River Flow Analysis

By Scott Burstein

Objectives

1. Outline steps I took to complete coding challenge.
2. Share findings and contextualize results.
3. What I learned and what I am working towards.

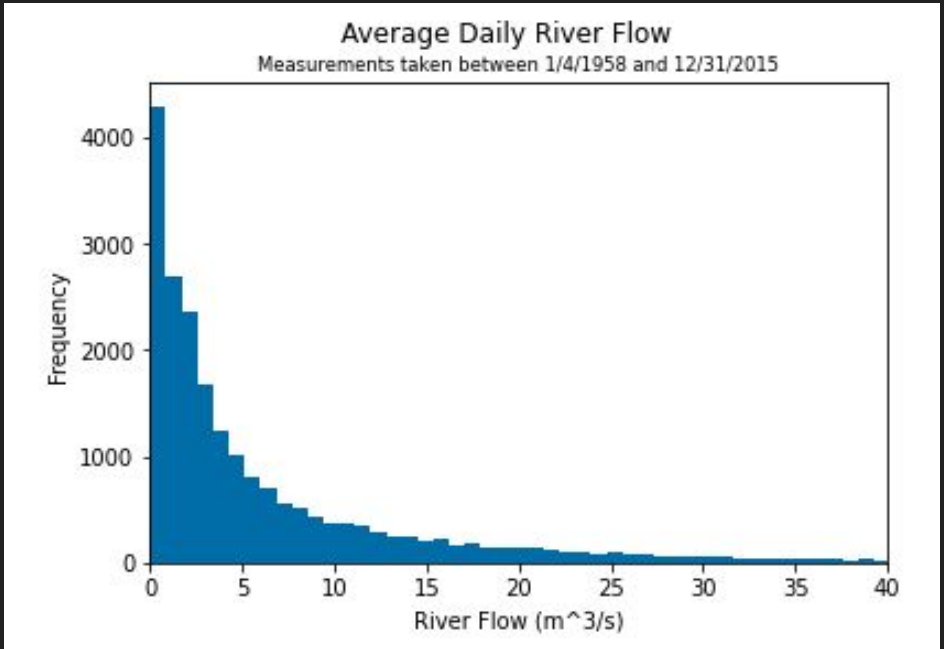
Structure of Analysis

1. Data Cleaning and Preparation (in R)
2. Summary Statistics and Data Transformation
3. Regression Analysis
4. Train ML Model

Future Steps:

5. Test Model
6. Apply to Client’s Production Scenario

Summary Statistics



Example histogram of daily river flow frequency

Important metrics for each variable:

- Median, Interquartile Range
- Outliers (min/max)
- Count (should all be the same)
- Data Types

The Challenge

Predict daily river flow from 9 surrounding temperature and precipitation measurement stations.

Potential applications include:

- Downstream agriculture ventures
- Water reservoirs
- Habitat restoration / Conservation efforts
- Academic research

Data Cleaning

1. Used R to un-pickle the data, rename variables and merge the two datasets.
2. Derived precipitation and temperature values corresponding to N days prior for each date (row):

Date	Flow (m^3/s)	Precipitation (mm)	Temperature (°C)
01/04/1958	4	4	4
01/05/1958	5	5	5
01/06/1958	6	6	6

Date	Flow (m^3/s)	Precip_1	Temp_1	Precip_2	Temp_2
01/06/1958	6	5	5	4	4

Function to create new columns:

```
def derive_nth_day_feature(df, feature, N):
    rows = df.shape[0]
    nth_prior_measurements = [None]*N + [df[feature][i-N] for i in range(N, rows)]
    col_name = "{}_{}".format(feature, N)
    df[col_name] = nth_prior_measurements
```

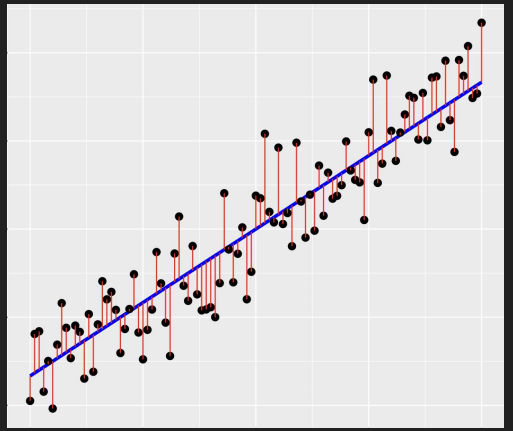
Linear Regression Model

1. Select relevant independent variables.

   For each river flow forecast, there are 27 precipitation and 27 temperature predictor variables.

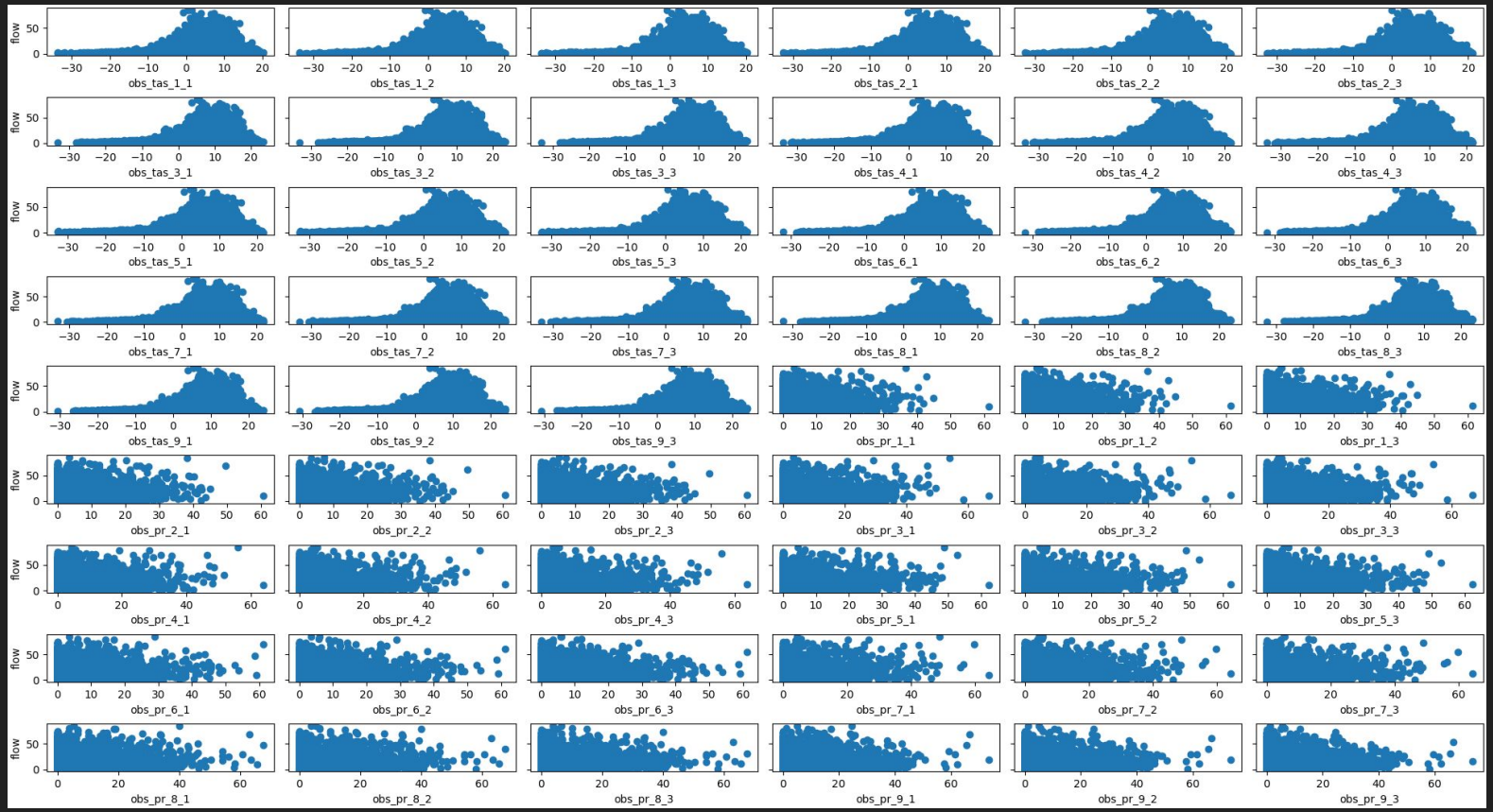
2. Assess Pearson correlation coefficients.

   All 54 correlation coefficients have range [0.19 - 0.36]



Linear Regression (cont.)

Predictor Variable Relationships with Response Flow Variable



Training a ML Model

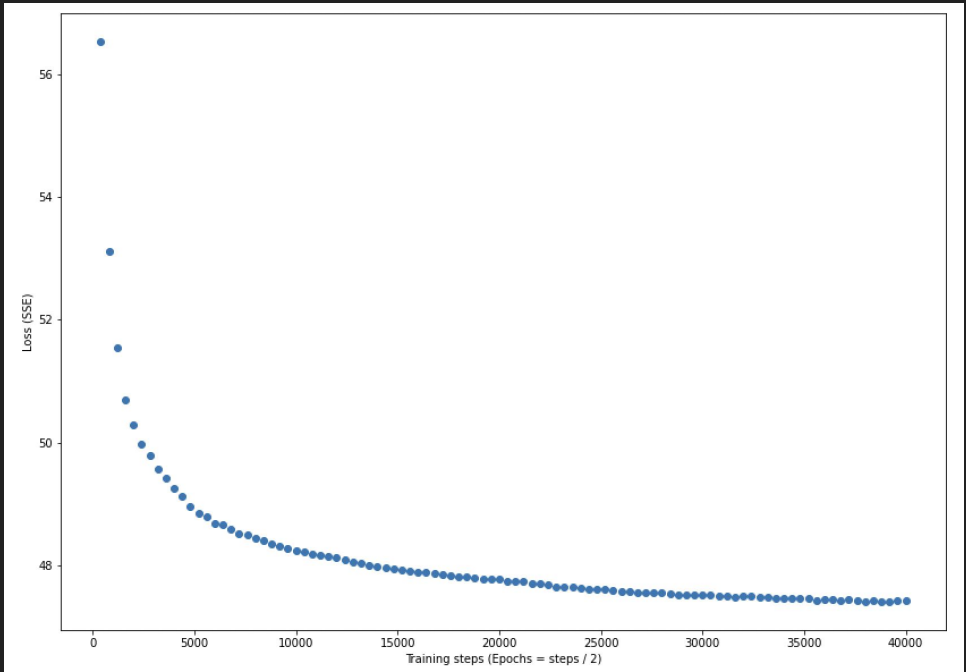
Using supervised machine learning methods to create a DNNRegressor:

1. More data cleaning!
2. Split data into (80%) training set, (10%) testing set, and (10%) validation set.
3. Instantiate neural network with 2 hidden layers.
4. Define reusable function to manage data input.
5. Export model.

Explained Variance: 0.38 **Mean Absolute Error:** 4.59 m^3/s flow **Median Absolute Error:** 2.41 m^3/s flow

Sources of Error

Loss SSE vs. Training Steps Plot



Bias?

Variance?

Indicates that the model was not overfitted since the evaluation losses never exhibit a significant change in direction toward an increasing value.

Future Steps

1. Create a more robust linear regression model.
2. Extract tensors from TensorFlow object.
3. Back-test data to assess model efficacy.
4. Deliver usable product to client.
5. Receive feedback.

What I am working towards:

1. Gain more hands-on ML experience
2. Develop stronger foundation in regression analysis
3. Complete Codecademy data science path / other online courses
4. Specialize in climate analytics / environmental data science applications

Thank You!

Questions?