Bike Sharing Demand Prediction

Project Deliverable 2

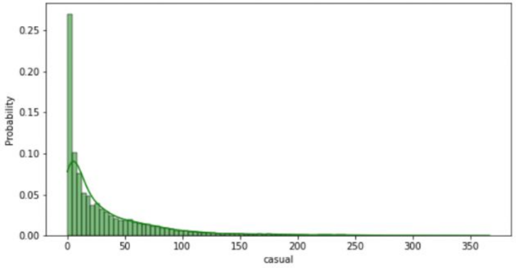
Team members: Tobias Hanl (th2999), Shuyu He (sh4330), Jingyi Feng (jf3495), Mendel Branover (mb4869), Zixiang Yin (zy2444)

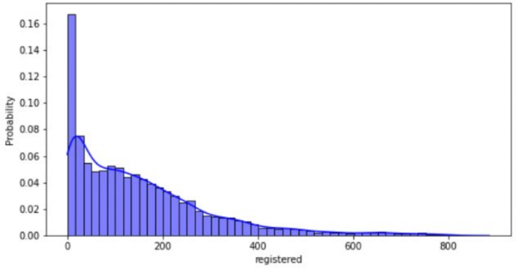
Columbia University

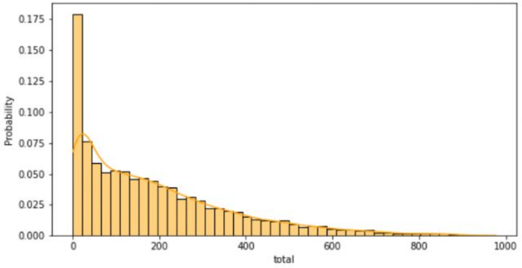
Data Exploration - Variables

- A total of 10,866 observations with 12 features
- Five categorical features: *datetime*, *season*, *holiday*, *workingday*, and *weather*
- No missing values in the dataset
- Highly Skewed targets (*casual*, *registered*, *count*)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10886 entries, 0 to 10885
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   datetime    10886 non-null  object
 1   season      10886 non-null  int64
 2   holiday     10886 non-null  int64
 3   workingday  10886 non-null  int64
 4   weather     10886 non-null  int64
 5   temp        10886 non-null  float64
 6   atemp       10886 non-null  float64
 7   humidity    10886 non-null  int64
 8   windspeed   10886 non-null  float64
 9   casual      10886 non-null  int64
 10  registered  10886 non-null  int64
 11  count       10886 non-null  int64
```





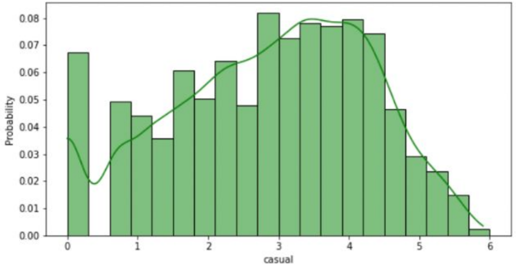


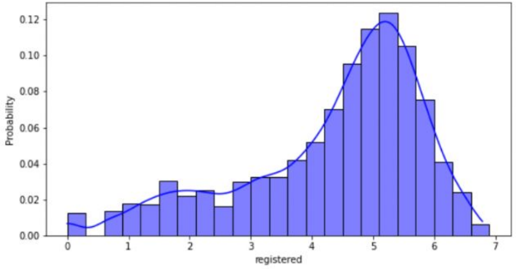
Data Preprocessing - Variables

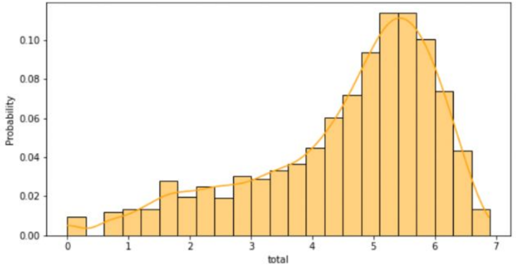
```python
df = df.rename({'count': 'total'}, axis=1)
df.head(5)
```

- Rename *count* to *total* to avoid conflict with built-in functions
- Log transform the targets
- Enrich the dataset by parsing the *datetime* into *year*, *month*, *day*, *hour*, and *dayofweek*

```python
df['year'] = pd.DatetimeIndex(df.datetime).year
df['month'] = pd.DatetimeIndex(df.datetime).month
df['day'] = pd.DatetimeIndex(df.datetime).day
df['hour'] = pd.DatetimeIndex(df.datetime).hour
df['dayofweek'] = pd.DatetimeIndex(df.datetime).dayofweek
df = df.drop("datetime", axis=1)
df.head()
```



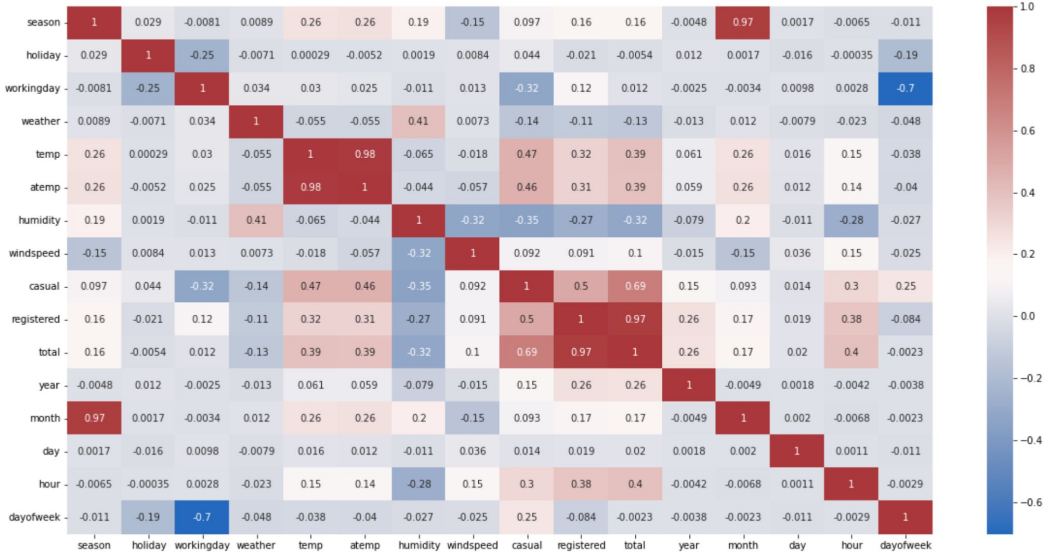




Data Exploration - Correlation Analysis

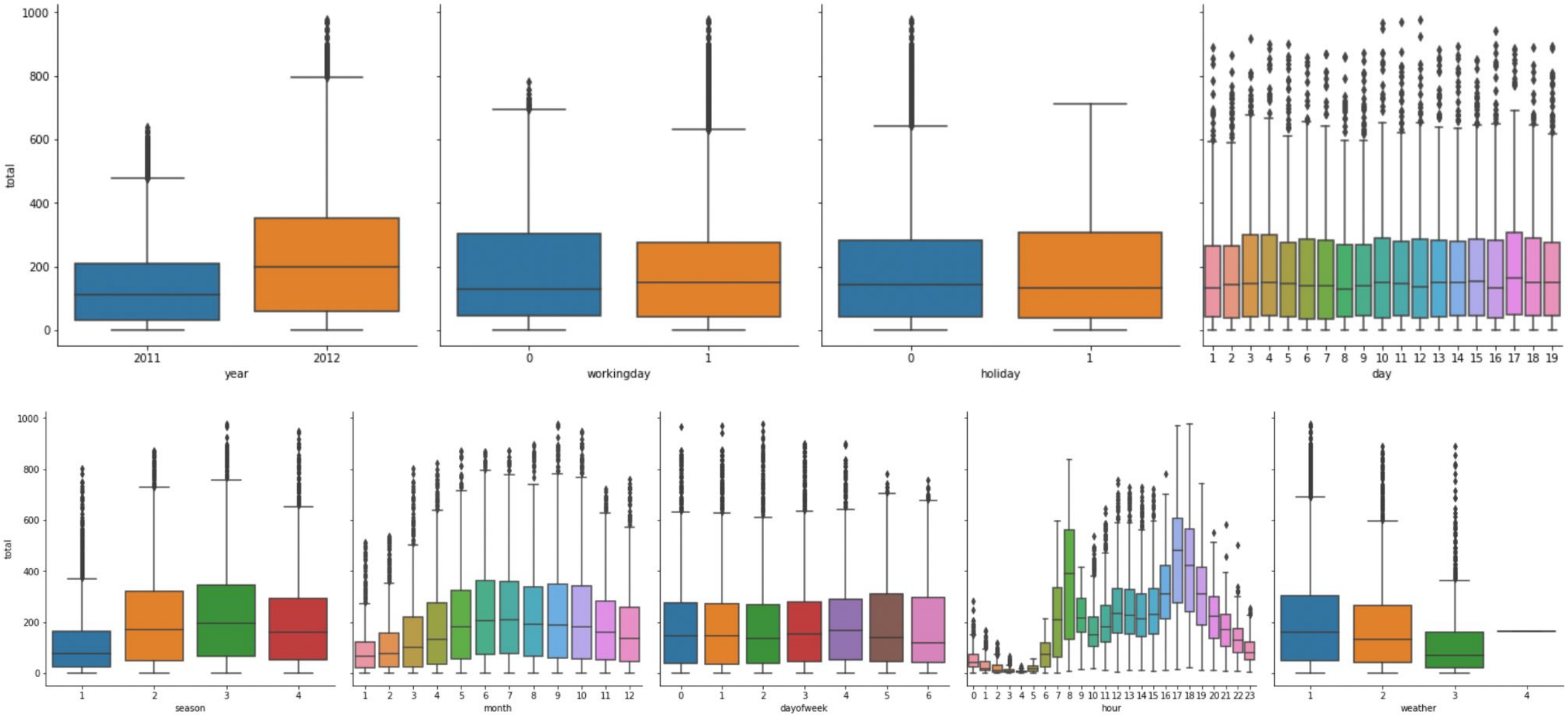
- *temp* and *atemp* have almost perfect correlation, so one can be dropped
- *month* and *season* have an almost perfect correlation, so season could potentially be dropped
- *temp* has a positive correlation with *total* (+0.37) and *registered* (0.33), and especially *casual* (+0.53)
- *hour* has positive correlation with *total* (+0.57), *registered* (+0.57) and a bit less with *casual* (+0.42)
- *humidity* has a negative correlation with *total*
- *dayofweek* and *workingday* have a positive and negative correlation with *casual* respectively, but do not influence *registered* much
- *season* has some positive correlation with all targets
- *humidity* has negative correlation with all targets
- *windspeed*, *year* and *month* have slight positive correlation with all targets
- *holiday* and *day* have little correlation with any target
- *workingday* can also be dropped since it is just a combination of dayofweek and holiday.

Correlation heatmap between 16 features



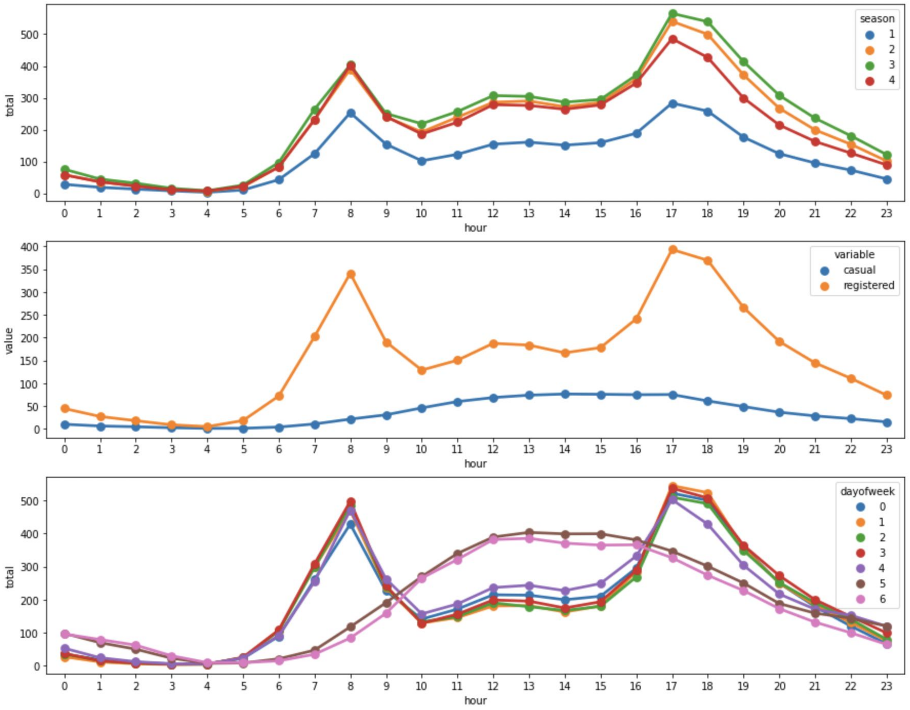
Data Visualization - Boxplots

Total against year, workingday, holiday, day, season, month, dayofweek, hour, weather



Data Visualization - Pointplot

Total against hour, categorized by season, casual/registered, dayofweek



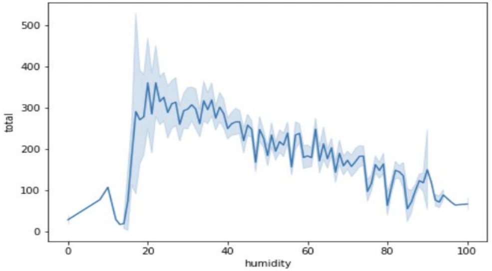
Observations:

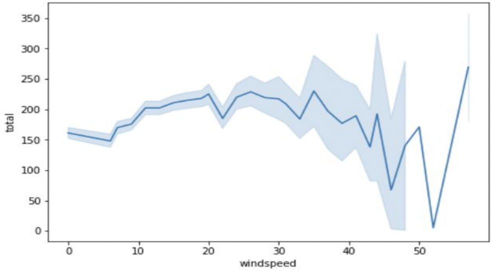
1. We can see a clear bimodal trend in both the second and the third plot:
    - During weekdays, the peak of bike rental time is around 8 am and around 5 pm to 6 pm (rush hours)
    - The peak of bike rental time for registered riders is also around 8 am and around 5 pm to 6 pm (rush hours)
2. During weekends, the plot is unimodal: the peak of bike rental time is around 11 am to 4 pm

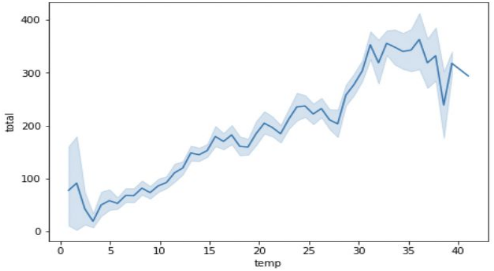
Columbia

Data Visualization - Line Plots

Total against humidity, windspeed, temp, month, month by year



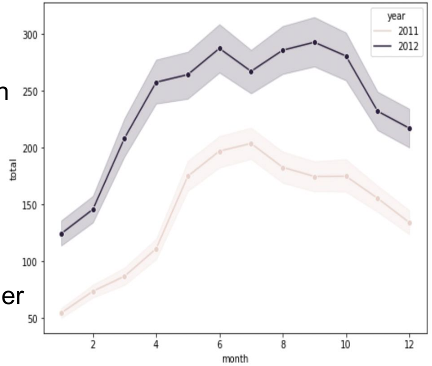


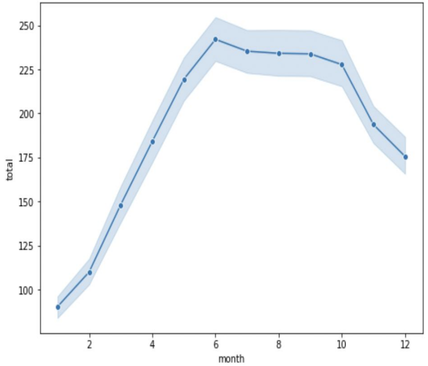


Observations:

- Higher humidity, lower bike rentals in general, but if humidity is under 20, we can see a significant drop
- Windspeed generally does not influence bike rentals, but we can see a huge drop occur when windspeed is around 50
- Higher temperature, higher bike rentals in total
- The peak of bike rentals starts in June and ends in October
- More bike rentals in 2012 than in 2011

Weather seems to influence bike rentals a lot (higher rentals if the weather is good, i.e., humidity around 20 and temperature around 30 to 35 °C).



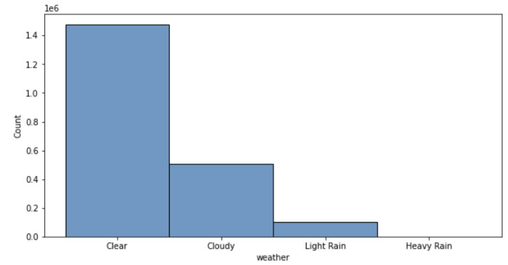


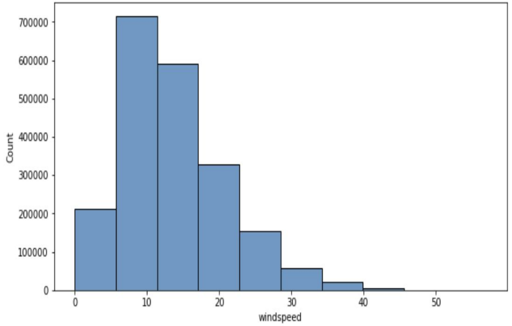
Data Visualization - Histogram

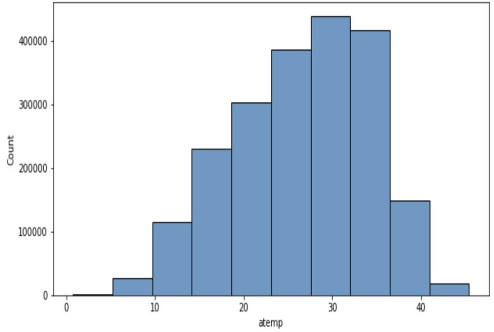
Distribution of rentals by weather and season

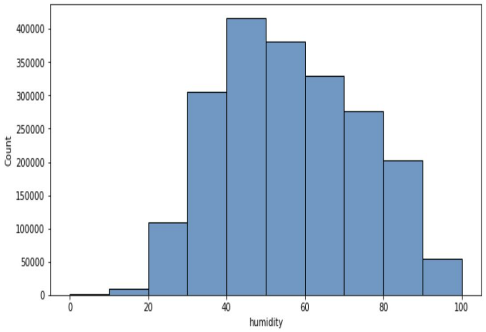
Observations:

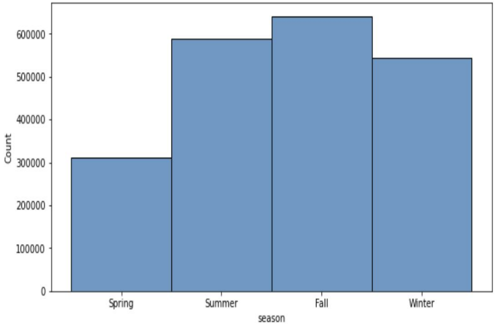
- There appear to be more rentals in the Fall season and fewer in Spring.
- The rentals appear to generally decrease as wind speed and humidity increase
- The rentals appear to increase as *atemp* increases until around 35 degrees
- It appears that most rentals are on clear sky days











Columbia

Methodology & Feature Engineering

Big Picture:

- Build two types of models for predicting the targets: Regression v.s. Time Series
- Predict *total* directly v.s. *registered + casual* to get *total*

Feature Engineering for regression on *total* directly (current progress):

- Drop features: *datetime*, *day*, *temp*, *season*, *workingday*
- Ordinal encoding *year*
- One-hot encoding weather
- Target encoding *month*, *dayofweek*, *hour*
- Normalize the transformed features

```
num_features = ['atemp', 'humidity', 'windspeed']
ord_features = ['year']
ohe_features = ['weather']
te_features = ['month', 'dayofweek', 'hour']
preprocess = make_column_transformer((OrdinalEncoder(), ord_features),
                                     (OneHotEncoder(), ohe_features),
                                     (TargetEncoder(), te_features), remainder='passthrough')
```

Columbia University

Preliminary ML Model Training & Results:

- We selected 6 regression models Elastic Net, KNN, Linear SVM, Decision Tree, Random Forest, and HistGradient Boosting and trained them on both original (first row) and transformed features (second row)
- R square is used as the metric during cross-validation for comparing model performance and doing model selection as shown below
- Before feature engineering, no model achieves a score higher than 0.2.
- After feature engineering, the lowest score of 0.7258 was achieved by KNN
- Among all models, HistGradient Boosting Regression reached the best performance score of 0.8929.