Bike sharing demand forecasting

(a) Station-free bike sharing

(b) Bike sharing system with docking station

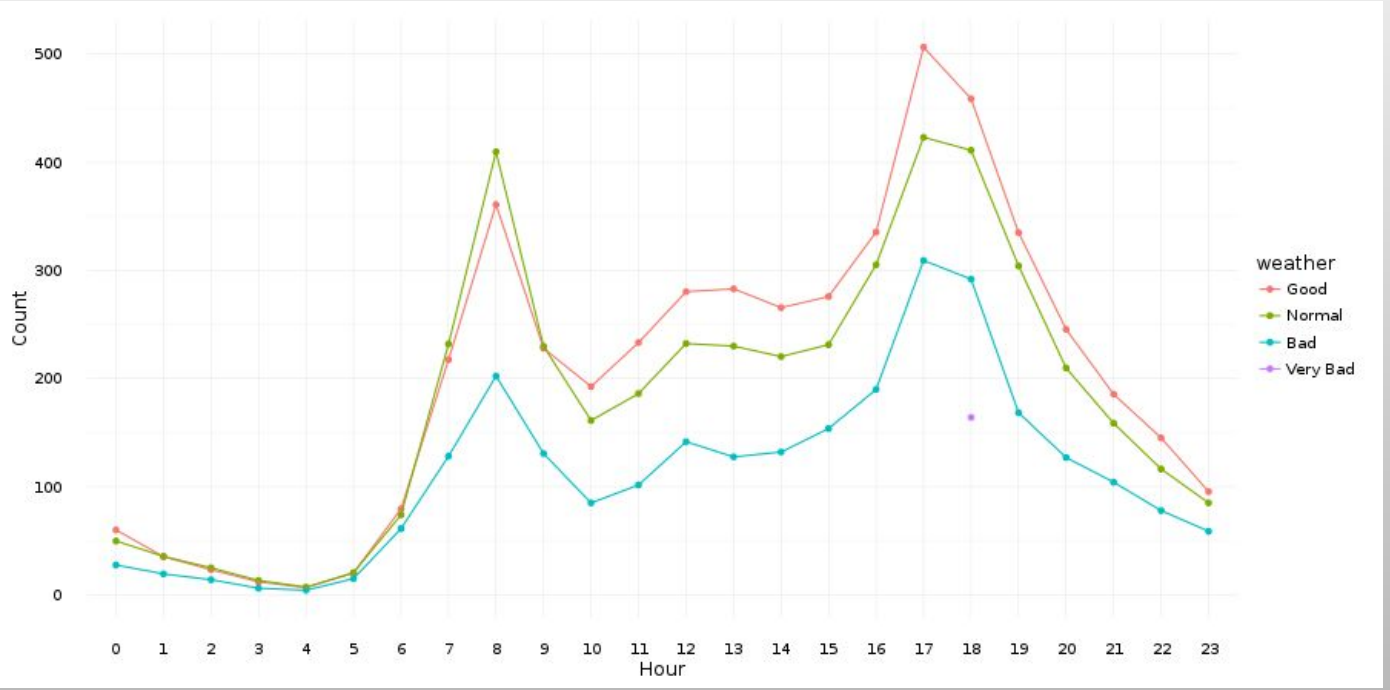
C. Xu et. al., Transportation research, 2018

Tianning Gao
Siva Saket Sripada

Motivation

Massive imbalance in demand and availability of rental bikes, especially during “peak” hours

Demand forecasting



Challenges

Imbalance between demand and supply

Predicting dynamic bike demand based on current and previous time

Time series data - Auto Regression

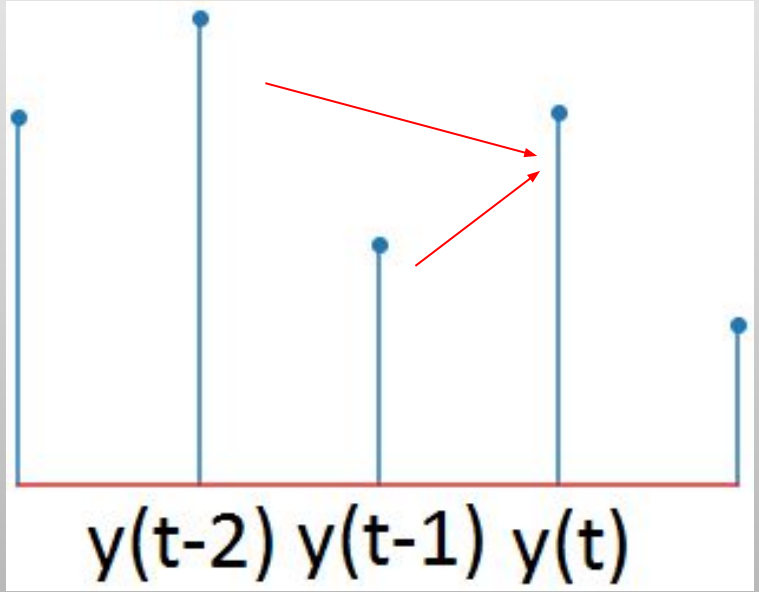
Null values in “windspeed” feature affect prediction

Consume a lot of computing resources

Register, casual features skewed initial predictions - just like categorical features

Auto Regression

y_pred(t) = f(X(t), y(t-1), y(t-2))



Time	X(t)	y	y(t-1)	y(t-2)
0	3	16	Nan	Nan
1	76	5	16	Nan
2	78	0	5	16
3	15	9	0	5
4	386	20	9	0
5	48	24	20	9
6	66	38	24	20
7	49	25	38	24

Dataset (originally from Capital Bikeshare, maintained by UCI ML repository)

datetime - **hourly date** + timestamp

season - **1** = spring, **2** = summer, **3** = fall, **4** = winter

holiday - whether the day is considered a holiday

workingday - whether the day is neither a weekend nor holiday

weather -

1: Clear, Few clouds, Partly cloudy, Partly cloudy

2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist

3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds

4: Heavy Rain + Ice Pellets + Thunderstorm + Mist, Snow + Fog

temp - temperature in Celsius

atemp - "feels like" temperature in Celsius

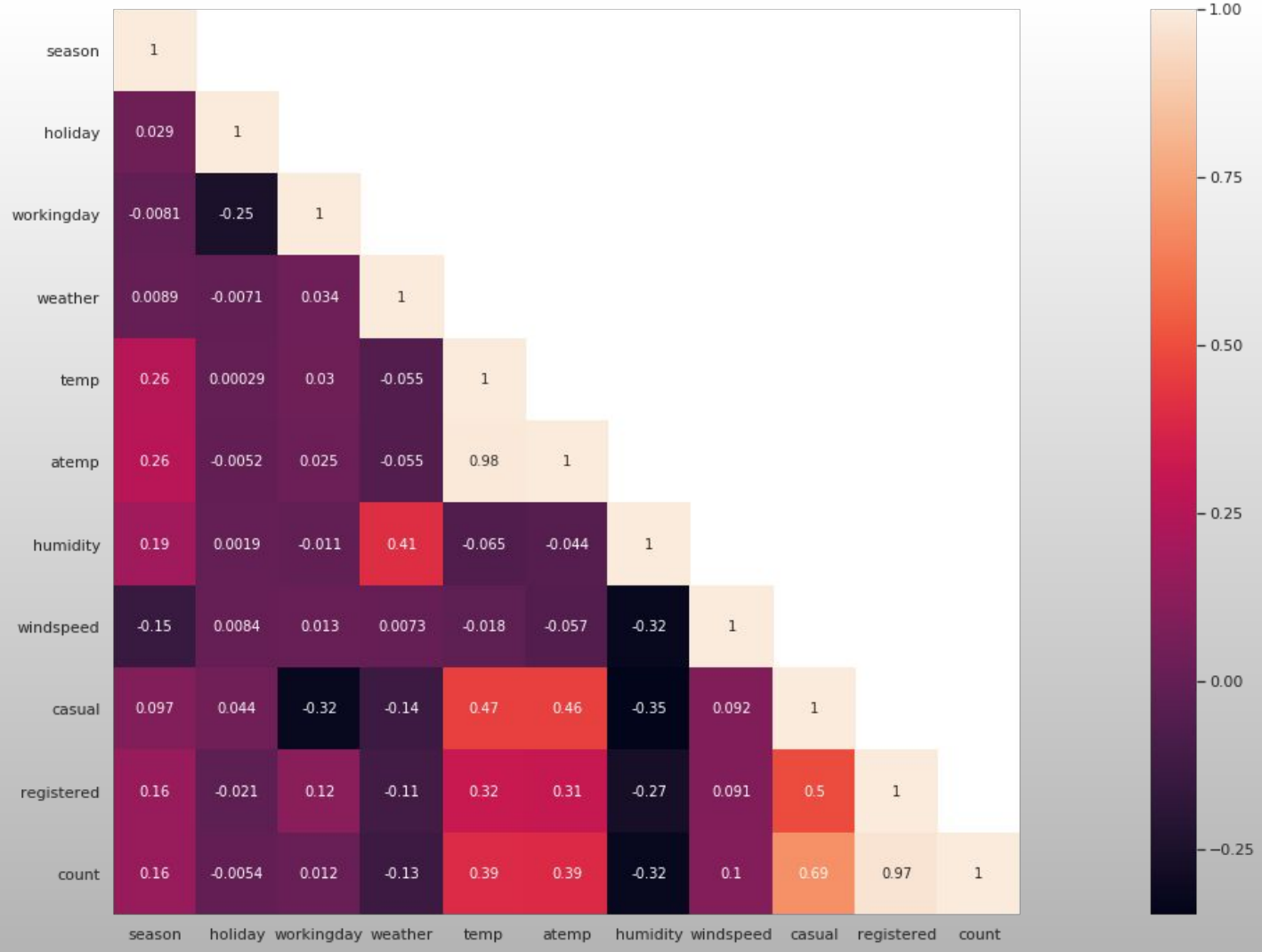
humidity - relative humidity

windspeed - wind speed

casual - number of non-registered user rentals initiated

registered - number of registered user rentals initiated

count - number of total rentals


season 1
holiday 0.029 1
workingday -0.0081 -0.25 1
weather 0.0089 -0.0071 0.034 1
temp 0.26 0.00029 0.03 -0.055 1
atemp 0.26 -0.0052 0.025 -0.055 0.98 1
humidity 0.19 0.0019 -0.011 0.41 -0.065 -0.044 1
windspeed -0.15 0.0084 0.013 0.0073 -0.018 -0.057 -0.32 1
casual 0.097 0.044 -0.32 -0.14 0.47 0.46 -0.35 0.092 1
registered 0.16 -0.021 0.12 -0.11 0.32 0.31 -0.27 0.091 0.5 1
count 0.16 -0.0054 0.012 -0.13 0.39 0.39 -0.32 0.1 0.69 0.97 1
season holiday workingday weather temp atemp humidity windspeed casual registered count
1.00
0.75
0.50
0.25
0.00
−0.25

Methods deployed

Support Vector Machine (Regression)

Ridge Regression

Bagging Regression (Base: Decision Tree Regressor)

Random Forest Regressor (Base: Decision Tree Regressor)

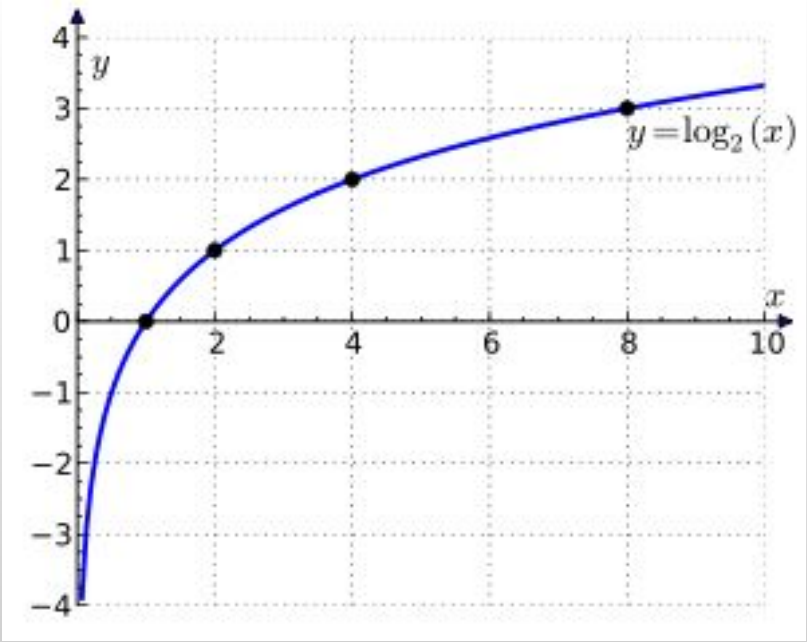
Adaboost Regressor (Base: Decision Tree Regressor)

Auto-Regression applied to every algorithm

Evaluation

Evaluated by RMSLE (The Smaller the better):

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(y\_pred_i + 1) - \log(y\_true_i + 1))^2}$$



“Under estimation costs more than over estimation”

Linear models

SVR=SVR(C=100.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma=0.025118864315095794, kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False)
Ridge=Ridge(alpha=11.313708498984761, copy_X=True, fit_intercept=True, max_iter=None, normalize=False, random_state=None, solver='auto', tol=0.001)

1. Support Vector Regressor (SVR with rbf kernel) without Auto-regression (AR)
   Root-mean-squared-log-error - **0.6098589475763458**
2. SVR with AR using 1 hr of time-delay
   Root-mean-squared-log-error - **0.5837179267091405**
3. Ridge Regressor without AR
   Root-mean-squared-log-error - **1.2028770646253422**
4. Ridge Regressor with AR using 2 hr of time-delay
   Root-mean-squared-log-error - **0.7235342542289207**

Ensemble models

Bag = BaggingRegressor(base_estimator=DecisionTreeRegressor( ), bootstrap=True, bootstrap_features=False, max_features=1.0, max_samples=1.0, n_estimators=100, n_jobs=None, oob_score=False, random_state=None, verbose=0, warm_start=False)

1. Bagging Regressor without Auto-regression (AR)

   Root-mean-squared-log-error - **0.534895803604716**

2. Bagging Regressor with AR using 1 hr of time-delay

   Root-mean-squared-log-error - **0.3334065624791147**

Ensemble models

RFR = GridSearchCV(RandomForestRegressor(random_state=0,n_estimators=500),RFRparam_grid, cv=5,scoring='neg_mean_squared_log_error')

1. Random Forest Regressor (RFR) without Auto-regression (AR)

   Root-mean-squared-log-error - 0.535

2. RFR with AR using 1 hr of time-delay

   Root-mean-squared-log-error - 0.335

3. RFR with AR using 2 hr of time-delay

   Root-mean-squared-log-error - 0.333

Results

	SVR	Ridge Regressor	Random Forest Regressor	Bagging Regressor	AdaBoost Regressor
RMSLE (no AR)	0.60	1.20	0.53	0.53	0.89
RMSLE (best with AR)	0.58	0.72	0.33	0.33	0.59

Future work

LSTMs

Fast Recurrent Neural networks

Graphical Weight Correlated Network

Graph convolutional neural network

Thank You

Questions?