Lab in Class (Poisson & Quasipoisson)

Enrique J. De La Hoz D.

19 de octubre de 2018

Ajustar un modelo para predecir el alquiler de bicicletas

En este ejercicio, se construirá un modelo para predecir el número de bicicletas alquiladas en una hora en función del clima, el tipo de día (vacaciones, día laborable o fin de semana) y la hora del día. El modelo de entrenamiento será el mes de Julio. Este dataframe fue la base de una competición en www.kaggle.com

- cnt: the number of bikes rented in that hour (the outcome)
- hr: the hour of the day (0-23, as a factor)
- holiday: TRUE/FALSE
- workingday: TRUE if neither a holiday nor a weekend, else FALSE
- weathersit: categorical, "Clear to partly cloudy"/"Light Precipitation"/"Misty"
- temp: normalized temperature in Celsius
- atemp: normalized "feeling" temperature in Celsius
- hum: normalized humidity
- windspeed: normalized windspeed
- instant: the time index – number of hours since beginning of data set (not a variable)
- mnth and yr: month and year indices (not variables)

Es importante determinar el parametro family = poisson or family = quasipoisson al ajustar el modelo en la función glm().

Dado que existan muchas variables predictoras, utilizar el vector creado para crear la formula a traves del comando paste().

The data frame bikesJuly is in the workspace. The names of the outcome variable and the input variables are also in the workspace as the variables outcome and vars respectively.

Model 1

- Crear la fórmula. (vars <- c("hr", "holiday" , "workingday", "weathersit", "temp", "atemp" ,"hum" ,"windspeed"))
- Calcule la media (mean()) y la varianza (var()) de bikesJuly$cnt.
- ¿Debes usar la regresión de poisson o quasipoisson?
- Use glm () para ajustar un modelo a los datos de bikesJuly: bike_model.
- Utilice glance() para ver las estadísticas de ajuste del modelo. Asigna la salida de glance() a la variable perf.
- Calcular el pseudo-R-cuadrado del modelo.

Predecir el alquiler de bicicletas en el mes siguiente

En este ejercicio, utilizará el modelo que construyó en el ejercicio anterior para hacer predicciones para el mes de agosto. El conjunto de datos bikesAugust tiene las mismas columnas que bikesJuly. Recuerde que debe especificar type = "response" con predict() cuando predice cnt a partir de un modelo de glm Poisson o Quasipoisson. Predecir el número de bicicletas por hora en los datos de bicicletas de agosto. Asignar las predicciones a la columna bikesAugust$pred.

Model 2

- Obtener el RMSE de las predicciones de los datos de agosto.
- Crear el gráfico de las predicciones vs valores reales
- ¿Alguna de las predicciones son negativas?

Visualize the Bike Rental Predictions

In the previous exercise, you visualized the bike model's predictions using the standard "outcome vs. prediction" scatter plot. Since the bike rental data is time series data, you might be interested in how the model performs as a function of time. In this exercise, you will compare the predictions and actual rentals on an hourly basis, for the first 14 days of August.

To create the plot you will use the function tidyr::gather() to consolidate the predicted and actual values from bikesAugust in a single column. gather() takes as arguments:

- The "wide" data frame to be gathered (implicit in a pipe)
- The name of the key column to be created - contains the names of the gathered columns.
- The name of the value column to be created - contains the values of the gathered columns.
- The names of the columns to be gathered into a single column.

En el ejercicio anterior, visualizó las predicciones del modelo de las bicicleta utilizando el diagrama de dispersión estándar de "resultados frente a predicción". Dado que los datos de alquiler de bicicletas son datos de series temporales, es posible que le interese saber cómo funciona el modelo en función del tiempo. En este ejercicio, comparará las predicciones y los alquileres reales por hora, durante los primeros 14 días de agosto.

Para crear la gráfica, utilizará la función tidyr::gather() para consolidar los valores reales y predichos de bikesAugust en una sola columna. gather() toma como argumentos:

- El dataset original (implícito en un %>%)
- El nombre de la columna clave que se creará - el cual contiene los nombres de las columnas reunidas.
- El nombre de la columna de valor que se creará - el cual contiene los valores de las columnas recopiladas.
- Los nombres de las columnas a ser reunidas en una sola columna.

```
# Plot predictions and cnt by date/time
___ %>%
  # set start to 0, convert unit to days
  mutate(instant = (___ - min(___))/24) %>%
  # gather cnt and pred into a value column
  gather(key = ___, value = ___, ___, ___) %>%
  filter(instant < ___) %>% # restric to first 14 days
  # plot value by instant
  ggplot(aes(x = ___, y = ___, color = valuetype, linetype = valuetype)) +
  geom_point() +
  geom_line() +
  scale_x_continuous("Day", breaks = 0:14, labels = 0:14) +
  scale_color_brewer(palette = "Dark2") +
  ggtitle("___")
```

Model 3

- Fill in the blanks to plot the predictions and actual counts by hour for the first 14 days of August.
- convert instant to be in day units, rather than hour
- gather() the cnt and pred columns into a column called value, with a key called valuetype.
- filter() for the first two weeks of August

- Plot value as a function of instant (day).
- Does the model see the general time patterns in bike rentals?

Modelo de crecimiento de soja con GAM

En este ejercicio, modelará el peso promedio de las hojas en una planta de soja en función del tiempo (después de la siembra). Como verá, la planta de soja no crece a un ritmo constante, sino que tiene un "impulso de crecimiento" que eventualmente disminuye. Por lo tanto, el peso de la hoja no está bien descrito por un modelo lineal.

Recuerde que puede designar qué variable desea modelar de forma no lineal en una fórmula con la función s ():

- y ~ s (x)

También recuerde que gam() del paquete mgcv tiene la interfaz gam (fórmula, familia, datos)

Para la regresión estándar, use family = gaussian (el valor predeterminado).

Los datos de entrenamiento de la soja, soybean_train se cargan en su espacio de trabajo. Tiene dos columnas: el peso del resultado y la variable Tiempo. A modo de comparación, el modelo lineal model.lin, que se ajustó con la fórmula peso ~ Tiempo, también se ha cargado en el área de trabajo.

Model 4

- Usar ggplot para graficar weight vs Time (Time en el eje x). ¿La relación parece lineal?
- Ajustar un modelo lineal (como lo hemos hecho antes). Llamarlo model.lin
- Cargar el paquete mgcv.
- Cree la fórmula fmla.gam para expresar el peso como una función no lineal de Time.
- Ajuste un modelo de aditivo generalizado en soybean_train usando fmla.gam.
- Realizar un summary()en el modelo lineal model.lin. ¿Cuál es el R2?
- Realizar un summary() en 'model.gam. Analizar el R2 no ajustado del modelo. ¿Cuál es el R2?. Comentar al respecto
- ¿Qué modelo parece encajar mejor con los datos de entrenamiento?
- Implementar la función plot() en model.gam para ver la relación derivada entre Tiempo y peso.