dataset, we choose the top 5 features based on F scores and mutual information as our raw features, which is the optimal choice from above linear models. Then, we further perform grid search over polynomial degrees from 1 to 10 and report both train and validation RMSE using 10-fold cross-validation. The result is shown in Figure 5,

mean test score	mean train score	param_model	param_model__alpha	standardize	feature selection
-11.851441	-11.856209	Ridge	100	False	F scores
-11.852156	-11.856813	Ridge	100	True	F scores
-11.851460	-11.856118	Lasso	0.01	False	F scores
-11.852200	-11.856863	Lasso	0.01	True	F scores

Table 10: Effect of Standardization on Regularized Linear Regression for Video Transcoding Time Dataset

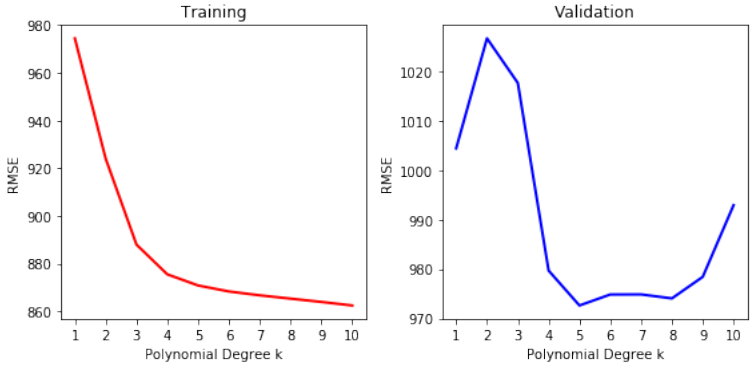

Figure 5: Performances for Polynomial Regression with Different Degrees for Bike Sharing Dataset

The optimal degree of polynomial we select is **5**. From the validation performance we can observed that starting from the polynomial degree of 5, validation RMSE has an increasing trend, which results in overfitting. Thus we should stop increasing the polynomial degree.

Remark 2. For video transcoding time dataset, we choose the top 5 standardized features based on F scores as our raw features, which is the optimal choice from above linear models. Then, we further perform grid search over polynomial degrees from 1 to 10 and report both train and validation RMSE using 10-fold cross-validation. The result is shown in Figure 6,

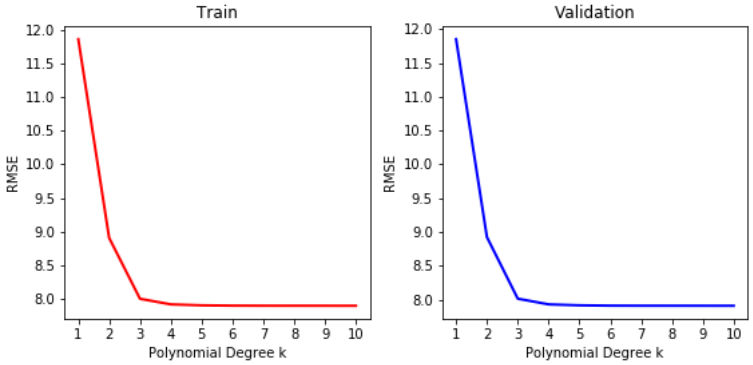

Figure 6: Performances for Polynomial Regression with Different Degrees for Video Transcoding Time Dataset

The optimal degree of polynomial we select is **3**. Although the validation RMSE is decreasing monotonically, there isn't too much change after degree three. Therefore, in order to avoid overfitting and make our polynomial model more efficient and interpretable at the same time, we choose $k = 3$ as the optimal degree, instead of $k = 10$.

QUESTION 15. For the transcoding dataset it might make sense to craft inverse of certain features such that you get features such as $\frac{x_i x_j}{x_k}$, etc. Explain why this might make sense and check if doing so will boost accuracy.

Remark. One inverse transformation we try here is $\frac{\texttt{o\_height} \times \texttt{o\_width}}{\texttt{o\_bitrate}}$. This new created feature is closed

connected to the target, since the numerator `o_height` × `o_width` can be interpreted as the total number of pixels of the output video per frame for transcoding, and taking the ratio of `o_bitrate`, which is the bits processed per unit of time for output video, we can get something proportional to the total transcoding time.

Remark 2. Adding this new feature to the original ordinary linear regression model, we figure out that the validation RMSE has decreased from **11.852156** to **11.709975**. This is a reasonable improvement to the first-order model, considering that merely one feature is added.

3.2.3 Neural Network

Try a multi-layer perceptron (fully connected neural network). You can simply use `sklearn` implementation and compare the performance. Then answer the following:

QUESTION 16. Why does it do much better than linear regression?

Remark. Based on the experimentation we did on video transcoding time dataset, the validation RMSE for neural network models with default hyper-parameter settings is around 5.4, while the validation RMSE for the optimal regularized linear model is around 11.6, indicating neural network is much better than the linear regression.

The main reason is that neural network can capture non-linear relationship between features and target while linear regression model cannot. Furthermore, multi-layer perceptron model can fit very complicated relationship by introducing multiple hidden layers, thus we simply include all the features in this part.

QUESTION 17. Adjust your network size (number of hidden neurons and depth), and weight decay as regularization. Find a good hyper-parameter set systematically.

Remark. In order to find a good hyper-parameter setting, we follow this heuristic method: first off, we start from the neural network with only one hidden layer, and perform a grid search over the number of units in the hidden layer, ranging from 1 to 50, with weight decay penalty term `alpha` holding as default value 0.0001.

After find the optimal number of units for the first hidden layer, we move on to add the second hidden layer, and perform another grid search over the number of hidden units in that layer, ranging from 1 to 50, also with weight decay penalty term `alpha` holding as default value 0.0001.

Last but not least, we fix the neural network structure, which has two hidden layers with their respective optimal number of units derived from aforementioned procedure, and grid search the regularization term over $\{10^x | x = -3, -2, \cdots, 2, 3\}$.

Remark 2. For bike sharing dataset, the optimal number of units for the first layer is **39**, and the optimal number of units for the second layer is **42**. Furthermore, the last grid search result is summarized in the Table 11.

mean_test_score	mean_train_score	param_model__alpha
-831.822559	-420.913948	0.001
-833.707344	-420.012518	0.1
-834.264457	-421.378254	0.01
-837.478613	-420.455306	1
-840.436903	-416.662118	10

Table 11: Grid Search on Regularization Term for Video Transcoding Time Dataset

Hence, the optimal value of L_2 penalty parameter is **0.001** for this two-layer fully-connected neural network.

Remark 3. For video transcoding time dataset, the optimal number of units for the first layer is **16**, and the optimal number of units for the second layer is **40**. Furthermore, the last grid search result is summarized in the Table 12.

mean_test_score	mean_train_score	param_model__alpha
-4.931672	-3.122236	10
-5.165881	-2.123665	0.1
-5.217628	-2.292360	1
-5.390692	-4.970341	100
-5.479199	-2.130981	0.01

Table 12: Grid Search on Regularization Term for Video Transcoding Time Dataset

Hence, the optimal value of L_2 penalty parameter is **10** for this two-layer fully-connected neural network. While penalty term being equal to 0.001, 0.01 or 1 will cause the neural network to overfit the training data.

QUESTION 18. What activation function should be used for the output? You may use none.

Remark. The activation function is none for the output, since here we are performing regression analysis. If it were classification, the activation function should be softmax (or sigmoid) for multi-class (or binary class) classification problems.

QUESTION 19. What reasons would stop us from too much increase of the depth of the network?

Remark. The main reason that stops us from too much increase of the depth of the network is to avoid overfitting problem. Moreover, it is very time-consuming to perform grid search to find a good hyper-parameter settings for some neural network with many hidden layers.

3.2.4 Random Forest

Apply a random forest regression model on datasets, and answer the following.

QUESTION 20. Random forests have the following hyper-parameters:

- Maximum number of features;
- Number of trees;
- Depth of each tree;

Fine-tune your model. Explain how these hyper-parameters affect the overall performance? Do some of them have regularization effect?

Remark. In order to fine-tune the random forest model, we first set the number of trees and maximum depth of each tree as the default value and perform grid search over the maximum number of features, whose proportion ranging from 0.1 to 1. Then, sequentially, we further perform grid search over the number of trees and maximum depth of each tree for candidate number of trees ranging from 10 to 200 and candidate maximum depth of tree ranging from 1 to 30.

Smaller fraction of features considered in constructing the tree and shallower depth of each tree tend to have a higher regularization effect on the random forest model, leading to monotonically increase in training RMSE but optimal validation RMSE somewhere in between.

Remark 2. For bike sharing dataset, the optimal random forest regression model after fine-tuning consists of **160** trees. The maximum number of features considered for each tree is **40%** of all features, and the maximum depth of tree is **22**.

Remark 3. For video transcoding time dataset, the optimal random forest regression model after fine-tuning consists of **190** trees. The maximum number of features considered for each tree is **40%** of all features, and the maximum depth of tree is **26**.

QUESTION 21. Why does random forest perform well?

Remark. Based on the experimentation we did on video transcoding time dataset, the validation RMSE of random forest regression models with default hyper-parameter settings is around 4.0, which is even smaller than that of neural networks.

Reasons for this include bootstrapping the original training dataset and considering only a fraction of features to avoid some dominating feature always appearing at the root of each tree, so as to decorrelate trees ensembled in the random forest. Although one single deep tree may easily overfit the training data, random forest models solve the overfitting problem by introducing aforementioned techniques.

QUESTION 22. Randomly pick a tree in your random forest model (with maximum depth of 4) and plot its structure. Which feature is selected for branching at the root node? What can you infer about the importance of features? Do the important features match what you got in part 3.2.1?

Remark. For bike sharing dataset, we randomly pick one tree from the fine-tuned random forest model, and visualize the tree structure in Figure 7.

Feature branching at the root node is `mnth_1`. Thus, it's safe to conclude that the most significant feature in this randomly picked tree is `mnth_1`. More generally, any feature that appears at the top node of the tree structure tends to be important. Furthermore, top nodes here including `mnth_1, atemp, yr` also match the conclusion derived from the linear regression model.

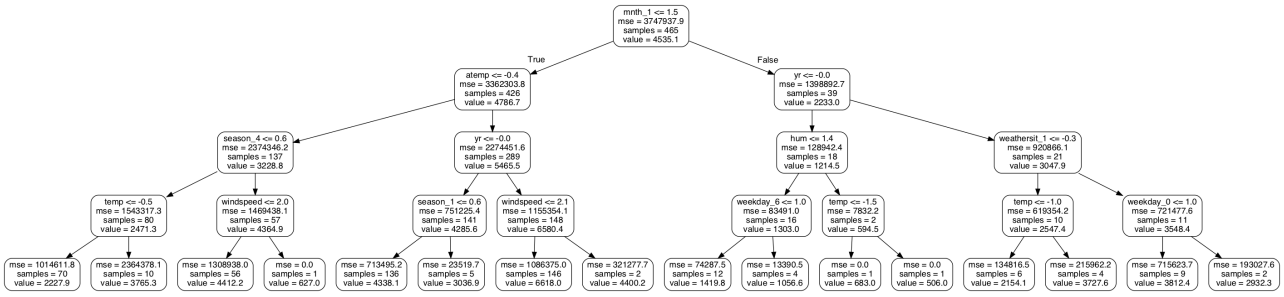

Figure 7: Tree Structure in Random Forest Model for Bike Sharing Dataset

Remark 2. For video transcoding time dataset, we randomly pick one tree from the fine-tuned random forest model, and visualize the tree structure in Figure 8.

Feature branching at the root node is `o_height`. Thus, it's safe to conclude that the most significant feature in this randomly picked tree is `o_height`. More generally, any feature that appears at the top node of the tree structure tends to be important.

Furthermore, top nodes here including `o_height, o_codec_h264, o_codec_mpeg4` also match the conclusion derived from the linear regression model.

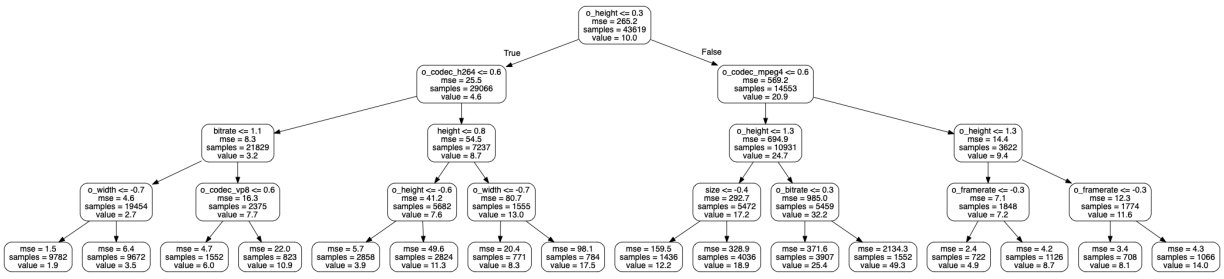

Figure 8: Tree Structure in Random Forest Model for Video Transcoding Time Dataset

3.3 Evaluation

QUESTION 23. Perform 10-fold cross-validation and measure average RMSE errors for training and validation sets. Why is the training RMSE different from that of validation set?

Remark. For all the grid searches, we evaluate both training and validation RMSE as is reported above. These two metrics can be significantly different, since as we increase the model complexity, the training RMSE will decrease monotonically. However, at last we will the point where overfitting problem occurs.

In general, as training RMSE decreasing monotonically, validation RMSE tends to decrease first and then increase. All the fine-tuning procedures we do above is to find the optimal model structure that gives us the minimal validation RMSE.

QUESTION 24. For random forest model, measure "Out-of-Bag Error" (OOB) as well. Explain what OOB error and R2 score means.

Remark. Since we bootstrap the training dataset, for one particular tree, some data point may not be selected for constructing the tree, and the RMSE for that data point is called "Out-of-Bag Error". Therefore, for the random forest regression model, we don't need a separate validation dataset and this OOB error can be served as the validation performance evaluation. On the contrary, R2 provides the coefficient of determination for the trained model on the given training dataset.

Remark 2. For bike sharing dataset, the random forest regression model after fine-tuning possesses a R2 score of **0.9839** and OOB score of **0.8829**.

Remark 3. For video transcoding time dataset, the random forest regression model after fine-tuning possesses a R2 score of **0.9991** and OOB score of **0.9937**. This makes sense, since OOB score can be interpreted as validation R2 score, and should be slightly smaller than R2 score.