UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

# ASSIGNMENT 2

The goal of this assignment is to get you started with predictive analytics. You will first prepare and explore the data, and run a basic regression. You will then predict the variable COUNT as a function of the other variables. You will also determine the effect of bad weather on the number of bikes rented. Finally, you will build alternative models, measure and compare their predictive performance, make *data-informed* and *data-driven* inferences for a business case.

**Assignment Instructions**

You will use data from DC's Capital Bikeshare (also serves Maryland and Virginia). Capital Bikeshare has about 30K members, and served about 23.6 million trips through its 550 stations. In this dataset, we combined the Capital Bikeshare data with weather data to gather insights.

**Data Dictionary:**

1. DATE -*You'll also create a* MONTH *variable using this*
2. HOLIDAY: Whether the day is a U.S. holiday or not.
3. WEEKDAY: If a day is neither a weekend nor a holiday, then WEEKDAY is YES.
4. WEATHERSIT: The values are (1) Clear/Few clouds (2) Misty (3) Light snow or light rain (4) Heavy rain, snow, or thunderstorms.
5. TEMP: Average temperature in Celsius.
6. ATEMP: "Feels like" temperature in Celsius.
7. HUMIDITY: Humidity out of 100 (not divided by 100).
8. WINDSPEED: Wind speed in km/h.
9. CASUAL: Count of bikes rented by casual bikeshare users.
10. REGISTERED: Count of bikes rented by registered bikeshare members.
    COUNT: Total count of bikes rented by both casual users and members **-You'll create this**

Before you start:

- Load the following four libraries in the given order: *tidyverse, tidymodels, plotly, skimr*
- Load the bikeshare data and call it *dfb0rg*
- Explore the dataset using `skim()` etc.

1) **Data preparation**
    a) **Create the additional variables:**
        i) Create the COUNT variable and add it to the data frame.
        ii) Extract MONTH from the DATE variable and add it to the data frame. **This time, do NOT use lubridate. Use the base `months()` function instead.**
    b) **Scale the data (and save it as *`dfbStd`* ):** Start by standardizing the four variables, TEMP, ATEMP, HUMIDITY, WINDSPEED. If you don't remember what it means to standardize a variable, see the link. Surely, you don't need to do this manually!

2) **Basic regression in R:** In `dfbStd`, run a regression model `fitAll` using COUNT as the DV, and all the variables as independent variables. [ Don't forget to use summary(fitAll) ]
    a) Does this appear to be a good model? Why or why not?
    *This does not appear to be a good model. The R squared and adjusted R squared are 1, this means that the model completely explains the variance in the DV using the IVs. This is possible because the DV count is a mathematical sum of Casual and Registered IVs. Therefore, this model fails to capture the true statistic significance and impact of other IV on the DV count reliably making it and overfitting model.*

    b) According to your model, what is the effect of humidity on the total bike count in a formal interpretation? Does this finding align with your answer to Part (a)?
    *When the humidity increases by 1 unit the total bike count changes by $1.4*10^{-13}$ on average provided that all the other parameters remain the same.*
    *This shows that the humidity does not impact the total bike count (can also be seen from a relatively high p value of 0.0942) but in reality, humidity would have an impact on the total bike count as people may avoid biking outdoors based on humidity. Thus, it aligns with the explanation above that due to overfitting introduced due to the casual and registered variables the model is not able to predict the effect of other IVs properly.*

**In the rest of the assignment, use the original data frame `dfbOrg`:**

3) **Working with data and exploratory analysis:**
    a) Add a new variable and call it **BADWEATHER**, which is "YES" if there is light or heavy rain or snow (if WEATHERSIT is 3 or 4), and "NO" otherwise (if WEATHERSIT is 1 or 2). You know what functions to use at this step.

b) Present a scatterplot of COUNT (y-axis) and ATEMP (x-axis). Use different colors or symbols to distinguish "bad weather" days. Briefly describe what you observe.
*We see a non-linear relationship between the "feels like" temperature ATEMP and COUNT. The count of bikes used increases as the temperature increases from 0 but then decreases again as it gets hotter. Also, we observe that the usage is considerably low on bad weather days than on the other days.*

c) Make two more scatterplots (and continue using the differentiated coloring for BADWEATHER) by keeping ATEMP on the x-axis and changing the variable on the y-axis: One plot for CASUAL and another for REGISTERED. Given the plots:

   i) How is *temperature* associated with casual usage? Is that different from how it is associated with registered usage?
   *Casual users are more sensitive to temperature changes than registered users. The variance in casual users when the temperature rises is much higher than that of registered users. Also, the drop in casual users when temperature becomes increasingly high is also high than registered users.*
   How is *bad weather* associated with casual usage? Is that different from how it is associated with registered usage?
   *In case of bad weather days, the casual bikers do not use the bikes, very few of them use the bikes. But we can see that a considerable number of registered users still use the bikes on a bad weather day as compared to the casual users.*

   ii) Do your answers in (i) and (ii) make logical sense? Why or why not?
   *Yes, they do make logical sense because registered users are inclined to use the bike even is slightly uncomfortable conditions as they may have already paid the registration fees and feel the urge to use it while the casual users will prefer to use the bikes more only when they find it really comfortable and avoid using the bikes otherwise as they may have other options.*

   iii) Keep ATEMP in the x-axis, but change the y-axis to COUNT. Remove the color variable and add a geom_smooth() without any parameters. How does the overall relationship between temperature and bike usage look? Does this remind you of Lab 2? Why do you think the effects are similar?
   *The overall relationship between the COUNT and ATEMP variables is non-linear and close to a parabolic curve. Yes, it does remind us on lab 2 where we compared sales with the temperature. The effect of temperature on both total bike count and sales is similar as it does not explain the causal effect between the variables. Instead, it is related to how people/customers behave when the weather fluctuates. When the temperature increases from low values the people will be more inclined to go outdoors. When the temperature rises beyond the range of pleasant weather people are less inclined to be outdoors thus creating a non-linear parabolic relationship.*

4) **More linear regression:** Using dfbOrg, run another regression for COUNT using the variables MONTH, WEEKDAY, BADWEATHER, TEMP, ATEMP, and HUMIDITY.

   a) What is the resulting adjusted $R^2$? What does it mean?
   *The resulting adjusted R2 value is 0.521. The value explains how much variance in the model is explained by the independent variables. Unlike R2 it also considers the number of independent variables in the model.*

   b) State precisely how BADWEATHER is associated with the predicted COUNT.
   *When the weather is bad the total bike count **reduces by 1954.835** on average compared to when the weather is not bad provided all the other parameters remain the same.*

   c) What is the predicted count of rides on a weekday in January, when the weather is BAD, and the temperature is 20° and feels like 18°, and the humidity is 60%?
   *The predicted count of rides would be approximately **2520.497.***
   *Calculation:*
   *3967.981 + 69.745 – 858.334 – 1954.835 + 20 x 184.596 -48.64 x 18 -60 x 25.341*

   d) Do you have any concerns about this model or your predicted COUNT in **Q4-c**? Why or why not?
   *Yes, there are concerns with the prediction of this model. Though the model may be able to explain effects of most IVs involved it does not explain the effect of ATEMP and WEEKDAY variables very well. The p-value for these two variables are high (0.184 and 0.526 respectively) which means that their coefficients are not statistically very significant. Thus, a prediction made using these coefficients may not be accurate.*

5) **Regression diagnostics:** Run the regression diagnostics for the model developed in **Q4**. Discuss whether the model complies with the assumptions of multiple linear regression. ***If you think you can mitigate a violation, take action,*** and check the diagnostics again.
**Hint:** The Q-Q plot and the other diagnostics from the plot() function look fine to me!
*On doing a check for multicollinearity and autocorrelation we see that there is multicollinearity between temp and atemp (because they are essentially sourced from the same data and are dependent on each other) and there is a autocorrelation in the residuals.*
*To fix the multicollinearity problem we can remove the atemp variable from the model as it has a higher p-value than temp and a smaller coefficient than temp.*
*To fix the autocorrelation we introduce the variable date into the model. This adds granularity to the time series data and reduces the autocorrelation in the residuals.*
*On making these changes and running the diagnostics again we can see that both the issues are now resolved to a great extent and the model has also improved considerably in terms of the adjusted $R^2$ value.*

6) **Even more regression:** Run a simple linear regression to determine the effect of bad weather on COUNT when **none** of the other variables is included in the model.

a) Compare the coefficient with the corresponding value in **Q4**. Are they different? Why or why not?

*The coefficient for bad weather has changed from -1954.835 to -2780.95. This is because the other IVs in the previous model explains some of the variation in count along with the bad weather variable. But in the new model which has only bad weather, it must explain the variation in count to the maximum extent possible. Thus, the value of the coefficient increases.*

b) A consultant has indicated that bike use is affected differently by bad weather on weekdays versus non-weekdays, as people go to work on weekdays. How can you add this domain knowledge to the regression model you built in (a)? Why?

*We can add this domain knowledge by introducing an interaction term into the regression model. The interaction term captures the interaction that occurs when an independent variable has a different effect on the outcome depending on the values of another independent variable. Thus, an interaction term like badweather*weekdays can explain if the effect of badweather is different on weekdays and weekends.*

c) Run a new model with your addition from (b). Is this a better or worse model than your original model in (a)? How do you decide?

*This is not a better model than the one in (b). Though we see a slight increase in R2 due to new explanatory variables, the adjusted R2 goes down. This indicates that it is not a better model (considering the chance of it being better) Also, the p-value for the new IVs introduced are very high which means that they are not statistically significant. Thus, the model gives us no new or improved information.*

d) Using your model from (c),

i) interpret the average effect of bad weather on the COUNT depending on whether it is a weekday or not, and

*The COUNT decreases by 2631.1 on average in case of a bad weather day. But if the bad weather day is also a weekday the count further decreases by 15.9 (total by 2647) on average.*

ii) quantify the effect of bad weather on the COUNT in different scenarios (be sure to calculate *all* effect sizes for the **four alternatives (2x2)** here).

*[ In calculating the effects here, do **not** worry about the statistical significance]*

| ***Type of day*** | ***Calculation*** | ***Effect on count*** |
|---|---|---|
| *Bad weather weekday* | *-2637.1+185.3-201.2* | *Decreases by 15.9* |
| *Bad weather non weekday* | *-2637.1* | *Decreases by 2637.1* |
| *Non bad weather weekday* | *185.3* | *Increases by 185.3* |
| *Non bad weather non weekday* | *NA* | *No effect can be inferred* |

7) **Predictive analytics:** Follow the steps below to build two predictive models. Which model is a better choice for predictive analytics purposes? Why? Does your conclusion remain the same for explanatory analytics purposes? Please copy and paste the predictive and explanatory performance levels of both models into your response.

***Predictive Analytics performance comparison:***

*Model 1: fitOrg*

```{r}

performance <- metric_set(rmse, mae)
performance(resultsOrg, truth = count, estimate = predictedCount)

```

| .metric <chr> | .estimator <chr> | .estimate <dbl> |
|---|---|---|
| rmse | standard | 1024.3471 |
| mae | standard | 748.9409 |

*Model 2: fitNew*

```{r}

performance <- metric_set(rmse, mae)
performance(resultsNew, truth = count, estimate = predictedCount)

```

| .metric <chr> | .estimator <chr> | .estimate <dbl> |
|---|---|---|
| rmse | standard | 938.9830 |
| mae | standard | 680.3111 |

2 rows

*Based on the rmse and mae we can say that the second model is better at predicting the count of total bikes as the values have reduced while using the second model.*

***Explanatory Analytics performance comparison:***

*Model 1: fitOrg*

```
fitOrg <- lm(formula = count ~ date+month+badWeather+temp+humidity, data = dfbTrain)
summary(fitOrg)
```

```
Call:
lm(formula = count ~ date + month + badWeather + temp + humidity,
    data = dfbTrain)

Residuals:
    Min      1Q  Median      3Q     Max
-3319.1  -388.1    72.1   536.8  2809.8

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -82350.567   3048.984 -27.009  < 2e-16 ***
date                5.672      0.199  28.503  < 2e-16 ***
monthAugust      -548.911    207.817  -2.641  0.00849 **
monthDecember   -1643.685    198.086  -8.298 7.72e-16 ***
monthFebruary    -850.034    195.218  -4.354 1.58e-05 ***
monthJanuary     -896.729    203.145  -4.414 1.21e-05 ***
monthJuly        -641.012    218.851  -2.929  0.00354 **
monthJune        -159.285    199.164  -0.800  0.42418
monthMarch       -316.229    178.022  -1.776  0.07621 .
monthMay          278.596    184.085   1.513  0.13073
monthNovember    -754.154    192.446  -3.919 9.98e-05 ***
monthOctober       -4.250    182.434  -0.023  0.98142
monthSeptember    -87.751    194.118  -0.452  0.65141
badWeatherYes   -1760.774    233.484  -7.541 1.85e-13 ***
temp               89.657      9.986   8.978  < 2e-16 ***
humidity          -16.446      2.625  -6.264 7.39e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 860 on 569 degrees of freedom
Multiple R-squared:  0.8071,    Adjusted R-squared:  0.802
F-statistic: 158.7 on 15 and 569 DF,  p-value: < 2.2e-16
```

*Model 2: fitNew*

```r
fitNew <- lm(formula = count ~ date+month+badweather+temp+humidity+windspeed, data = dfbTrain)
summary(fitNew)
```

```
Call:
lm(formula = count ~ date + month + badWeather + temp + humidity +
    windspeed, data = dfbTrain)

Residuals:
    Min      1Q  Median      3Q     Max
-3467.1  -357.2    62.8   504.2  2577.3

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -79950.219   2971.882 -26.902  < 2e-16 ***
date                5.578      0.193  28.906  < 2e-16 ***
monthAugust      -634.084    201.374  -3.149 0.001726 **
monthDecember   -1777.023    192.660  -9.224  < 2e-16 ***
monthFebruary    -932.287    189.189  -4.928 1.09e-06 ***
monthJanuary     -974.839    196.796  -4.954 9.63e-07 ***
monthJuly        -808.984    213.232  -3.794 0.000164 ***
monthJune        -264.608    193.271  -1.369 0.171508
monthMarch       -346.404    172.188  -2.012 0.044716 *
monthMay          198.959    178.422   1.115 0.265278
monthNovember    -846.481    186.631  -4.536 7.01e-06 ***
monthOctober     -108.224    177.141  -0.611 0.541478
monthSeptember   -204.293    188.573  -1.083 0.279106
badWeatherYes   -1513.943    229.041  -6.610 8.87e-11 ***
temp               88.728      9.656   9.189  < 2e-16 ***
humidity          -21.211      2.646  -8.016 6.24e-15 ***
windspeed         -43.650      6.845  -6.377 3.74e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 831.5 on 568 degrees of freedom
Multiple R-squared:   0.82,     Adjusted R-squared:  0.8149
F-statistic: 161.7 on 16 and 568 DF,  p-value: < 2.2e-16
```

*Based on the values of adjusted $R^2$ and $R^2$ we can say that the second model is better at explaining the variance in the variable count as it has higher values of these parameters. Also, the p-value for the newly added windspeed is also very low which means that it is statistically significant. Also, the output of the anova test indicates that the explained variation is statistically significant.*

```r
# Anova test to compare models
anova(fitOrg, fitNew)
```

```
Analysis of Variance Table

Model 1: count ~ date + month + badWeather + temp + humidity
Model 2: count ~ date + month + badWeather + temp + humidity + windspeed
  Res.Df       RSS Df Sum of Sq     F    Pr(>F)
1    569 420787402
2    568 392671140  1  28116262 40.67 3.737e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

*Thus, the new model with windspeed is better than the old one for both explanatory and predictive purposes.*

a) Set the seed to **333** (Always set the seed and split your data in the same chunk!).
b) Split your data into two: 80% for the training set, and 20% for the test set
    i) Call the training set *dfbTrain* and the test set *dfbTest*
c) Build two different models, calculate, and compare performance.
    i) The first model will include the variables in **Q4 with any adjustments you may have made during the diagnostics tests in Q5** (call this one `fitOrg`). The second model will add WINDSPEED to this model -*Call it* `fitNew`.

**Hint:** Remember, every time you build a new model, there are three steps you need to follow to be able to calculate the predictive performance of the model:

i. Build the model and store it as *fitXxx*
ii. Create a new copy of the test dataset *dfbTest* by adding the predicted values as a new column. Name this new dataframe as *resultsXxx*
iii. Calculate the performance measures (RMSE and MAE) using the actual and predicted values stored in the results dataframe *resultsXxx*

-*You'll replace* Xxx *with the model names you use (*Org *&* New *are suggestions)*

You may have trouble with the metric_set() function if you used `modelr` in Q5 for the diagnostics test. Trouble means learning. If you run the following code, you can simply ask R to unload `modelr` and you'll be fine: **`detach('package:modelr', unload=TRUE)`**

8) **More predictive analytics:** In this final question, experiment with the time component. In a way, you will almost treat the data as a time series. We will cover time series data later, so this is just a little experiment. Taking into account date, you can't split your data randomly (well, evidently, you would not want to use future data to predict the past). Instead, you have to split your data by time. Start with `dfbOrg` and **use the variables you used in `fitOrg` from Q7c**. Split your data into training using the year "2011" data, and test using the "2012" data. Has the performance improved over the random split that assumed cross-sectional data (which you did in the previous questions)? Why do you think so? Split again by assigning 1.5 years of data starting from January 1st, 2011 to the training set and the remaining six months of data (the last six months) to the test set. Does this look any better? Discuss your findings.

*By splitting the datasets into training and test based on the year, we find that the predictive performance has decreased. This can be inferred from the increased values of rmse and mae. This maybe because in the earlier split when the data was assumed to be cross sectional the training set already contained some data from the future and thus was able to learn better and was able to predict the data in past from the test set much better. While when we split based on year, we only provide past data in the training set and the model now predicts the future data with only the knowledge of the past data and no knowledge of future data.*

*On splitting the data set by assigning 1.5 years of data to training we provide more information to the train the model and ask it to predict for a less period into the future. This enables the model to perform better and it does perform better than the previous results. This can be observed by the reduced values of rmse and mae.*

9) **Data-*informed* decision making:** Based on your quick analysis of the Capital Bikeshare data, what are some actions you would take if you were managing Capital Bikeshare's pricing and promotions? How do you think you would use your predictions?

*Based on the analysis we can know the approximate demand for bikes on a day. This information can be directly used to optimize the usage of the bikes and ensure availability. Also, the dependency identified for count on variables like bad weather and temperature are good outputs. We can provide promotions and cheaper pricing on days where the usage is expected to be low due to temperature/bad weather to incentivize more people to use the bikes. We can engage more casual riders by employing penetrative pricing strategies based on what day of the week it is and what is the weather like and how likely are they to use the bike. This can also help convert casual riders into registered users. More registered users are good as they have a higher tendency to use the bikes regularly. The predictions can also be useful towards optimizing the operations and maintenance. We can schedule the maintenance and repair of bikes on a day where the predicted demand is low thus reducing the impact on customer experience due to bike availability.*

10) **Data-*driven* solutions to "the" big challenge of bikeshare:** As shown in the visuals on the next page, Capital Bikeshare (like most other shared services) has an inherent challenge. In the morning, people use bikes to commute to their workplaces, leaving the bike racks empty in residential areas (this is called *rush-hour surge*). In the evening, the same phenomenon repeats in the opposite direction. Shared-service companies attempt to resolve this problem by *rebalancing*, which is basically moving bikes manually during the off-peak hours using trucks (which you may have seen on the streets) and other means. **Assuming you have access to all the data Capital Bikeshare collects, and you can collect new data**, what is a data-driven solution you would pursue? Be specific about the data you would collect (if any) and the analytics project/model you would use.

*Based on the demand predictions for the next day morning we can rebalance the bikes using trucks overnight to avoid shortage. During the day we can use smaller carts and vehicles to move the bikes around based on the demand predictions and the traffic status. We can also use the holiday and weekend predictions to direct our rebalancing efforts accordingly.*

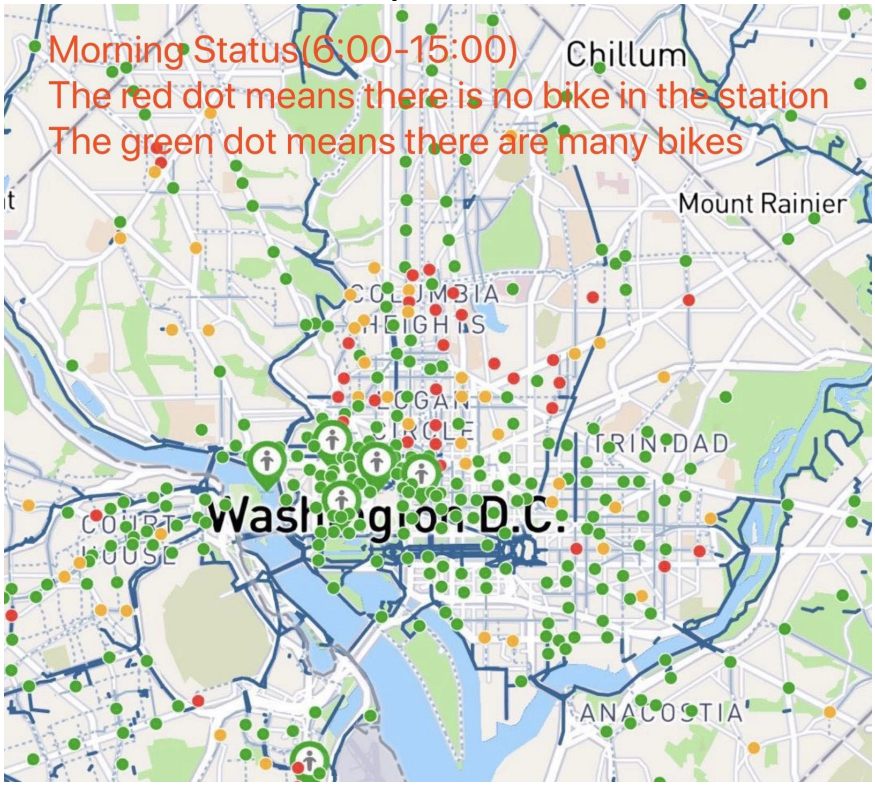
*Additionally, we can use a gamification technique (like the one being used by citibike in NY) to crowdsource the rebalancing efforts. This helps rebalance and at the same time increases the network of users for the bike sharing system.*

***Additional data to be collected:***

- *Demand at each station based on different hours in the day*
- *No of docks at each station*
- *No of bikes at each station*
- *Traffic data near each station*

*We can also use simulation models to work on this problem as it may be very difficult to explain or predict the variations in demand and supply using conventional regression models.*

**Morning -Green dots are stations with many bikes, red ones are those with no bikes:**




**Evening -Green dots are stations with many bikes, red ones are those with no bikes:**