Homework #2: The Data Product

PPOL 670

The *Data Product* is an output of the data science process. Data products can take on many forms, such as internet search engines, automated alerts for trending patterns, APIs for audio transcription and image detection, among other things. The creation of a data product starts with identifying a need, then creating a prototype that addresses that need, then testing the limits of that prototype. In this assignment, you will develop and test the limits of a prototypical idea using satellite imagery.

A little background. Low Earth Orbiting (LEO) satellites capture imagery of Earth everyday. In recent years, scientists have tried their hand at using satellite imagery to predict aggregate human behavior across the planet, whether its to predict *quarterly earnings of publicly traded firms*, estimate *wasted gas from flaring*, or guess *population in the developing world*. This is made possible by processing imagery into a form that can be used as the input features to predict a known quantity. In the most advanced cases, computer vision techniques are used to count individual features on the ground (e.g. cars in a parking lot). In simpler cases, aggregate data extracted from pixels correlate with certain behavior.

The task. A client is interested to see if metropolitan area employment patterns can be estimated using nighttime lights. For this to work, you'll need to see if the following regression specification can achieve some degree of reliability and accuracy for each metropolitan area i:

$$Y(\text{employment}_i) = f(\text{light distribution}_i)$$

You have been provided with 48 months of summary statistics of nighttime satellite imagery for each of 182 metropolitan areas in the file "homework2_data.Rda" (link to data in homework template); However, you are provided only 42 months of employment data. The employment data has been indexed relative to each metro area. The dataset contains 9 features:

- `GEOID`: Five character string. ID for each of the 182 metropolitan areas.
- `date`: Date in `YYYY-MM-DD` format.
- `emp`: Numeric. Mean-indexed employment estimate metro area (1.0 = mean)
- `year`: Integer. Year in `YYYY` format.
- `month`: Integer. Month.
- `avg_rad.50`: Numeric. Radiance at the 50% percentile.
- `avg_rad.sum`: Numeric. Sum of radiances in metro area.
- `avg_rad.mean`: Numeric. Average radiance in metro area.
- `complete`: Boolean. Indicates if record is available for training (FALSE = value is blank)

Note that radiances are floating point values with units in $\frac{\text{nanoWatts}}{cm^2 \times sr}$. All values have been multiplied by 1E9 (billion) to alleviate problems with small values. For the most part, the units are not important in this exercise.

The specific task. Knowing the best performance of a prediction model will give a sense of what's possible. If a model's best result is subpar, then the data product is likely to be a flop. Your job will be to develop regression model(s) using the `lm()` method and identify 10 cities where your regression model could be deployed. Feasibility is dependent on error and model fit (e.g. prediction errors and R-squared), thus your goal is to find cities where you could minimize error and maximize explained variance.

Tips.

- It's unlikely a single model will work well for all metro areas. You may consider training models for each city (`READ: Loop through each metro area`).
- Start by constructing a model for just one metro area. The training process usually involves:

- Create a test and training set from your data. Note that of the $n = 8736$ records in the dataset, only $n = 7644$ have employment records. The employment variable for the remaining $n = 1092$ records have been set aside for scoring your homework.
- Extract a metro area by `GEOID`.
- Partition your data into train and test (note that this is different than the 6 blank `emp` values). Maybe try a 70%-30% split or even k-folds cross validation.
- Try running a simple `lm()` model of your choosing using the training set and assign the model as an object (maybe name it `fit`). Use the `fit` object to score the training set and the test set.
- Calculate the MAPE for the training set and test set. Perhaps extract the R-squared (`summary(fit)$r.squared`). It might be a good idea to append/store your both model performance results and predictions using the "dummy, then append" paradigm.
- Once you've figured out one city, place the whole process into a loop to run through each city. Collect the training and test model performance and use those to identify your 10 best cities.
- Functions you'll likely need: `lm()`, `predict()`, `summary()`.

Submission.

- You will be given a `.R` script template and your submission will be a functioning R script. This script will be run by class instructors and the output of which will be graded automatically using a grading script. Save this `.R` file following this naming convention "firstname-lastname-hmwk2.R". For example: a file for Bill Clinton would be "bill-clinton-hmwk2.R".
- Your code should generate a data frame of predictions labeled `myPredictions`, which should contain the following fields labeled in this manner: `GEOID`, `date`, and `yhat` (your prediction). Choose your 10 best cities for grading and be sure that your `myPredictions` dataframe only contains those 10 cities (there should be about $n = 480$, of which $n = 60$ are records without the employment values to be used for scoring and $n = 420$ are your training set). A point will be deducted if your data frame and variables are not labeled exactly as stated above.
- Your grade will be based on the out-of-sample accuracy of your 10 cities as provided in the `myPredictions` dataframe that you will produce. As we have been provided `emp` values for 42 of 48 months in the sample but satellite data for all 48 months, you will be able to 'predict' or 'score' the 6 months of missing `emp` data. Again, you should label your the field with your predictions as `yhat` in the `myPredictions` dataframe.
- Scoring. 8 points for accuracy (see below). 2 points if your code runs without error.
- Accuracy metric. Your accuracy score will be based on the Mean Absolute Percentage Error (MAPE) of your out-of-sample predictions:

$$\text{MAPE} = 100 \times \frac{\sum |(\hat{y}_i / y_i) - 1|}{n}$$

  where $\hat{y}_i$ is a predicted value of y_i and n is the number of records in sample. The MAPE equation is provided in the homework template for your convenience. The following points will be awarded for performance in your out-of-sample predictions.
- $MAPE < 2.5$: 8 points
- $MAPE < 3$: 7 points
- $MAPE < 4$: 6 points
- $MAPE < 10$: 5 points
- Submit your code via Blackboard before class.