Linear Model Performance

Gwen Rino

April 20, 2018

Model Description

The linear model uses the features identified as having significant p-values in Pearson correlation tests against the target variable `total_cases` (see "dengue/src/EDA/EDA.R"). The features used in the model are:

- ndvi_nw
- reanalysis_air_temp_k
- reanalysis_avg_temp_k
- reanalysis_dew_point_temp_k
- reanalysis_max_air_temp_k
- reanalysis_min_air_temp_k
- reanalysis_relative_humidity_percent
- reanalysis_specific_humidity_g_per_kg
- station_avg_temp_c
- station_max_temp_c
- station_min_temp_c
- weekofyear

I ran the model with missing values imputed two ways, first by median values and second by k nearest neighbor values.

```
# MEDIAN VALUE IMPUTATION

# Create training and test sets
set.seed(555)
train_set.1 <- dengue.med.sm %>% sample_frac(0.7)
test_set.1 <- anti_join(dengue.med.sm, train_set.1)

# Fit model
lm_1 <- lm(total_cases ~ ., data = train_set.1)

# KNN VALUE IMPUTATION

# Create training and test sets
set.seed(777)
train_set.2 <- dengue.knn.sm %>% sample_frac(0.7)
test_set.2 <- anti_join(dengue.knn.sm, train_set.2)

# Fit model
lm_2 <- lm(total_cases ~ ., data = train_set.2)
```

Model Evaluation

Applying these models to a test set of a random 30% of the data, a comparison of the predicted to the actual number of cases yields a Mean Absolute Error (MAE) of 27.5 (median imputation model) and

26.4 (knn imputation model), both of which are actually worse than the Naive Model. It is noteworthy that by far the most influencial feature is `weekofyear`. I need to learn how to model and predict time series!

```
# Predictions from lm_1 (MEDIAN VALUE IMPUTATION)
predictions.lm_1 <- predict(lm_1, newdata = test_set.1)
# Find MAE of error (actual number of cases - predicted number of cases)
print(paste("lm_1 MAE = ",
            mean(abs(test_set.1$total_cases - predictions.lm_1))))
```

```
## [1] "lm_1 MAE =  27.5360203853305"
```

```
# Predictions from lm_2 (KNN VALUE IMPUTATION)
predictions.lm_2 <- predict(lm_2, newdata = test_set.2)
# Find MAE of error (actual number of cases - predicted number of cases)
print(paste("lm_2 MAE = ",
            mean(abs(test_set.2$total_cases - predictions.lm_2))))
```

```
## [1] "lm_2 MAE =  26.4072696284139"
```