Comparative Analysis of Regression Models for Personal Finance Predictions

CSI 4106 – Introduction to Artificial Intelligence
Fall 2018
Project Coordinator: Yazan Otoum
Date: December 4, 2018

Group Number: 22
Student Name: Dominic Roy-Stang
Student Number: 7483706

CONTEXT

The objective of this research project was to evaluate the feasibility of predicting an individual's personal finances based on past results with a greater accuracy than a simple linear regression. This sort of result could, in practice, be used to help people predict their retirement date, which would in turn, help them assess if they are living within their means.

SUMMARY OF APPROACHES

To accomplish these goals, I first started by establishing a linear regression predictor as a baseline because of its simplicity. Then, I generated two separate models: a *Time Series Regression (using linear regressions)*, and a *Gaussian Process Regression (with Exp-Sine-Squared kernel)*. These two predictors were then compared to the linear regression to see if more complex models can outperform a simple linear regression on this specific task. Then, the three models will be compared to each other to be ranked based on their performance on the given data set.

These are three fairly different methods that can be used to draw regression lines, and hence, could be used to model financial data. However, they work in different ways.

In simple terms, linear regression simply draws a best-fit linear line through the data. A gaussian process regression can be thought of as a non-linear version of a linear regression. It is typically used to predict data based on similar previously-seen examples. A time-series, on the other hand, will use previous data points, and run a regression to find the next point. Then it will repeat the process using the new point to find the next point. This process is repeated so the time-series builds upon itself, and then a line can be drawn through the generated points.

Each of these approaches will be explored in more detail later in the *details of the approaches* section of this report.

DATASETS

Finding appropriate datasets for this research was a bigger issue than expected. Due to the fact that banks typically abide to strict privacy rules, it was difficult to find personal banking data that would be suitable for analysis. The best dataset I was able to find was data from a bank in the Czech Republic, which provided anonymized transactions for over 10,000 accounts.

The only issue with the dataset is that it only contains data spanning a period of approximately 6 years for each account. This is not optimal for long-term financial predictions, and this will be further discussed later in the report.

DETAILS OF THE APPROACHES

Summaries

Linear Regression

A linear regression is probably the simplest form of regression. It can be thought of as drawing a line of the form *y* = *m***x* + *b* that minimizes the errors on the training set. In fact, in its implementation in scikit-learn, the linear regression runs ordinary least squares from scipy, and wraps it as a predictor object.

Time Series

A Time Series is distinct from other types of regressions because it has a clear temporal ordering. While a regression may normally find a relation between the date and the balance, it wouldn’t necessarily find a relation that directly depends on the balance at the points prior to it. This is where a time-series comes in useful, and this is why they are typically used for for stock market predictions.

Gaussian Process Regression

A gaussian process regression takes a probability distribution of functions that may describe the dataset. To be more precise, it consists of two parts. The first is a mean function, which is the mean of several functions that fit the data. The second is a covariance function that is determined by the kernel. The kernel is a function that describes how much impact one data point has on another, which affects the smoothness of the functions in the distribution. This set of functions together determine the mean and the confidence interval.

Implementation

The implementation for this project can be found at the following link:
https://github.com/DominicRoyStang/finance-predictor

Instructions regarding required software, as well as instructions on running the code are listed in the README.md file at the root of the project. Sample datasets can be found under the *datasets* folder, and all the source code is located under the *src* folder.

Feature Engineering

Since the main goal was long-term financial planning, I concluded that a good model would take a *date* as the input, and output a *net worth* or *balance* at that the specified input date. Hence, I created a module that I called *feature engineering* that would take data from csv files of different formats, and convert them to the desired *date, net worth* format.

The *feature engineering* module supports data with several different formats, notably preformatted, Mint™, and Czech. Since data from the Czech bank provided the information of all 10,000 accounts in one massive CSV file, I had to create a function that could take an aggregate csv, and split it into one CSV file for each account. The *split_dataset.py* file under the *src* folder uses this feature of the module.

Model Selection

The most important section for this project is model selection script. The script runs several types of regressions against a given dataset. For each of the three regression models, it splits the data into a training set and a testing set, runs the regression, and outputs the score of that model on the test set. If a folder is provided, every file inside that folder will be run, and in the end, the script will output the average score of each model on all the datasets in the folder.

If you look more closely at the *model_selection.py* script, you will notice that there are functions for other regression types. Indeed, to I initially planned to use a support vector regression instead of a time series, but due to several reasons, I decided that a time series would be more interesting for the purpose of financial predictions. I also looked at other methods such as lasso, all of which have been left in for those interested.

Finance Predictor

Finally, the *finance_predictor.py* script is similar to the model_selection.py script, except that it can only run one dataset (no folders of datasets), and includes graphing of input data and predictions. This script tries to run and visualize long term predictions on one individual's spending data. Please note that this script is not as flexible as it could be, and mostly only works for the provided *preformatted.csv* data set because of some hard-coded time frames used for predictions.

Evaluation

To evaluate the different models, I was initially using the root mean squared (RMS) value because it gave an easy to understand metric. Basically, it would say "your predictions on the test set were off by $<amount> on average". While RMS is great for understanding how wrong a model is on an individual's data, it is actually a poor measure when looking at the financial data of several users because financial data is not normalized.

Hence, I decided to use a metric based on variance for scoring: R². The maximum value possible for an R² score is 1, which would occur if the model perfectly predicted all the balance values in the test set. A score of 0 occurs if your average score is as good as a constant function set to the mean value in the test set. Finally, a negative score occurs if the model performs worse than the aforementioned constant mean function.

RESULT ANALYSIS

Running the model selection script on all the data from the Czech bank is too slow on my computer, so I decided to run it a subset of 1000 randomly-selected accounts. The table below highlights the results from the tests.

Model	**Average R² Score**
Linear Regression	-0.77
Time Series	-177.5
Gaussian Process Regression	-4.6

Table 1: R² scores for each of the evaluated models.

As indicated above, linear regression had the best score with an R² of -0.77, followed by a Gaussian Process Regression with a score of -4.6, and Time Series with a score of -177.5. The R² values were calculated on the test set of the data, which was the last 25% of transactions of each dataset. The preceding 75% of transactions was used as a training set. Running a finance prediction on my personal transaction data yielded the following results.

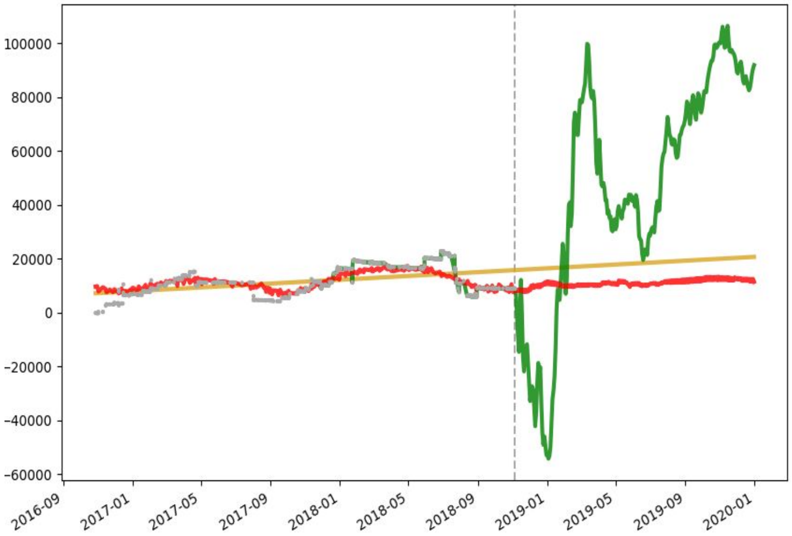

Figure 1: 2-year predictions on my personal data

The training data is shown to the left of the vertical dashed gray line, and the predicted values over the next year are shown to the right of of the vertical line. The linear regression is represented by the yellow line, the time series regression is represented by the green line, and the gaussian process regression is represented by the red line.

When looking at graph visualizations such as the one seen in figure 1, it became apparent that the kernel used by the gaussian process regression could be tweaked because the values were not as “smooth” as one would expect. Additionally, the time series regression has wild fluctuations, and could likely use some tweaking. However, these are the results at the current state after having spent well over 35 hours working on this project.

To observe how well each of the three regressions perform on long-term data, I decided to look at predictions and find out when I would be able to retire. Assuming I retire at a net worth of one million dollars, the linear regression predicted a retirement date of 2284, the time series predicted retirement in the year 2050, and the gaussian process regression kept fluctuating up and down, never reaching the target value of one million.

Of course, since we do not have long-term data, more research is needed to be able to draw any final conclusions, but it does seem that the time series gives the most realistic (although fairly optimistic) prediction despite having the worst performance on short-term predictions.

CONCLUSION

For short term predictions, my data suggests that a simple linear regression is the best predictor for personal net worth. The time-series regression and gaussian process regression could very likely be tweaked to achieve better predictions. However, performance on long-term predictions is still a question that remains unresolved.

The lack of available complete datasets prevented the original goal of evaluating performance on long-term predictions, but the work performed for this short study does pave the road for future research into the topic of personal finance predictions by highlighting the requirements that would be needed to achieve such goals.

REFERENCES

Ebden, Mark. *Gaussian Processes for Regression: A Quick Introduction*. Aug. 2008, ftp://ftp.tuebingen.mpg.de/pub/ebio/chrisd/GPtutorial.pdf.

“What Is Linear Regression.” *Statistics Solutions*, 2013, www.statisticssolutions.com/what-is-linear-regression/.

“Linear Regression Documentation.” *Scikit-Learn Documentation*, scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html.

“Gaussian Process Regressor Documentation.” *Scikit-Learn Documentation*, scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.GaussianProcessRegressor.html.

“Gaussian Process Regression Models.” *MathWorks*, www.mathworks.com/help/stats/gaussian-process-regression-models.html.

What is Gaussian Process Regression?, YouTube, 8 Nov. 2017, www.youtube.com/watch?v=mqOdIYxxNCs.

“What Is R Squared And Negative R Squared.” *Fairly Nerdy*, www.fairlynerdy.com/what-is-r-squared/.

Hamilton, Mark. “Tseries Package Repository.” *Github*, github.com/mhamilton723/tseries.

Vergard, F. “How (Not) to Use Machine Learning for Time Series Forecasting: Avoiding the Pitfalls.” *Towards Data Science*, towardsdatascience.com/how-not-to-use-machine-learning-for-time-series-forecasting-avoiding-the-pitfalls-19f9d7adf424.