Michael Baldwin

Josh Engelsma

Adam Terwilliger

January 19, 2016

CIS 678 – Machine Learning

Project 1

Abstract

To demonstrate our programming abilities as an introductory project in CIS 678 – Machine Learning, we explored book sales data with basic regression analysis. Using Python, we were able to generate a simple linear regression model that predicted the amount of downloads using the amount of hours since the book was released. Additionally, we explored polynomial regression as we noted a second and third order model fits the model more appropriately. To visualize our results, we utilized matplotlib to bin the data by time of day, day of week, and day of month; as well as, D3.js to generate the scatterplot of the dataset with the three distinct lines representing unique order polynomial models.

Implementation details

Our backend program is written in Python 3.0 and frontend in Javascript/CSS with the Data Drive Documents (D3) package/libraries installed. These programs were executed locally on each member's respective Macbook Pro (2012).

Summary of Problem

Linear regression is the foundational method when first understanding supervised learning. In supervised learning, we look to characterize a set of variables or model that can predict a known response variable. In this instance, we have just a single explanatory variable (simple linear regression), time in hours since the release date, and we are predicting amount of downloads as the response. A simple linear regression line can be represented with a slope and intercept, as seen in Figure 1.

$$
\begin{aligned}
\hat{\beta} &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\
&= \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{j=1}^{n} y_j}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2} \\
&= \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2} = \frac{\operatorname{Cov}[x, y]}{\operatorname{Var}[x]} \\
&= r_{xy}\frac{s_y}{s_x}, \\
\hat{\alpha} &= \bar{y} - \hat{\beta}\,\bar{x},
\end{aligned}
$$

Figure 1. Proof of Slope and Intercept for Simple Linear Regression

Results

We note that in looking to predict *y*, we first must characterize *x*, through its *covariance* with *y* and its *variance*. As such, we have calculated the slope and intercept using this method by hand, as well as, with a matrix solver in Python that allows for us to apply higher order polynomial regression models; seen in the sample output from Figure 2.

```
Linear Regression calculated by hand:

y = 2.61928481625x + 983.227482833


Linear Regression using Matrix Solver:
y = 2.61928481625x + 985.846767649


Quadratic Regression using Matrix Solver:
y = 0.0106160902962x^2 + -5.29438577386x + 1974.18190474


Cubic Regression using Matrix Solver:
y = 3.07350980198e-05x^3 + -0.0237269023042x^2 + 4.95214473622x + 1335.36380284
```

Figure 2. Sample Output from Python Regression program

We can observe the performance of our models through R^2 values in Table 1, and visually in Figure 3. As we uncovered, the linear fit to the data explains only approximately 42% of the variation in amount of downloads. However, as it appears in Figure 3, we see that the quadratic and cubic fits to the data show great promise explaining approximately 67% and 75% of the variation in amount of downloads; respectively.

Model	R-Squared
Linear	0.41824
Quadratic	0.67230
Cubic	0.74762

Table 1. R-Squared values for First, Second, and Third order models

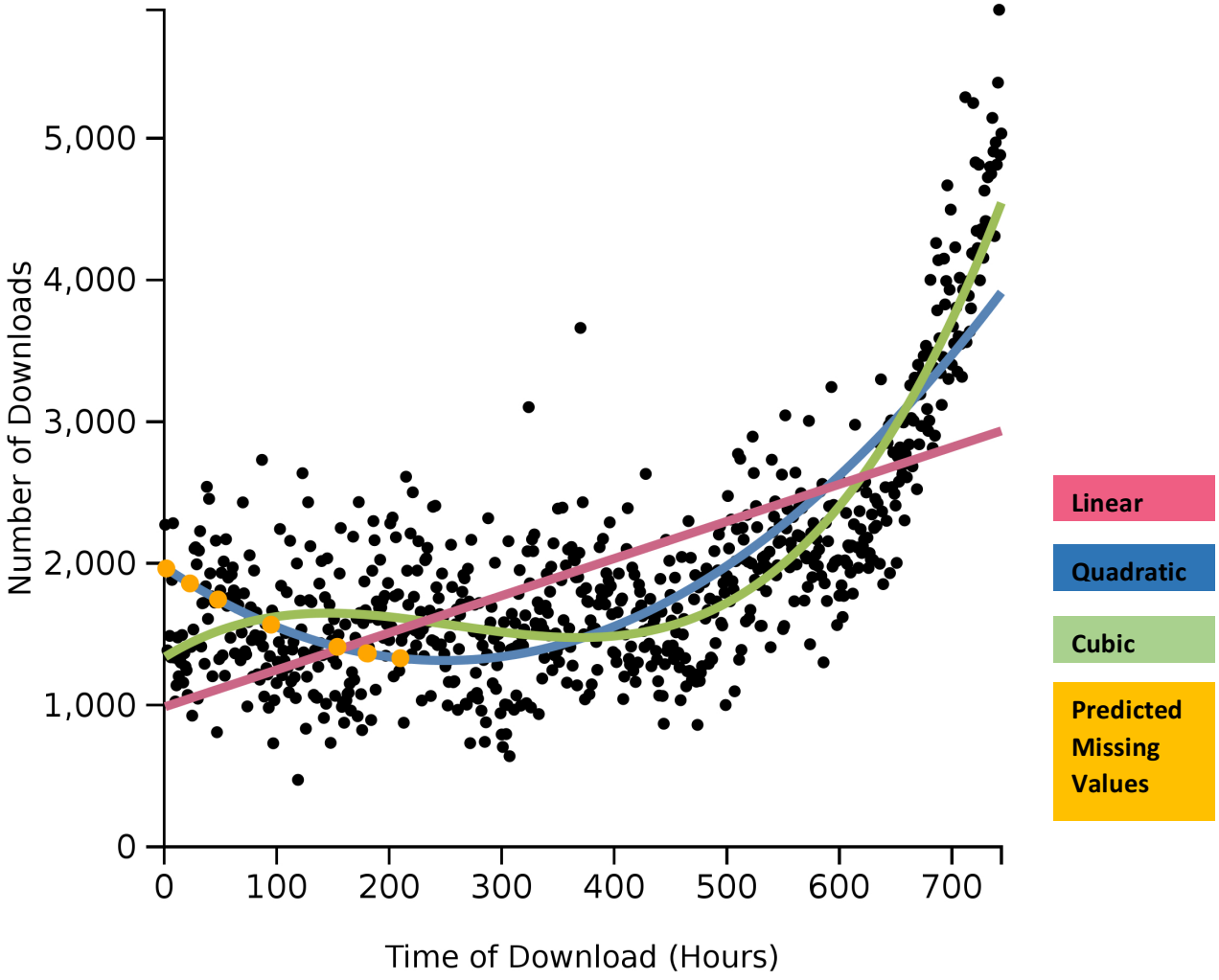

Figure 3. Scatterplot of Time vs. Downloads with Models and Missing Value Imputation

Discussion

One interesting feature we find in Figure 3 is missing value imputation. In our original dataset, 7 of the 744 total data points were missing. As such, we imputed these values with the quadratic predicted values for number of downloads. We choose this model for the imputation over the cubic with the principle of balancing model simplicity with the amount of variation explained.

We began to extract features of the dataset in Time of Day, Day of Week, and Day of Month as seen in Figures 4, 5, and 6; respectively.

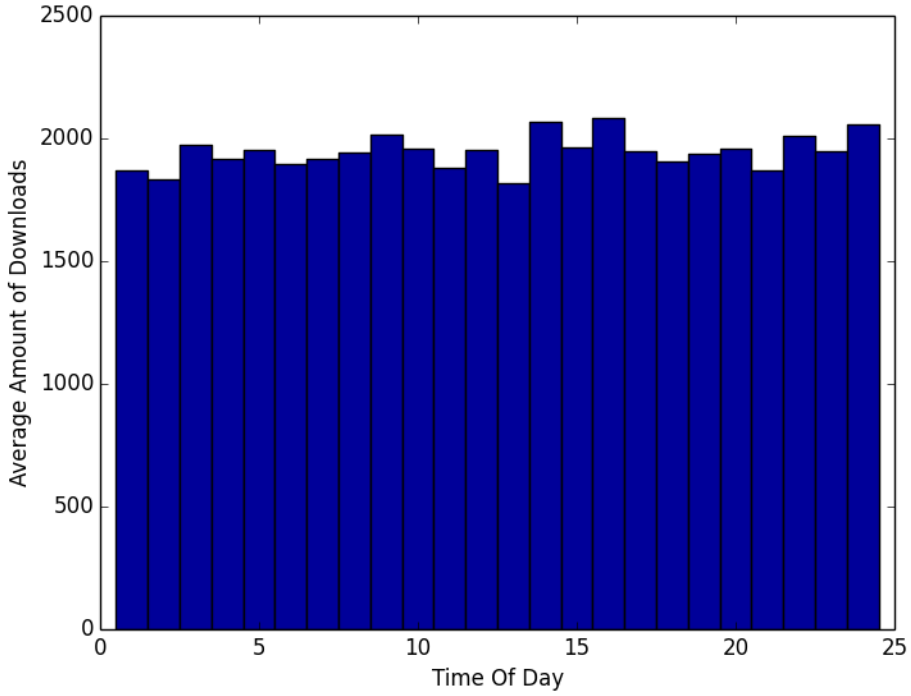

Figure 4. Histogram of Time of Day vs. Average Amount of Downloads

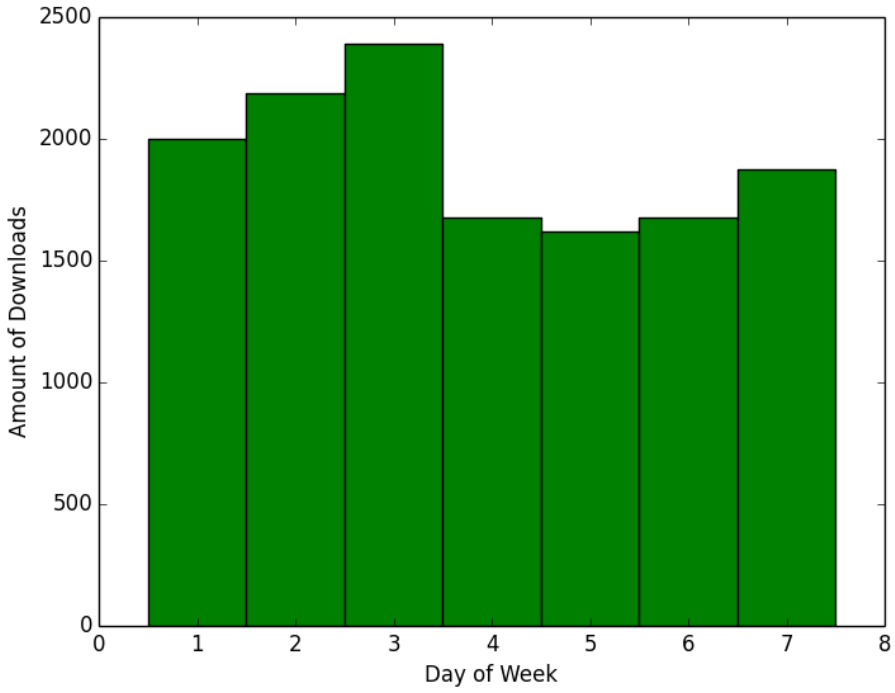

Figure 5. Histogram of Day of Week vs. Average Amount of Downloads

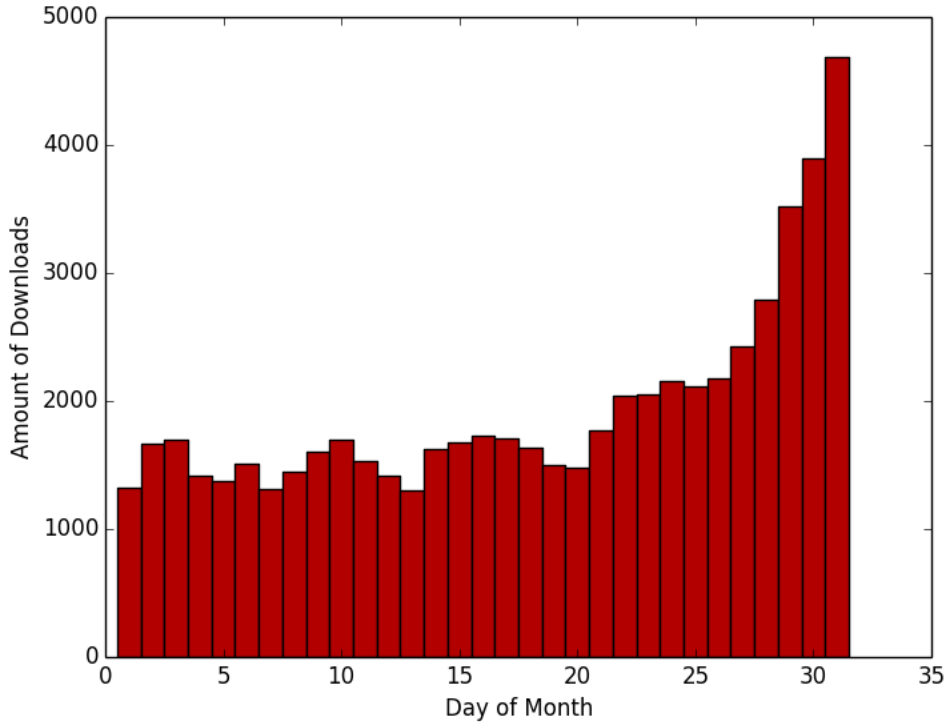

Figure 6. Histogram of Day of Month vs. Average Amount of Downloads

Figure 4 provides little to no additional information, as we can infer that Time of Day would not be a valuable feature in a multiple regression model due to equal variance throughout the day with only a slight peak around 4/5 pm. Additionally, Figure 5 shows a great peak on days 1, 2, and 3 (we did not have day markers i.e. Sunday, Monday, etc.). However, this is a result of having 1 additional day contributing to the average downloads for the day, with the first three days showing the effect of the rise in downloads at the end of the month, as seen in Figure 6.

Our final note is with regards to avoiding overfitting the model as we may be encouraged by a higher order model explaining more of the variation in downloads; however, in future work, we should apply appropriate machine learning techniques of training and test sets to avoid this issue.

Future Work

Due to time constraints, we were not able to implement the histograms for each respective feature in D3.js. We value these data-driven libraries as we appreciate the customization and future forms of interactivity and animation that are crucial in exploratory data analysis.

We would also be interested in separating our data into training and test sets to better validate our model.

Credits

Editors

- [Style Guide](https://www.python.org/dev/peps/pep-0008/)
- [Vim](http://vim.wikia.com/wiki/Converting_tabs_to_spaces)
- [Sublime Text](https://www.sublimetext.com/docs/2/indentation.html)
- [Atom Editor](https://atom.io/packages/tabs-to-spaces)

Python
- [Try Except, Continue For Loop](http://stackoverflow.com/questions/4799974/continue-on-except-of-a-try-block-in-python)
- [Calculating Polynomial Regressions](http://hotmath.com/hotmath_help/topics/quadratic-regression.html)
-
[Numpy](http://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.solve.html#numpy.linalg.solve)

JavaScript
- [Window Onload](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/onload)
- [Global Variable Across Files](http://stackoverflow.com/questions/3244361/can-i-access-variables-from-another-file)

D3 Visualizations
- [D3 Intro](http://d3js.org/#introduction)
- [D3 Fundamentals](http://alignedleft.com/tutorials/d3/fundamentals)
- [D3 Scatterplot Tutorial](https://www.oreilly.com/learning/making-a-scatterplot-with-d3-js)
- [D3 Trend Line](http://bl.ocks.org/benvandyke/8459843)
- [D3 CSV Import](https://github.com/mbostock/d3/wiki/CSV)