~~~
---
title: "POGOH Bikeshare Trip Count Model"
output: html_document
date: "2024-07-06"
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

library(readr)
library(lubridate)
library(dplyr)
library(tidyverse)
library(TSstudio)
library(pacman)
library(ggplot2)
library(timetk)
# Helper packages
library(dplyr)      # for data wrangling
library(ggplot2)    # for awesome graphics
library(rsample)    # for creating validation splits
library(recipes)    # for feature engineering
# Modeling packages
library(caret)       # for fitting KNN models
library(corrplot)
library(olsrr)
library(caTools)
```

Get POGOH bikeshare trip data from tsv:

```{r get-trip-data}
df<-readr::read_tsv("pogoh_trips.tsv", show_col_types = FALSE)
df$date <- as.Date(df$start_time, '%d/%m/%y')
```

Aggregate trip data by date and station name:

```{r aggregate-dates}
by_date <- df %>%
  group_by(date, start_station_name) %>%
  summarize(rides = n()) %>%
  arrange(desc(rides))
```

Get POGOH station details from csv:

``` {r get-station-data}
stations <- read_csv('stations.csv', show_col_types = FALSE)

avg_by_station <- group_by(by_date, start_station_name) %>%
                      summarise(daily_rides = mean(rides))

avg_by_station_with_geo <- inner_join(x = avg_by_station, y = stations, by =
c("start_station_name" = "Name")) %>%
  select(., Id, start_station_name, daily_rides, Latitude, Longitude)
```


Get historical NOAA data for Pittsburgh from csv, downloaded from Visual Crossing Weather,
and merge it with the daily trip data:

```{r get-weather-data}
weather_df<-readr::read_csv("historical_weather.csv", show_col_types = FALSE)
~~~

```
merged_data <- merge(x = by_date, y = weather_df, by.x = 'date', by.y= 'datetime')
```

Scale the data and split it for test/train:

``` {r test-train-split}
merged_data<- merged_data %>%
                merge(x = ., y = avg_by_station_with_geo, by = "start_station_name")


scaled <- mutate(merged_data,across(where(is.numeric), scale))

sample <- sample.split(merged_data$Id, SplitRatio = 0.8)
train  <- subset(scaled, sample == TRUE)
test   <- subset(scaled, sample == FALSE)
```

Attempt to fit a linear model:

```{r fit-linear-model}
model <- lm(rides ~ feelslike + humidity + precip + windspeed + sunrise + visibility +
daily_rides, data = train)

ols_step_forward_p(model)

test$prediction = predict(model, test)
test$residual = test$rides - test$prediction

mean(abs(test$residual))
median(abs(test$residual))

ggplot() + geom_point(data = test, aes(x = rides, y = residual))
```

Given the shape of the residual plot, a linear model doesn't seem like a great model type for this data.

We will try fitting a kNN model, which takes the 5 most similar datapoints and averages them to predict the station trips for a particular day:

```{r fit-knn-model}
model <- knnreg(rides ~ daily_rides*feelslike + daily_rides*humidity + daily_rides*precip
+ daily_rides*windspeed + sunrise*daily_rides + daily_rides*uvindex, k = 5, data = train)

test$prediction = predict(model, test)
test$residual = test$rides - test$prediction

mean(abs(test$residual))
median(abs(test$residual))

ggplot() + geom_point(data = test, aes(x = rides, y = residual))
```

The residuals look more randomly distributed for this model than the linear model. The average error is also much lower. It's a better model type for predictions on this model.