Test Paper Set 4

25 Marks

In this project we will be working with the UCI adult dataset. We will be attempting to predict if people in the data set belong in a certain class by salary, either making <=50k or >50k per year.

Instructions

Just complete the tasks outlined below.

Get the Data

Q1. Read in the adult_sal.csv file and set it to a data frame called adult.

Q2. Check the head of adult

Q3. Check the head,str, and summary of the data now.

Q4. Use table() to check out the frequency of the type_employer column.

Q5. How many Null values are there for type_employer? What are the two smallest groups?

Q6. Combine these two smallest groups into a single group called "Unemployed". There are lots of ways to do this, so feel free to get creative. Hint: It may be helpful to convert these objects into character data types (as.character() and then use sapply with a custom function)

Q7. What other columns are suitable for combining? Combine State and Local gov jobs into a category called SL-gov and combine self-employed jobs into a category called self-emp.

Q8. Use table() to look at the marital column

Q9. Check the country column using table()

Q10. Group these countries together however you see fit. You have flexibility here because there is no right/wrong way to do this, possibly group by continents. You should be able to reduce the number of groups here significantly though.

Q11. Use table() to confirm the groupings

Q12. Check the str() of adult again. Make sure any of the columns we changed have factor levels with factor()

Q13. Use ggplot2 to create a histogram of ages, colored by income.

Q14. Plot a histogram of hours worked per week

Q15. Rename the country column to region column to better reflect the factor levels.

Q16 Create a barplot of region with the fill color defined by income class. Optional: Figure out how rotate the x axis text for readability

Q17. Split the data into a train and test set using the caTools library

Q18. Train the model using glm() function

25 Marks

For this test you will be doing the Bike Sharing Demand Kaggle challenge!

Instructions

Just complete the tasks outlined below.

Get the Data

You can download the data or just use the supplied csv in the repository. The data has the following features:

- 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 Pallets + 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

Q1. Read in bikeshare.csv file and set it to a dataframe called bike.

Q2. Check the head of df

Q3. Create a scatter plot of count vs temp. Set a good alpha value.

Q4. Plot count versus datetime as a scatterplot with a color gradient based on temperature. You'll need to convert the datetime column into POSIXct before plotting

Q5. What is the correlation between temp and count?

Q6. Create a boxplot, with the y axis indicating count and the x axis begin a box for each season.

Q7. Create an "hour" column that takes the hour from the datetime column. You'll probably need to apply some function to the entire datetime column and reassign it. Hint:

```
time.stamp <- bike$datetime[4]

format(time.stamp, "%H")
```

Q8. Create the same plot for non working days.

Q9. Use lm() to build a model that predicts count based solely on the temp feature, name it temp.model

Q10. Get the summary of the temp.model

Q11. Finally build a model that attempts to predict count based off of the following features. Figure out if theres a way to not have to pass/write all these variables into the lm() function