Project 4: Regression Analysis

Prof. Vwani Roychowdhury

ECE 219 Winter 2020

UCLA, Department of ECE

Siyuan Peng 805426447 Haonan Huang 605322629
Yinghan Cui 005430212 Yunzheng Zhu 505231998

1. Introduction

In this project, we will conduct different experiments and identify the significance of practices of the different datasets explained in section 2. Based on the regression analysis, we explore common practices for best performance of regression. Sometimes, you can just apply the particular choice and make greedy decisions over the multiple choices.

2. Datasets

The two types of datasets we are implementing:

1. Bike Sharing Dataset (details can be found in the link: Bike Sharing Dataset Data Set)

The analysis is based on three different labels:

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

2. Video Transcoding Time Dataset (details can be found in the link: Video Transcoding Time Data set)

- The dataset has 68784 data points in total, and each instance has 19 features includes: codec, height, width, bitrate, frame rate,i, p, b, frames, i-size, p-size, b-size, o-codec, o-bitrate, o-framerate, o-width, o-height.
- We only use transcoding-mesurment.tsv for our regression analysis. There are two additional attributes: umem, utime. We set utime (total transcoding time for transcoding) as target variable. So in the preprocessing progress we will drop umem.

3. Required Steps

In this project, we are following the steps in pre-training and training with the data. In the pre-training part, we are working on the steps: data inspection, handling categorical features, standardization, and feature selection. In the training part, after the data is prepared, we will compare the performance of multiple algorithms: linear regression, polynomial regression, neural network, and random forest.

Based on the dataset in Section 2, we are splitting each question into two parts: Bike(bike sharing), and Video(video transcoding time).

3.1.1 Data Inspection

Question 1 - Plot a heatmap of Pearson correlation matrix of dataset columns

Bike:

The heatmap of Pearson correlation matrix is shown in Fig. 1 below. Since we are doing the analysis based on the three different labels: ‘casual’, ‘registered’, and ‘cnt’. According to the analysis, the highest absolute correlation values with each target variable are 0.544, 0.594, 0.631. And the corresponding target variables are ‘atemp’(normalized feeling temperature in Celsius), ‘yr’(year), ‘atemp’.

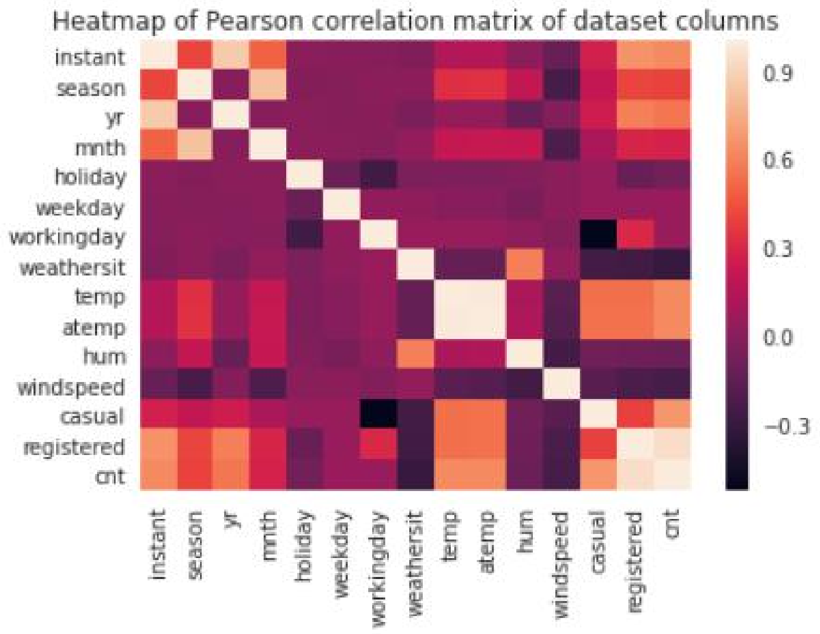


Figure 1. Heatmap of Pearson correlation matrix of Bike Sharing dataset columns. The numbers are ranging from -1(dark) to 1(bright).

According to the values given, we can see that ‘casual’ is highly related to ‘atemp’, which is the normalized feeling temperature in Celsius, which concludes that with the variation of the feeling temperature, the random casual bike renters are highly related. Registered users are highly related with the ‘yr’, the year number, mainly because the registered users are going to increase with the spread of the bike rental information. Lastly, the total count of bikes used is highly related to the feeling temperature as well.

Video:

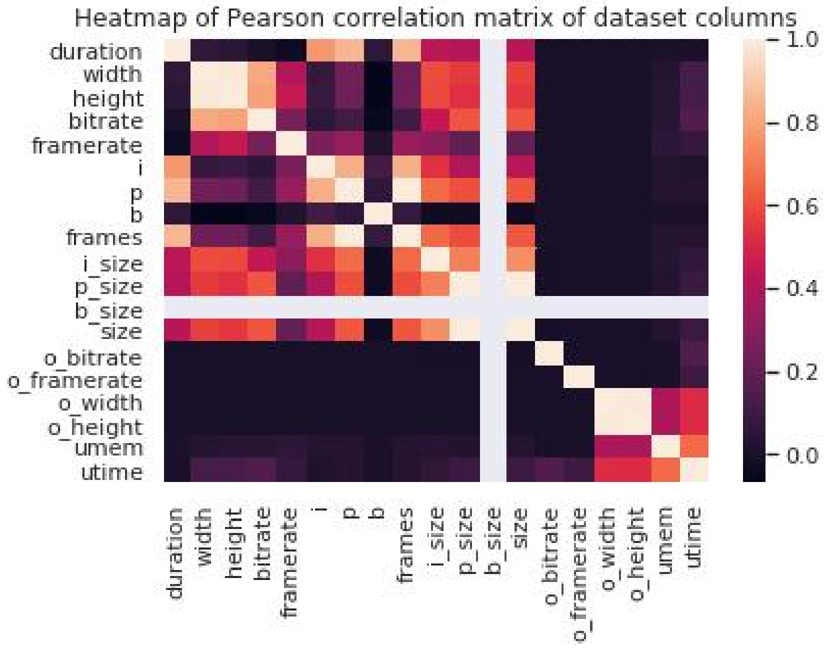


Figure 2. Heatmap of Pearson correlation matrix of Video Transcoding Time dataset columns. The numbers are ranging from -1(dark) to absolute value 1(bright).

As can be seen from the figure, 'o_width' , 'o_height' have the highest absolute correlation with the target variable 'utime' and they are relatively close. We output the highest value, which is 'o_width'. The highest absolute correlation is 0.523387661862717. The correlation of 'o_height' is slightly smaller. This result reveals that the length and width of the output video is highly related to the transcoding time.

Question 2 - Plot a histogram of numerical features

Bike:

Fig. 3 below shows the histogram of numerical features. Here we only apply on the features that are varying with respect to time continuously. The preprocessing can be done is regularization if the distribution of a feature has high skewness.

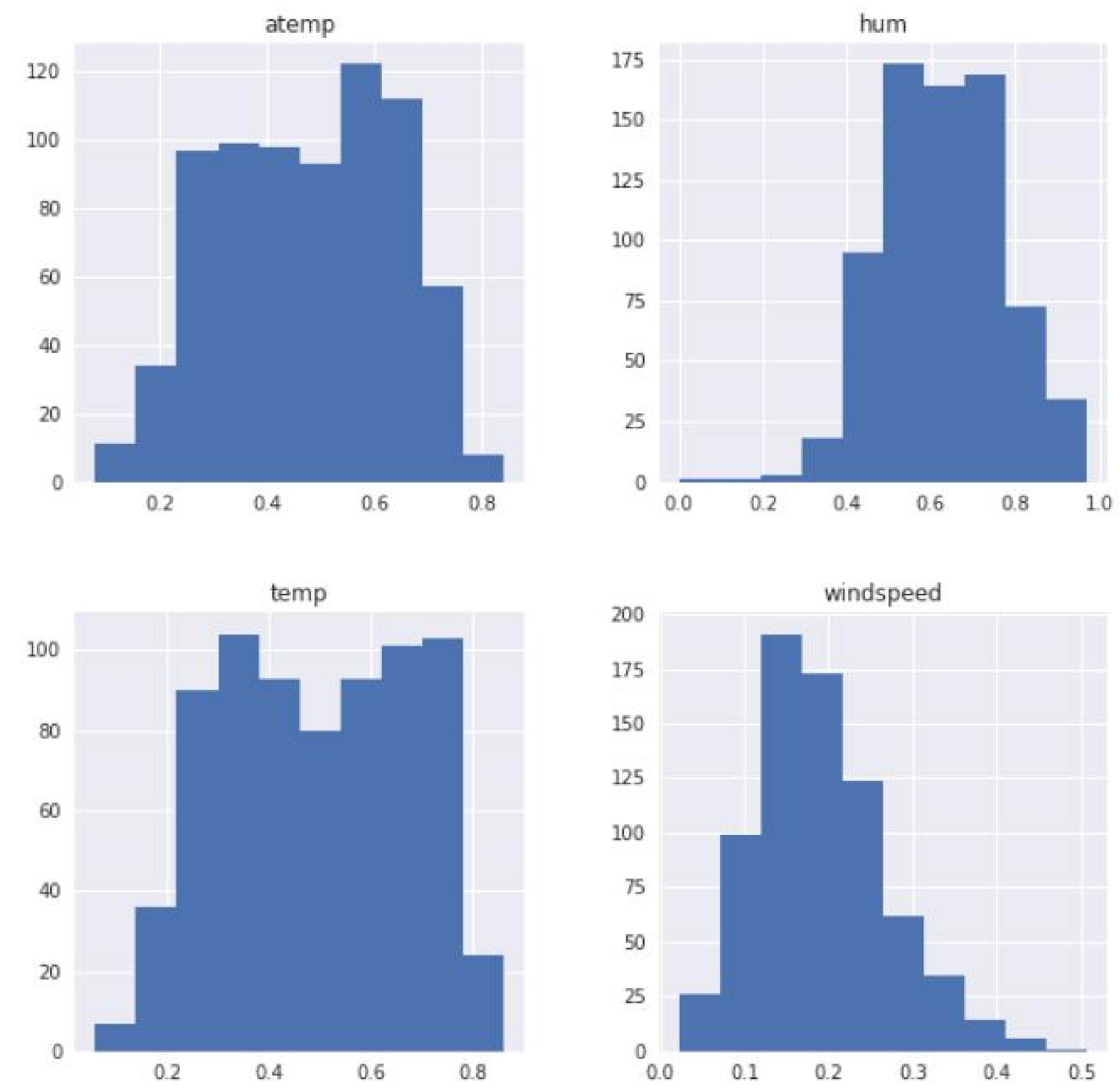


Figure 3. Histogram histogram of numerical features ‘atemp’, ‘hum’(normalized humidity), ‘temp’(normalized temperature in Celsius), ‘windspeed’(normalized wind speed).

Video:

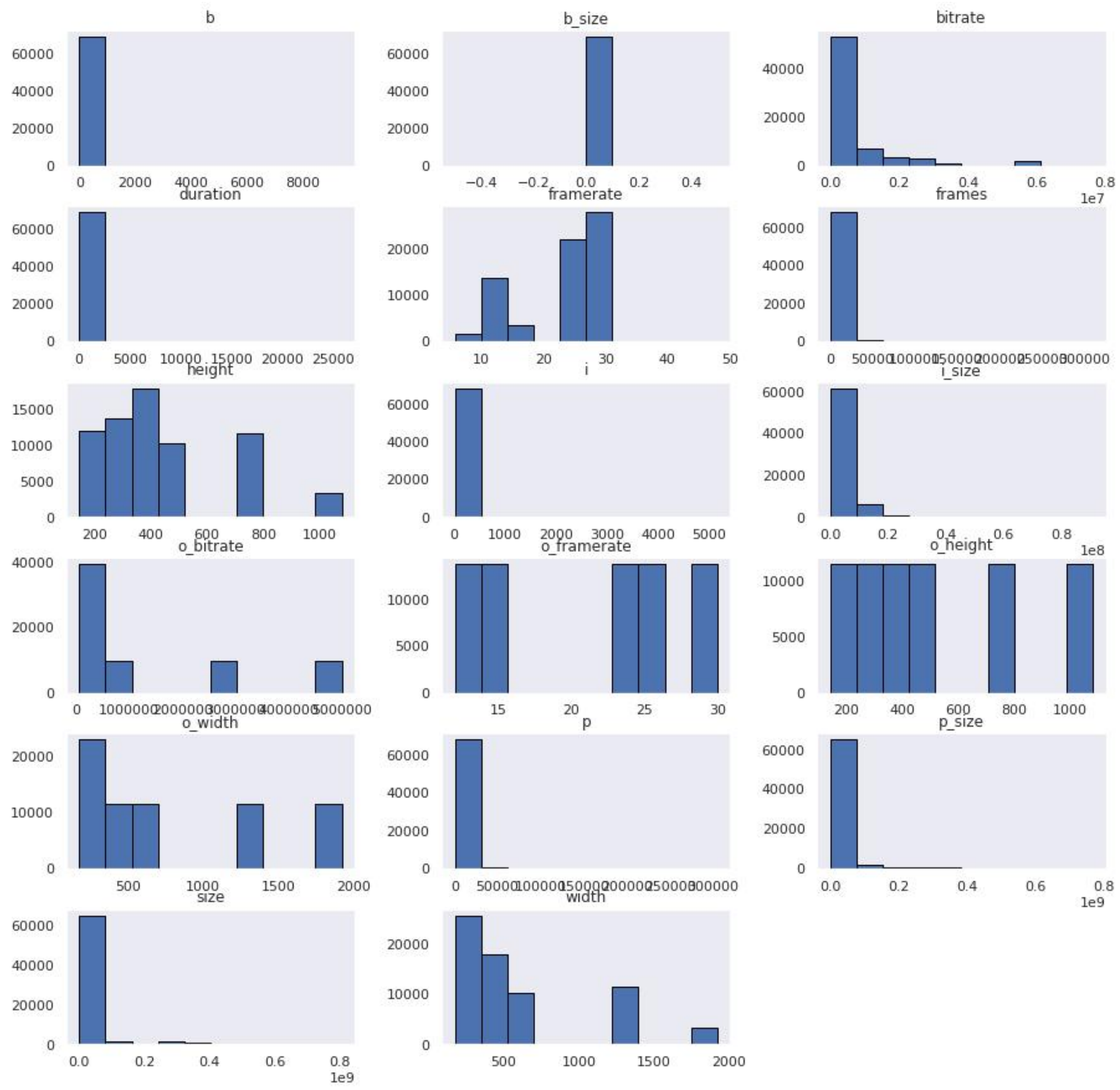


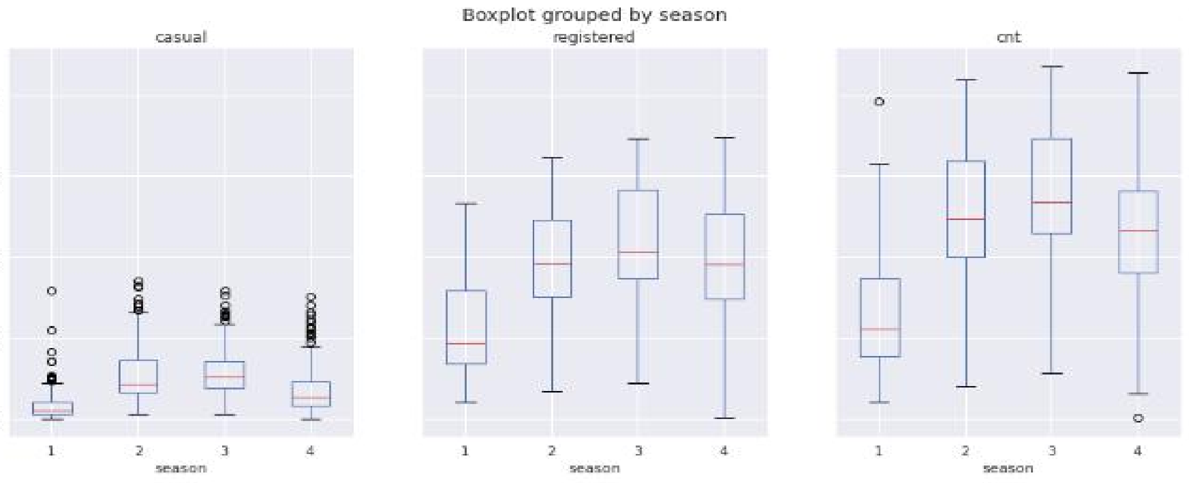
Figure 4. Histogram histogram of numerical features ('duration', 'height', 'width', 'bitrate', 'framerate', 'i', 'p', 'b', 'frames', 'i_size', 'p_size', 'b_size', 'size', 'o_bitrate', 'o_framerate', 'o_width', 'o_height')

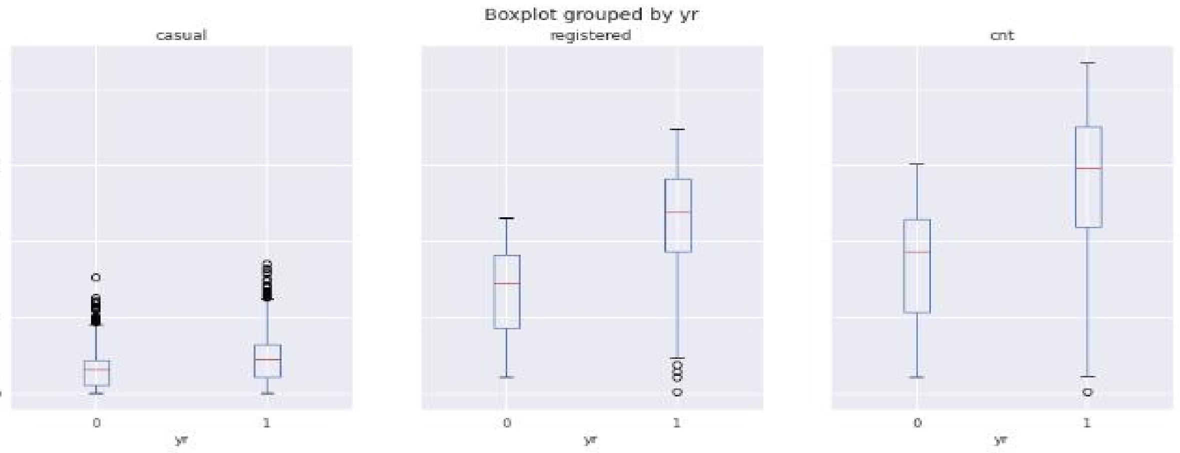
As we can see in the figure, some features have high skewness, such as the ‘bitrate’, we can replace the data with the log, square root, or inverse to remove the skew. Here we use regularization if the distribution of a feature has high skewness.

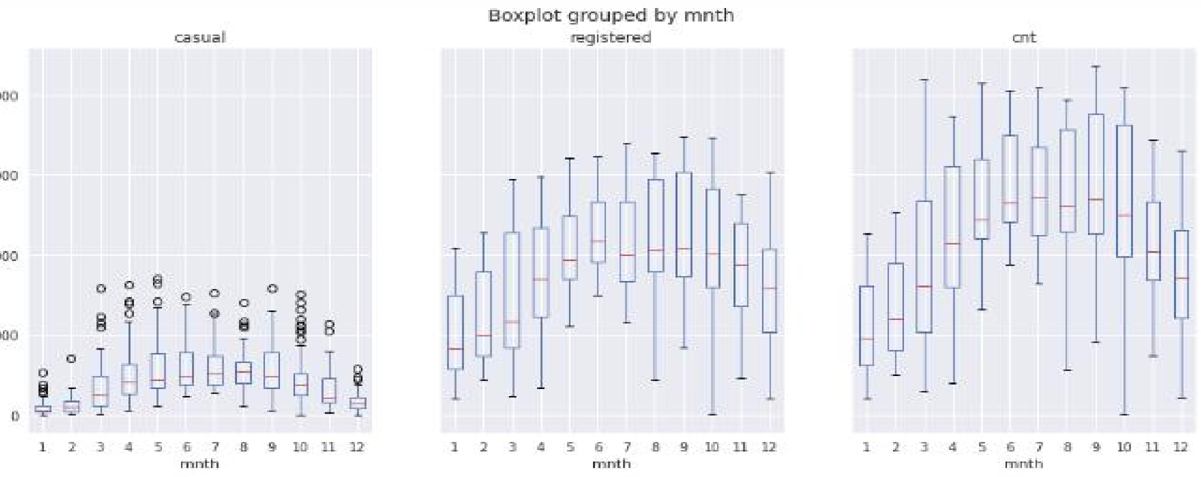
Question 3 - Inspect box plot of categorical features vs. target variable

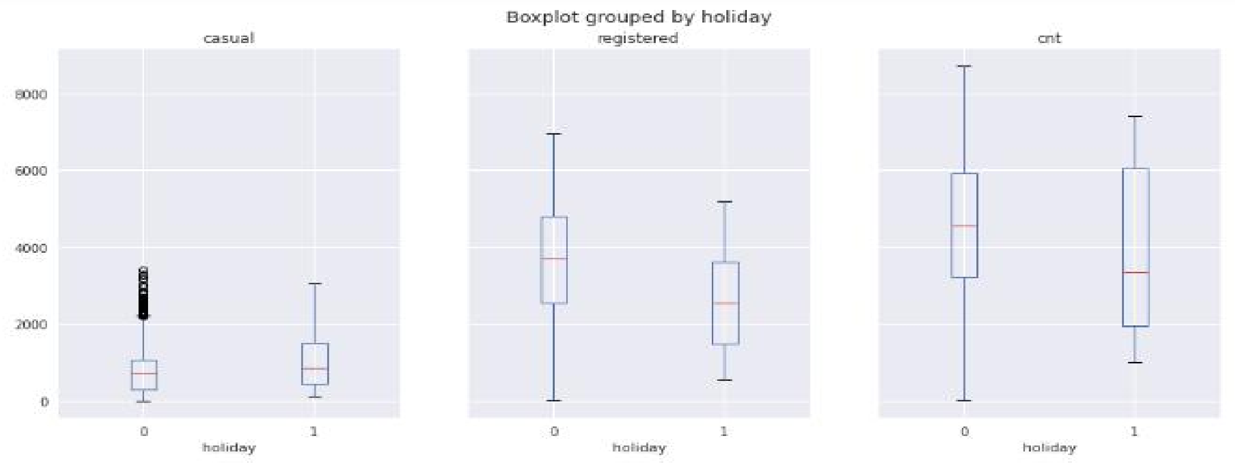
Bike:

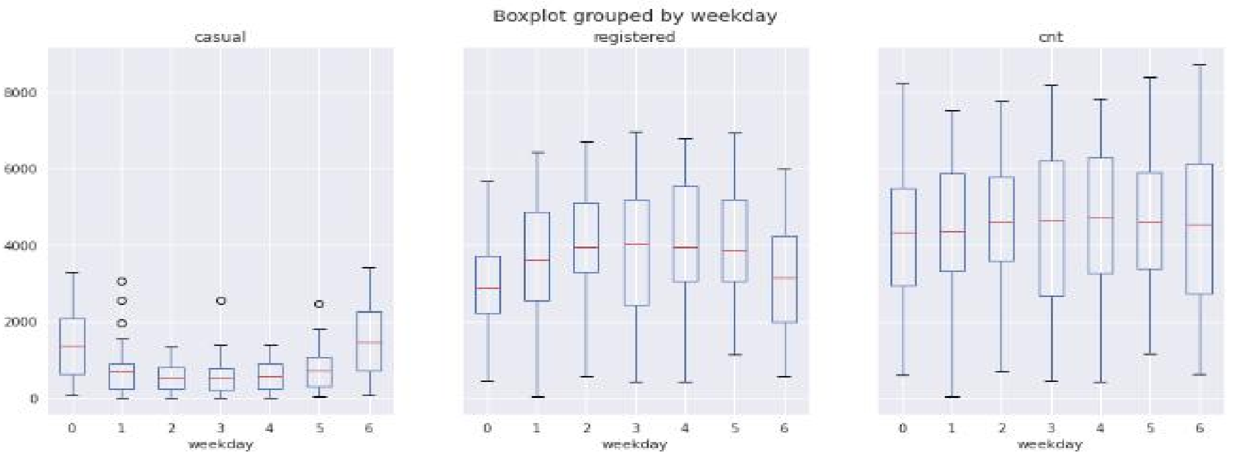
Fig. 5 below shows the box plot of the categorical features (‘season’, ‘yr’, ‘mnth’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’) vs. target variable(‘casual, ‘registered’, ‘cnt’).

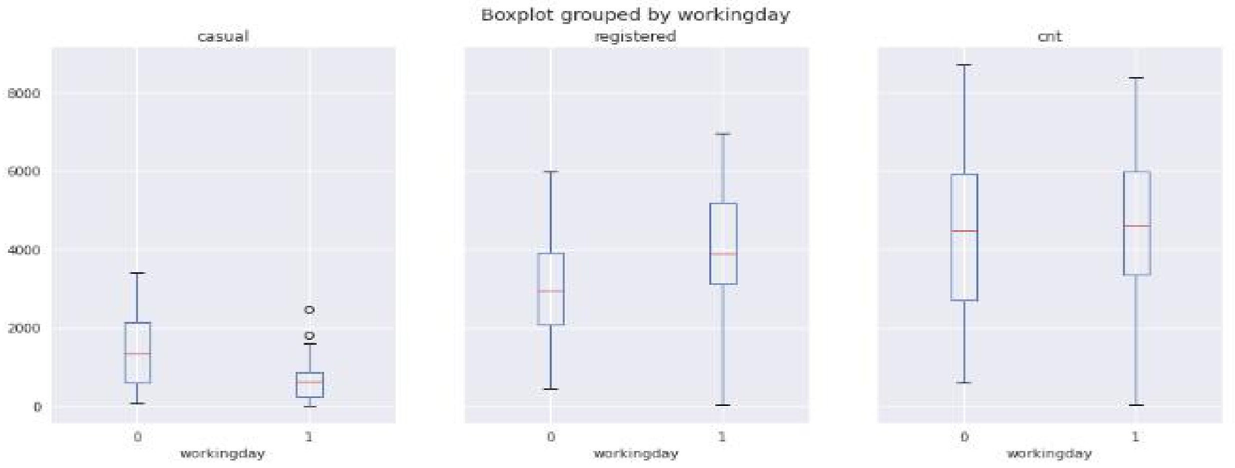

Boxplot grouped by season
casual
registered
cnt
0
2000
4000
6000
8000
1
2
3
4
season


Boxplot grouped by yr
casual
registered
cnt
0
2000
4000
6000
8000
0
1
yr


Boxplot grouped by mnth
casual
registered
cnt
0
2000
4000
6000
8000
1
2
3
4
5
6
7
8
9
10
11
12
mnth


Boxplot grouped by holiday
casual
registered
cnt
0
2000
4000
6000
8000
0
1
holiday


Boxplot grouped by weekday
casual
registered
cnt
0
2000
4000
6000
8000
0
1
2
3
4
5
6
weekday


Boxplot grouped by workingday
casual
registered
cnt
0
2000
4000
6000
8000
0
1
workingday

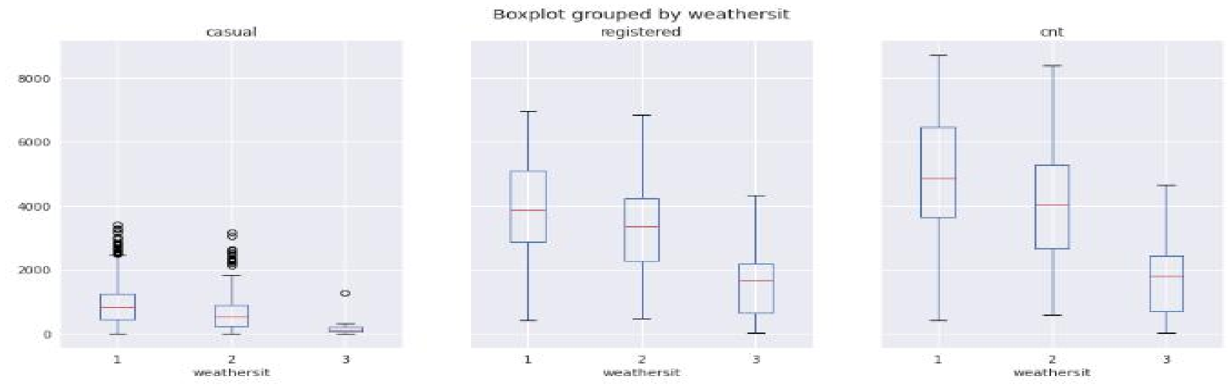


Figure 5. Box plot of the categorical features (‘season’, ‘yr’, ‘mnth’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’) vs. target variable(‘casual, ‘registered’, ‘cnt’).

By observation, the intuition we can get is that for the casual users, there are many outliers in all the categorical features options, which means it does not have too much relation. However, each categorical feature is highly related to the registered users. Total count of users are likely to be the combination of casual and registered users. Thus, the outliers are few and it has decent relation to each of the categorical features.

Video:

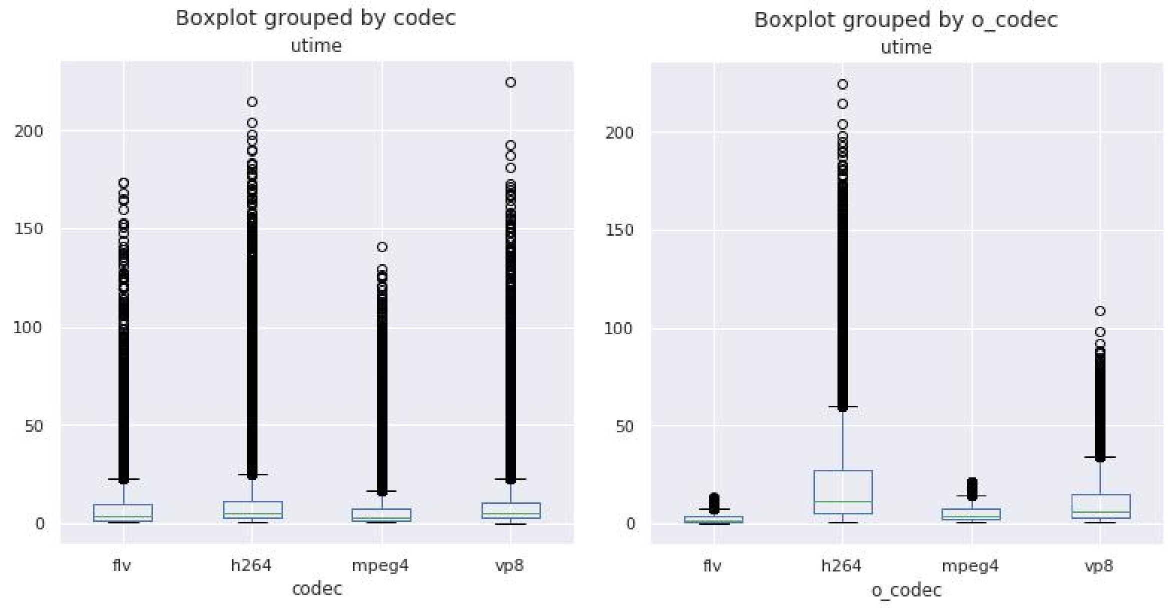


Figure 6. Box plot of the categorical features (‘codec’, ‘o_codec’) vs. target variable(‘utime’).

The figure reveals an interesting intuition, ‘utime’ has more related to the output codec used for transcoding (‘o_codec’) than the coding standard used for the input video (‘codec’). There are many outliers for all the coding standards used for the input video (‘codec’), and the quartiles of different coding standards are relatively close which shows that the target value is less ‘sensitive’ to the input video coding standard. However, the quartiles of different output video coding (‘o_codec’) vary greatly. In addition, the ‘flv’ and ‘mpeg4’ have fewer and more concentrated outliers than those of ‘h264’ and ‘vp8’, which shows the target value is more ‘sensitive’ to the output video coding standard.

Question 4 - Plot the count number per day for a few months (Bike only)

Fig. 7 below shows the count number per day for a few months. Each month is starting with a low count number, and it increases in the following days and reaches a peak in the second half(between day 15 to day 20) of the current month.

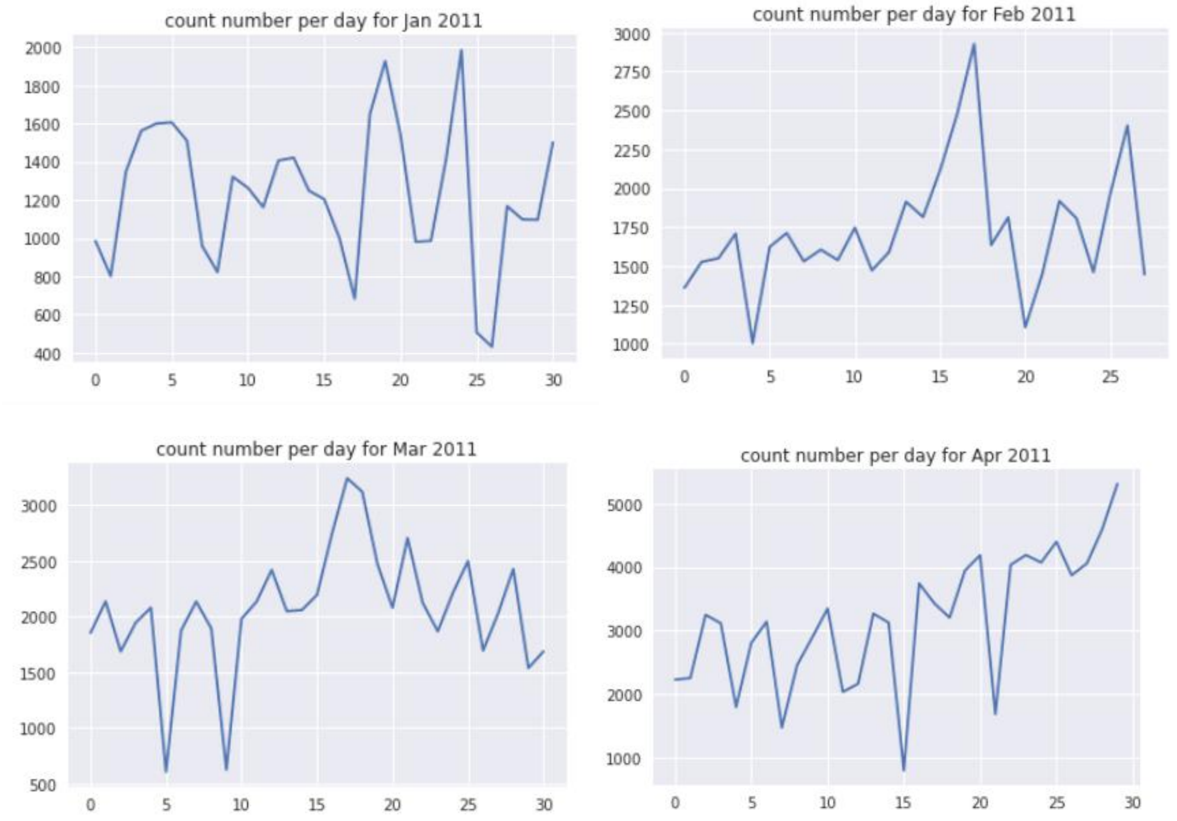


Figure 7. Count number per day for months from Jan. 2011 to Apr. 2011.

Question 5 - Plot the distribution of video transcoding times (Video only)

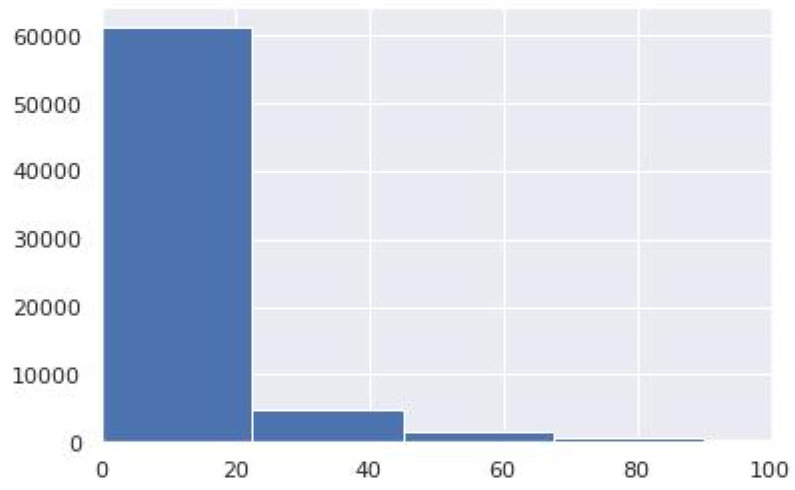


Figure 8. Distribution of video transcoding times.

The mean transcoding time is 9.99635482088516

The median transcoding time is 4.408.

We can see the transcoding time is concentrated between 0-20 and since the mean transcoding time is larger than the median transcoding time, the distribution is positively-skewed (right-skewed).

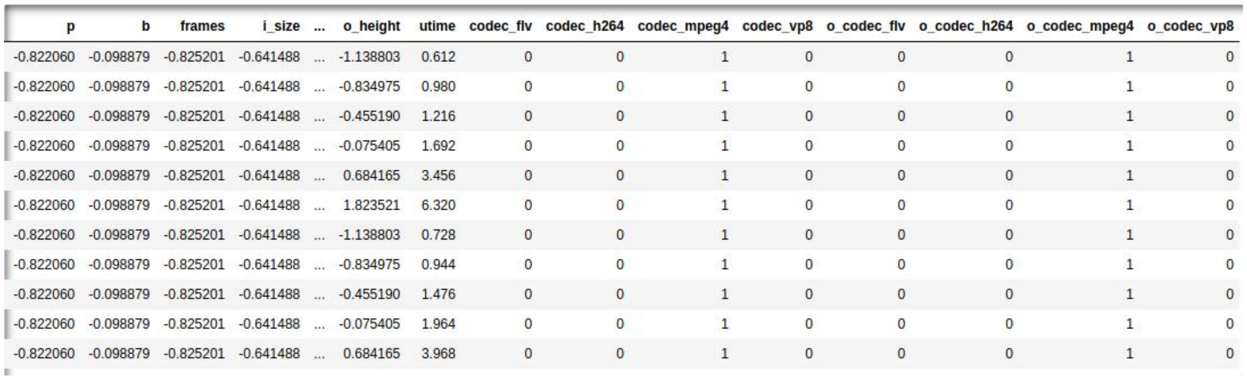
3.1.2 Handling Categorical Features

Question 6 - Handling Categorical Features

Bike:

Since one categorical feature can take on one of a limited number of possible values, here we convert categorical variables into numbers in preprocessing. The method we apply here is to assign a scalar.

Video:



p	b	frames	i_size	...	o_height	utime	codec_flv	codec_h264	codec_mpeg4	codec_vp8	o_codec_flv	o_codec_h264	o_codec_mpeg4	o_codec_vp8
-0.822060	-0.098879	-0.825201	-0.641488	...	-1.138803	0.612	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-0.834975	0.980	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-0.455190	1.216	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-0.075405	1.692	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	0.684165	3.456	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	1.823521	6.320	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-1.138803	0.728	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-0.834975	0.944	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-0.455190	1.476	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	-0.075405	1.964	0	0	1	0	0	0	1	0
-0.822060	-0.098879	-0.825201	-0.641488	...	0.684165	3.968	0	0	1	0	0	0	1	0

Figure 9. Dataset after applying one-hot encoding method.

For video transcoding data set, use one-hot encoding as figure 9 shows, one-hot encoding has the advantages of determining the state has a low and constant cost of accessing one flip-flop, and changing the state. But it also discards the message of the categorical features. In this case, the 'codec' and 'o_codec' are one-hot encoded.

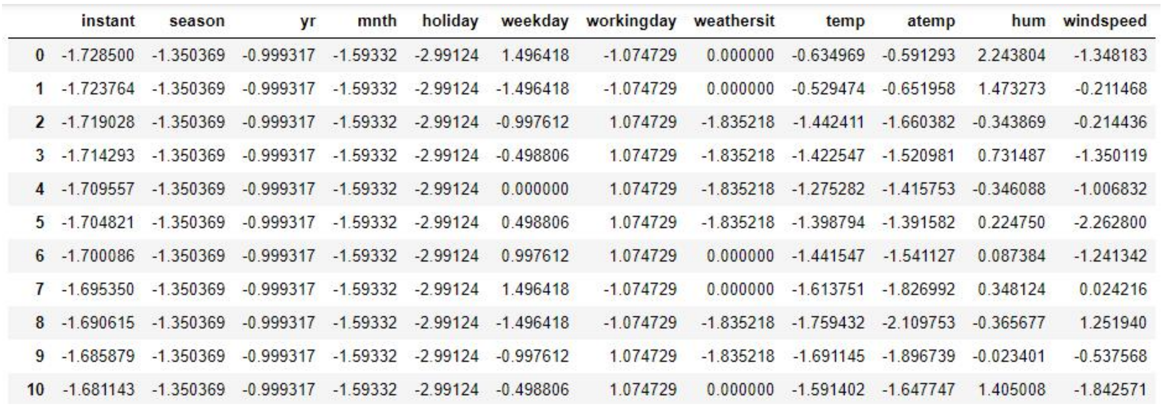
Based on the two different coding methods applying to the two different types of dataset, we can see that scalar coding is better for assigning the data with numerical meanings, though the one-hot coding is more common to the data with no numerical meaning.

3.1.3 Standardization

Question 7 - Standardize feature columns and prepare them for training

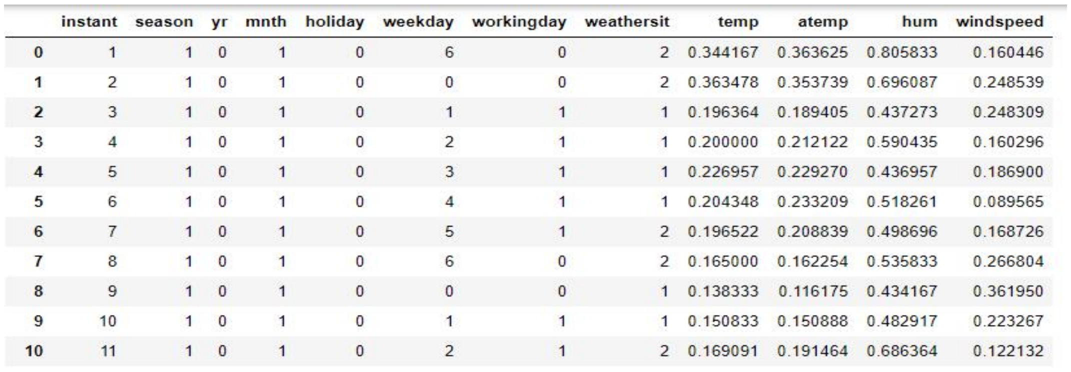
In this part, we are doing the standardization of the datasets, which is common to many machine learning estimators. The feature columns before and after standardization are shown below:

Bike:



	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
0	-1.728500	-1.350369	-0.999317	-1.59332	-2.99124	1.496418	-1.074729	0.000000	-0.634969	-0.591293	2.243804	-1.348183
1	-1.723764	-1.350369	-0.999317	-1.59332	-2.99124	-1.496418	-1.074729	0.000000	-0.529474	-0.651958	1.473273	-0.211468
2	-1.719028	-1.350369	-0.999317	-1.59332	-2.99124	-0.997612	1.074729	-1.835218	-1.442411	-1.660382	-0.343869	-0.214436
3	-1.714293	-1.350369	-0.999317	-1.59332	-2.99124	-0.498806	1.074729	-1.835218	-1.422547	-1.520981	0.731487	-1.350119
4	-1.709557	-1.350369	-0.999317	-1.59332	-2.99124	0.000000	1.074729	-1.835218	-1.275282	-1.415753	-0.346088	-1.006832
5	-1.704821	-1.350369	-0.999317	-1.59332	-2.99124	0.498806	1.074729	-1.835218	-1.398794	-1.391582	0.224750	-2.262800
6	-1.700086	-1.350369	-0.999317	-1.59332	-2.99124	0.997612	1.074729	0.000000	-1.441547	-1.541127	0.087384	-1.241342
7	-1.695350	-1.350369	-0.999317	-1.59332	-2.99124	1.496418	-1.074729	0.000000	-1.613751	-1.826992	0.348124	0.024216
8	-1.690615	-1.350369	-0.999317	-1.59332	-2.99124	-1.496418	-1.074729	-1.835218	-1.759432	-2.109753	-0.365677	1.251940
9	-1.685879	-1.350369	-0.999317	-1.59332	-2.99124	-0.997612	1.074729	-1.835218	-1.691145	-1.896739	-0.023401	-0.537568
10	-1.681143	-1.350369	-0.999317	-1.59332	-2.99124	-0.498806	1.074729	0.000000	-1.591402	-1.647747	1.405008	-1.842571

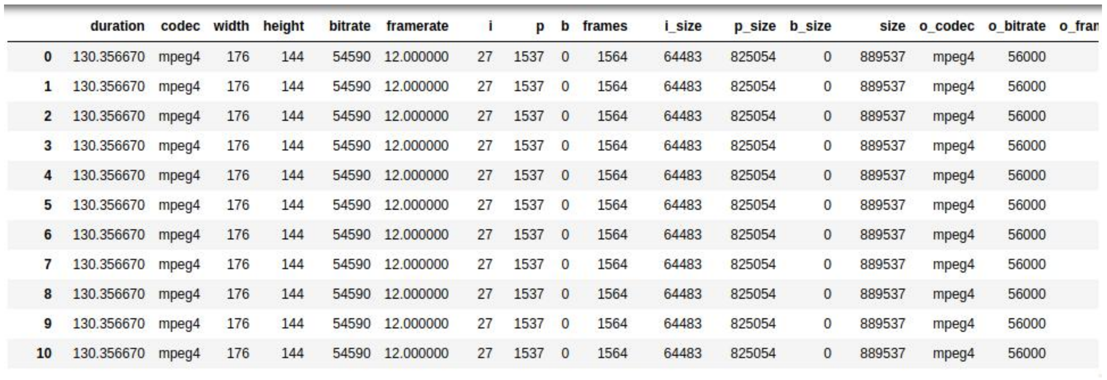
Figure 10. Bike dataset before standardization.



	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
0	1	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446
1	2	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539
2	3	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309
3	4	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296
4	5	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900
5	6	1	0	1	0	4	1	1	0.204348	0.233209	0.518261	0.089565
6	7	1	0	1	0	5	1	2	0.196522	0.208839	0.498696	0.168726
7	8	1	0	1	0	6	0	2	0.165000	0.162254	0.535833	0.266804
8	9	1	0	1	0	0	0	1	0.138333	0.116175	0.434167	0.361950
9	10	1	0	1	0	1	1	1	0.150833	0.150888	0.482917	0.223267
10	11	1	0	1	0	2	1	2	0.169091	0.191464	0.686364	0.122132

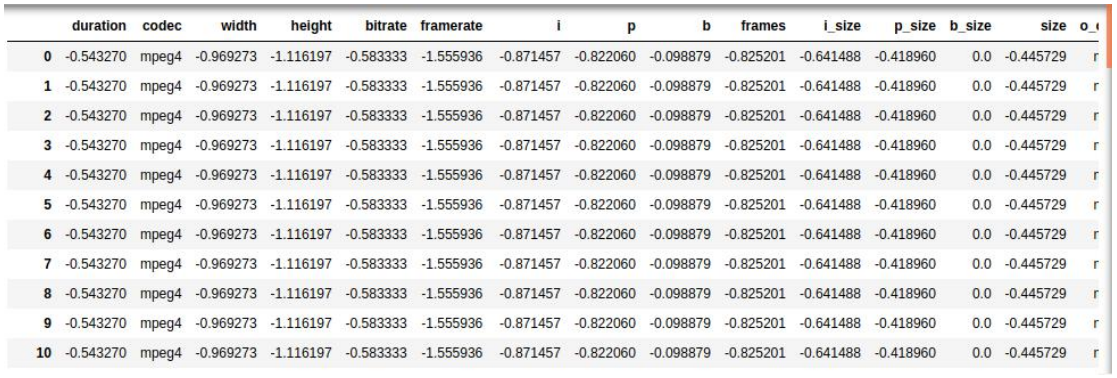
Figure 11. Bike dataset after standardization.

Video:



	duration	codec	width	height	bitrate	framerate	i	p	b	frames	i_size	p_size	b_size	size	o_codec	o_bitrate	o_fran
0	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
1	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
2	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
3	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
4	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
5	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
6	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
7	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
8	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
9	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	
10	130.356670	mpeg4	176	144	54590	12.000000	27	1537	0	1564	64483	825054	0	889537	mpeg4	56000	

Figure 12. Video dataset before standardization.



	duration	codec	width	height	bitrate	framerate	i	p	b	frames	i_size	p_size	b_size	size	o_
0	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
1	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
2	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
3	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
4	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
5	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
6	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
7	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
8	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
9	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r
10	-0.543270	mpeg4	-0.969273	-1.116197	-0.583333	-1.555936	-0.871457	-0.822060	-0.098879	-0.825201	-0.641488	-0.418960	0.0	-0.445729	r

Figure 13. Video dataset after standardization.

3.1.4 Feature Selection

In the feature selection, we are applying sklearn.feature_selection.mutual_info_regression and sklearn.feature_selection.f_regression functions. The first one is to estimate the mutual information (MI), a nonnegative value, between each feature and the label, which measures the dependency between the variables(0 for no dependency). The second one is to return 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 - Feature Selection

Bike:

Feature	instant	season	yr	mnth	holiday	weekday
value	0.90494168	0.21450326	0.27910152	0.37956772	0.01105096	0.04433741

Feature	workingday	weathersit	temp	atemp	hum	windspeed
value	0.02698772	0.06473073	0.38937705	0.46445913	0.04549788	0.05546288

Table 1. Mutual information(MI) of the features: ‘instant’, ‘season’, ‘yr’, ‘mnth’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’, ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’.

Feature	instant	season	yr	mnth	holiday	weekday
value	476.810739	143.967652	344.890585	62.0046245	3.42144104	3.33109137

Feature	workingday	weathersit	temp	atemp	hum	windspeed
value	2.73674228	70.7292978	473.471710	482.454310	7.46194	42.4378415

Table 2. F-score of the features: ‘instant’, ‘season’, ‘yr’, ‘mnth’, ‘holiday’, ‘weekday’, ‘workingday’, ‘weathersit’, ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’.

Video:

Feature	duration	codec	width	height	bitrate	framerate
value	0.88136783	0.23324829	0.44329772	0.43607145	0.97681943	0.54425916

Feature	i	p	b	frames	i_size	p_size
value	0.94751246	0.89155692	0.00728664	0.89732629	0.88729770	0.82902773

Feature	b_size	size	o_codec	o_bitrate	o_framerate	o_width
value	0.00954619	0.87332969	0.86411458	0.12055704	0.09391234	0.99544248

Feature	o_height
value	1.0

Table 3. Mutual information(MI) of the features: 'duration', 'codec', 'height', 'width', 'bitrate', 'framerate', 'i', 'p', 'b', 'frames', 'i_size', 'p_size', 'b_size', 'size', 'o_codec', 'o_bitrate', 'o_framerate', 'o_width', 'o_height'.

Feature	duration	codec	width	height	bitrate	framerate

value	1.35917894	2.23138873	79.8618128	78.4337189	106.831417	23.1332686

Feature	i	p	b	frames	i_size	p_size
value	0.13355904	0.00020355	0.06643592	0.00016288	5.36454949	18.7112796

Feature	b_size	size	o_codec	o_bitrate	o_framerate	o_width
value	nan	18.1296425	23.8971465	102.563468	53.7602181	1822.54839

Feature	o_height
value	1785.89675

Table 4. F-score of the features: 'duration', 'codec', 'height', 'width', 'bitrate', 'framerate', 'i', 'p', 'b', 'frames', 'i_size', 'p_size', 'b_size', 'size', 'o_codec', 'o_bitrate', 'o_framerate', 'o_width', 'o_height'.

We know that mutual information is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. While F scores is a way of comparing the significance of the improvement of a model, with respect to the addition of new variables.

So we know that 'o_width', 'o_height' have higher dependency and are more significant.

The step of using the above two functions to find the features based on the MI and F-scores will reduce the RMSE in the performance, which is to make the algorithm more efficient and accurate.

3.2 Training

In this part, we are training multiple algorithms and comparing their performance using RMSE.

3.2.1 Linear Regression

Question 9 - Explain how each regularization scheme affects the learned hypotheses

The ordinary linear regression is not penalized for the choice of weight, which means a large weight will be added to the particular feature. However, it could lead to overfitting in some small datasets. Lasso regression is penalized for the sum of absolute values of the weights, and Ridge regression is penalized for the sum of squared value of the weights, which tends to have smaller absolute values and more evenly distributed, correspondly saying closer to zeros.

Question 10 - Report the best regularization scheme

Bike:

Table 5 below shows the RMSE score with the default parameter alpha = 1.

Model	Linear	Ridge	Lasso
RMSE for train data	857.4141611925413	859.8053931082766	859.5323596000928
RMSE for validation data	892.2260488561709	888.6462039343294	889.0131348207345

Table 5. Comparison of RMSE between Linear Regression, Ridge Regression, and Lasso Regression.

By observation, we can see that Linear Regression has the lowest RMSE for train data, but Ridge Regression has the lowest RMSE for validation data. Overall, we conclude that Linear Regression is the best regularization scheme for the train data and Ridge is the best regularization scheme for the validation data.

Video:

Model	Linear	Lasso	Ridge
RMSE for train data	10.997130336405487	13.222088026653259	10.997131355003972
RMSE for validation data	11.092750131875865	13.226233864112562	11.071756306602174

Table 6. Comparison of RMSE between Linear Regression, Lasso Regression, and Ridge Regression.

We can see that the RMSE of linear regression and ridge regression is very close, both smaller than Lasso's. We conclude Ridge Regression to be the best regularization scheme.

Question 11 - Feature Scaling

Since we applied feature scaling in the original regularization, here we remove the feature scaling and compare the results again.

Bike:

Table 7 below shows the RMSE score with the default parameter alpha = 1 after removing feature scaling.

Model	Linear	Ridge	Lasso
RMSE for train data	857.4141611925413	865.0786146121791	859.5562587637785
RMSE for validation data	892.2260488561709	886.446381339652	887.0165204626616

Table 7. Comparison of RMSE between Linear Regression, Ridge Regression, and Lasso Regression without feature scaling.

By observation, we can still conclude that Linear Regression is the best regularization scheme for the train data and Ridge is the best regularization scheme for the validation data. Moreover, feature scaling does not affect the regular linear regression at all since the RMSE for both train data and validation data remain. Although both Ridge regression and Lasso regression are affected by removing the standard deviation of the model, Ridge Regression is affected more than the Lasso Regression, thus Ridge relies more on the feature scaling than Lasso.

Video:

Model	Linear	Lasso	Ridge
RMSE for train data	10.997192219488088	16.106139737795253	10.997215016689326
RMSE for validation data	77555708.90217151	16.056460831129616	11.05541212163902

Table 8. Comparison of RMSE between Linear Regression, Lasso Regression, and Ridge Regression.

We can see that without feature scaling, we may have higher RMSE using Lasso and Linear Regression

Question 12 - p-values

Table 9 below shows the F-score and the corresponding p-value for the 12 different features from linear regression packages.

Feature	F score	p-value
instant	476.8107399195832	1.0207631771174209e-81
season	143.96765259091728	2.1339966843421292e-30
yr	344.8905855356793	2.4835399044546224e-63
mnth	62.004624548332195	1.24311177786548e-14
holiday	3.4214410399719473	0.0647593579261283
weekday	3.3310913651745624	0.06839080695470057
workingday	2.7367422831914108	0.09849496160025555
weathersit	70.72929782920811	2.1509758214249938e-16
temp	473.4717105349773	2.8106223975901415e-81
atemp	482.4543105289897	1.854504125284329e-82
hum	7.46193999634535	0.006454143325437774
windspeed	42.43784159346339	1.3599586778866672e-10

Table 9. F-score and corresponding p-values for the 12 different features.

Based on the results shown above in F-score, ‘instant’, ‘yr’, ‘temp’, ‘atemp’ are the most significant features with high F scores and low p-values.

Video:

P-value is the probability of obtaining test results at least as extreme as the results actually observed during the test, assuming that the null hypothesis is correct. So we can see that, 'bitrate', 'o_bitrate', 'o_width', 'o_height' are the most significant features with high F scores and low p-values.

3.2.2 Polynomial Regression

Question 13 - Most salient features

Bike:

By using the scikit-learn library, we perform polynomial regression by crafting products of raw features up to a certain degree and applying linear regression on the compound features. We find that there are 7 most salient features in order, which are ‘atemp’, ‘instant’, ‘temp’, ‘yr’, ‘season’, ‘weathersit’, ‘mnth’.

Video:

By using the scikit-learn library, we can see the most salient features are 'o_width', 'o_height', 'o_bitrate', It is because that the length and width of the output video and the output bitrate significantly affect the transcoding time.

Question 14 - Degree of Polynomial

Bike:

Table 10 below shows the RMSE of regression in different degrees of polynomial (degree 1 to degree 7).

Degree of polynomial	RMSE for train data	RMSE for validation data
1	1745.7504107225916	1776.9896876501662
2	1116.1546737556946	1155.1378458650688
3	673.5811192644408	761.9635888638531
4	484.0019126234376	680.9692863733092
5	341.61544131410056	712.439976945861
6	222.98079502211868	859.2297648037695
7	111.75959855527886	1349.4577053143482

Table 10. RMSE of regression in different degrees of polynomial (degree 1 to degree 7)

By observation, the best degree of polynomial is at degree four, which has the best case of RMSE for validation data. For train data, the RMSE is decreasing as degree increases. However, after reaching a degree of 5 and more the validation RMSE is increasing significantly, which is overfitted. Thus, degree of four in polynomial regression is the best.

Video:

Degree of polynomial	RMSE for train data	RMSE for validation data
1	10.997215016689326	11.05541212163904
2	6.234357762566969	10.755119924667742
3	3.49018259153231	1265.0812629855243

Table 11. RMSE of regression in different degrees of polynomial (degree 1 to degree 3)

For video transcoding data set, we find that degree two of polynomial regression best fits. We can see from table 11, as degree goes from one to two, both the training RMSE and the validation RMSE go down. When the degree goes to three, although training RMSE is further reduced, validation RMSE has increased significantly. It is overfitted. So degree two of polynomial regression is best.

Question 15 - Explain the meaning of crafting inverse of certain features (Video only)

We craft inverse of certain features and the train RMSE becomes 6.216067659871525, while validation RMSE becomes 8.587766389112108. This is because that craft inverse introduces a new combining feature which is also highly related . This does boost accuracy.

3.2.3 Neural Network

We construct a multi-layer perceptron. And study the performance of different settings.

Question 16 - Comparison to Linear Regression

Bike:

We construct a 32*32*32 three-layer fully connected neural network. We select ‘ReLU’ as the activation function and ‘adam’ as the solver.

RMSE results of 10-fold cross validation of neural network are shown in Table 12. As we can see, both RMSE of train data of neural network and RMSE of validation data of neural network are better than those of linear regression. That is mainly because neural network introduces non-linearity so that a curve enables a better regression than several lines.

Model	Linear	Neural Network
RMSE for train data	857.4141611925413	460.2829252552424
RMSE for validation data	892.2260488561709	728.9421852296887

Table 12. Comparison between linear regression and neural network.

Video:

At first we use a Multilayer Perceptron with four hidden layers 64*64*64*64, set the max iter to 5000, then it seemed to be running endlessly using only cpu. So we change our method and start with a leaner network, a 13*13*13 three-layer fully connected neural network. Actually we did not give much hope for this but its performance is much better than the linear regression. That is mainly because neural network introduces non-linearity so that a curve enables a better regression than several lines. As we can see in table 13.

Model	Linear	Neural Network
RMSE for train data	10.997130336405487	2.308798067487414
RMSE for validation data	11.092750131875865	2.509650179355008

Table 13. Comparison between linear regression and neural network.

Question 17 - Good hyper-parameter set

Bike:

We adjust the number of neurons, depth, and weight decay rate to find a good hyper-parameter. RMSE results of different settings are shown in Table 12.

Setting	layer=32*32*32, beta1=0.9, beta2=0.999	layer=64*64*64, beta1=0.9, beta2=0.999	layer=128*128*128, beta1=0.9, beta2=0.999	layer=32*32, beta1=0.9, beta2=0.999	layer=64*64, beta1=0.9, beta2=0.999	layer=128*128, beta1=0.9, beta2=0.999	layer=32*32*32, beta1=0.7, beta2=0.999	layer=32*32*32, beta1=0.9, beta2=0.8	layer=32*32*32, beta1=0.7, beta2=0.8
RMSE for train data	460.2829252552424	278.4702171605001	183.01805140618214	541.6986442648421	452.58829363957267	323.7250982936125	457.1806766401222	469.19674401752246	695.3726630751802
RMSE for val data	728.9421852296887	765.3005667704758	844.5354561760271	687.1587006046102	681.0060434845893	700.8524142087555	737.2750392830538	721.5604740952037	827.2776575837263

Table 14. RMSE results of neural network of different settings

First, for the comparison of number of neurons, the first three settings show that with the number of neurons increases, RMSE for train data decreases while RMSE for val data increases. This infers that the loss of a bigger neural network decreases more easily with ‘adam’ solver. This could be due to ‘adam’ solver, that lowers learning rate at each iteration. However, the trained network may have lower generalization capability.

Second, as for the comparison of depth of neural network, the first six settings show that a deeper neural network has lower RMSE for train data due to similar reason, but a higher RMSE for validation data inferring a lower generalization capability.

Finally, with respect to weight decay rate, the last three settings compared to the first one can show the effect of the decay of simply first moment vector in adam, simply second moment vector in adam, and both. As we can see, a lower decay of simply first moment vector in adam or simply second moment vector has basically no effect. However, a decay of both leads to an increasing RMSE of both train data and validation data.

Hence, we recommend a neural network that has a layer with structure 64*64, beta1 with 0.9, and beta2 with 0.999, since it has a good RMSE for train data, and a fairly good RMSE for validation data.

Video:

Setting	layer=13*13*13	layer=32*32*32	layer=13*13	layer=32*32
RMSE for train data	2.308798067487414	1.5246550944322093	2.765986080149985	1.7055323453671423
RMSE for validation data	2.509650179355008	1.7885192378949566	2.9988394360155946	2.2743860083618714

Table 15. RMSE results of neural network of different settings

First, for the comparison of number of neurons, the comparison of the first two and the last two show that with the number of neurons increases, RMSE for both training and validation data decreases. Second, as for the comparison of depth of neural network, the comparison of the 1, 3 and the 2, 4 show that a deeper neural network has lower RMSE for train data due to similar reason, but a higher RMSE for validation data inferring a lower generalization capability.

Hence, we recommend a neural network that has a layer with structure 32*32*32, since it has a good RMSE for train data, and a fairly good RMSE for validation data.

Question 18 - Activation function

Bike:

We try four activation functions, ‘relu’, ‘identity’, ‘logistic’, ‘tanh’, and results are shown in Table 16.

Activation function	relu	identity	logistic	tanh
RMSE for train data	460.2829252552424	858.456861863232	4475.792377653004	4447.663715637059
RMSE for validation data	728.9421852296887	890.0988219194185	4465.422914540137	4437.807907574168

Table 16. RMSE results of neural network with different activation functions.

As we can see, the neural network with ‘relu’ as activation function has a lowest RMSE, which means that it has the best performance. And the neural network with ‘identity’ is better than ‘logistic’ and ‘tanh’.

Hence, we choose ‘relu’ as our activation function.

Video:

We try four activation functions, ‘relu’, ‘identity’, ‘logistic’, ‘tanh’, and results are shown in Table 17.

Activation function	relu	identity	logistic	tanh
RMSE for train data	2.308798067487414	11.049046917394174	2.5508341193993638	2.080869467866599
RMSE for validation data	2.509650179355008	10.926484563222383	2.7237836694008175	2.2965288607168537

Table 17. RMSE results of neural network with different activation functions.

As we can see, the neural network with 'tanh' as activation function has a lowest RMSE, which means that it has the best performance. While the neural network with 'relu', 'logistic' and 'tanh' performance similarly, the neural network with 'identity' has the highest RMSE which performance worst. So we choose 'tanh' as our activation function.

Question 19 - Depth of the network

Bike:

We design four neural network with different depth, namely 128*128, 128*128*128, 128*128*128*128, 128*128*128*128*128, and record their running time.

Layer	128*128	128*128*128	128*128*128*128	128*128*128*128*128
RMSE for train data	323.7250982936125	183.01805140618214	97.88289335811575	76.4491830796947
RMSE for validation data	700.8524142087555	844.5354561760271	956.9902331795414	950.1341182402355
Running time(s)	308.9836091995239	447.8299376964569	632.5863709449768	901.0887224674225

Table 18. RMSE and Running time of neural networks with different depth

As we can see from Table 18, a deeper neural network needs more time for training, which is one of the reasons that we do not want a very deep neural network. Another reason lies in generalization capability. A deeper neural network suffers from overfit, which lowers the RMSE for validation data.

Video:

Setting	layer=13*13	layer=13*13*13	layer=32*32	layer=32*32*32
RMSE for train data	2.765986080149985	2.308798067487414	1.7055323453671423	1.5246550944322093
RMSE for validation data	2.9988394360155946	2.509650179355008	2.2743860083618714	1.7885192378949566

Table 19. RMSE and Running time of neural networks with different depth

Just like we did at the beginning of this part of the neural network, we started with a four hidden layers 64*64*64*64 neural networks. The time it used is much longer than those with fewer layers. As the depth of the network increases, the time required for training increases significantly. Meanwhile a deeper neural network suffers from overfit, which lowers the RMSE for validation data.

3.2.4 Random Forest

Question 20 - Hyper-parameters

Bike:

Table 20 below shows the performance of the RMSE for train data and validation data by adjusting the three hyper-parameters(maximum number of features, number of trees, and depth of each tree).

Case	Max Feature	Number of trees	Depth of each tree	RMSE for train data	RMSE for validate data
1	1	100	20	264.87585741	695.12664640
2	2	100	20	250.40142482	650.60438881
3	3	100	20	245.48807604	636.20797168
4	4	100	20	246.33093333	636.20226799
5	5	100	20	244.56694391	637.54658713
6	6	100	20	246.50530142	645.15152315
7	12	100	20	254.03338637	662.30706094
8	3	50	20	249.98869540	652.41977965
9	3	200	20	242.16674711	640.37702938
10	3	100	10	294.68485967	653.35548965
11	3	100	40	246.00436202	639.91959039

Table 20. Performance of the RMSE for train data and validation data by adjusting the three hyper-parameters(maximum number of features, number of trees, and depth of each tree).

By observing cases 1-6, we find that initially RMSE decreases as the maximum number of features increases, but increases afterwards. By observing cases 3, 8, and 9, we find that RMSE decreases as the number of trees increases in training data, though the validation data increases afterwards, which means it is overfitting. By observing cases 3, 10, and 11, we find that RMSE decreases as depth of each tree increases initially, then RMSE increases as depth of each tree increases.

Video:

Case	Max Feature	Number of trees	Depth of each tree	RMSE for train data	RMSE for validate data
1	1	100	20	1.5937904191	6.3283292068734
2	2	100	20	1.11004848	5.018633169864
3	3	100	20	0.8171618290	4.3618593311508
4	5	100	20	0.6136713395	4.091401723226
5	10	100	20	0.4946636043	4.039363402723
6	15	100	20	0.482000093	4.10310560786
7	3	50	20	0.8923059169	4.485547831422
8	3	200	20	0.798839291	4.30222744794
9	3	100	10	5.605842807	6.58412778357
10	3	100	40	0.673365491	4.31450039854

Table 21. Performance of the RMSE for train data and validation data by adjusting the three hyper-parameters(maximum number of features, number of trees, and depth of each tree).

By observing cases 1-6, we find that initially RMSE decreases as the maximum number of features increases, but increases afterwards. By observing cases 3, 7, and 8, we find that RMSE decreases as the number of trees increases in training data, though the validation data increases afterwards, which means it is overfitting. By observing cases 3, 9, and 10, we find that RMSE decreases as depth of each tree increases initially, then RMSE increases as depth of each tree increases.

Since the simplest way to do regularization on random forest is to penalize the selection of new features, in our table, we control one parameter, and select the other parameter, after adjusting all the features at a limited range, the results will be same as regularization on random forest.

Question 21 - Performance

First of all, random forest performs implicit feature selection and provides a good indicator of feature importance, which is efficient. It requires no input preparation as well. Any features can be handled by a random forest. Moreover, the learning algorithm is simple to implement.

Question 22 - Pick a tree in the random forest model and plot the structure

Bike:

In this part, we pick a tree with depth of the tree that equals to four. We get the result that RMSE for train data of 688.88, and RMSE for validation data of 734.38. The tree plot is shown below in Fig. 14.

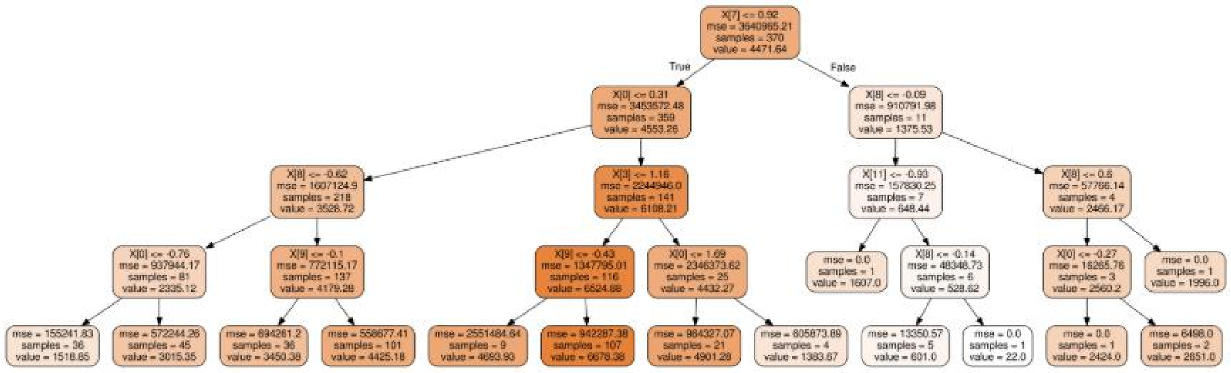


Figure 14. Random forest model with maximum depth of 4.

The feature selected for branching at the root node is ‘weathersit’. (X[7] represents the 8th feature among the 12 features). The importance of the feature increases as the depth increases. In other words, the more proportion of samples reach that node, the more important that feature is. Thus, the important features are X[0], X[9], X[8], which are correspondingly ‘instant’, ‘atemp’, and ‘atemp’. All three are included in the features explained previously, but ‘yr’ was picked as well.

Video:

In this part, we pick a tree with depth of the tree that equals to four. We get the result that RMSE for train data of 0.6733, and RMSE for validation data of 4.3145. The tree plot is shown below in Fig. 15.

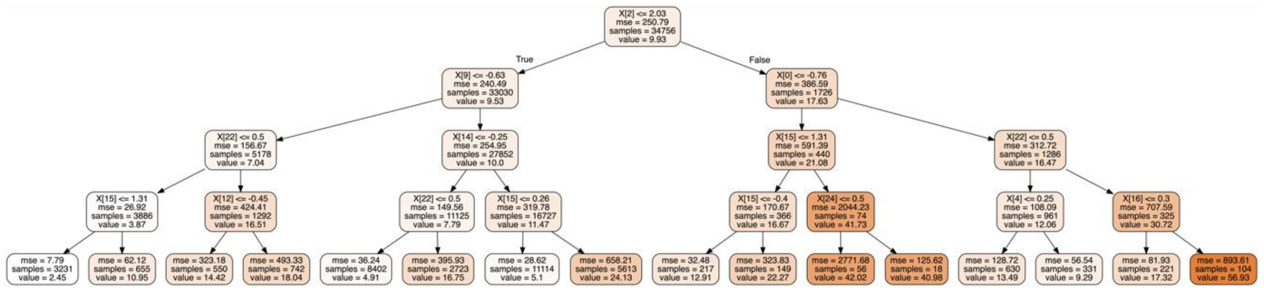


Figure 15. Random forest model with maximum depth of 4.

The feature selected for branching at the root node is ‘height. (X[2] represents the 3th feature among the features). The importance of the feature increases as the depth increases. In other words, the more proportion of samples reach that node, the more important that feature is. Thus, the important features are X[15], X[12], X[22], X[4], X[16], X[24] which are correspondingly ‘size, ‘o_width’, ‘o_codec_flv’, ‘framerate’, ‘o_height’, and ‘o_codec_mpeg4’. All three are included in the features explained previously.

3.3 Evaluation

Question 23 - Explain the difference between the training RMSE and that of validation set

Usually it is caused by the overfitting of the model. In the training RMSE, the features are adding up so the data is becoming more and more flexible. Thus, in the validation set, the training data will be easily overfitting because the pattern found in the training set is not existing in the validation set.

Question 24 - Measure and explain Out-Of-Bag Error (OOB)

Bike:
OOB score: 0.8227292501850562
R^2score: 0.12610140775517964
Video:
OOB score: 0.5900350616247864
R^2score: 0.40765868340124506
OOB score represents the score of the training dataset obtained using an out-of-bag estimate. The OOB score also represents whether to use out-of-bag samples to estimate the R^2score on unseen data.