Introduction to Linear Regression: Takeaways

Syntax

- Create dummy variables from a categorical column in a DataFrame:

```
df_with_dummies = pd.get_dummies(df, columns=["categorical_col"])
```

- Using matrix multiplication to solve for the coefficients:

```
XX_inv = np.linalg.inv(np.matmul(np.transpose(X), X))
XY = np.matmul(np.transpose(X), y)
beta_hat = np.matmul(XX_inv, XY)
```

Concepts

- **Regression**: a relationship between some predictors X and a numerical outcome Y.
- **Linear Regression**: a model that describes the relationship between outcome and predictors as a linear combination.
- **Cost Function**: a function that defines how close the model predictions are to the observed outcome — it summarizes the total amount of proximity over all of the observations.
  - In the case of linear regression, it's most common to use the **sum of squared error** as the cost function.
- Linear regression is "linear" because it describes the outcome as a *linear combination* of the predictors, plus some error.
- The error portion of a linear regression describes what the predictors can't explain.
- There are two types of predictors: **numerical** and **categorical**.
- Categorical predictors need to be split into $K - 1$ binary columns, where K is the number of categories in the original column.
- The coefficients of a linear regression model can be derived from the data in a closed-form solution:
  $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$

Resources

- Automobile Data Set
- Matrix Calculus
- The Normal Equation And Matrix Calculus
- NumPy's matmul() function
- NumPy's transpose() function
- NumPy's linalg.inv() function