Capstone Project

Machine Learning Engineer Nanodegree

Joshua Cook
September 2016

I. Definition

Project Overview

The purpose of this project is to solve a Kaggle competition using neural networks of varying complexities. The competition in question is sponsored by Red Hat. Given situational and customer information the goal is to protect customer behavior. This project will use these two data sources and neural network reinforcement learning techniques to prepare an algorithm capable of predicting outcomes against a third situational data source.

Data is provided in the form of two separate data sets encoded as CSV:

- `people.csv`
- `act_train.csv`

The third set is `act_test.csv`.

The action (`act_train.csv`) table makes reference to the people (`people.csv`) table. Beyond this, the sets has been scrubbed of any domain specific knowledge. Rather attributes are referred to generically as `char_1`, `char_2`, etc. As such the competition presents an interesting challenge in which domain knowledge is completely useless. The competition is in essence a “pure machine learning problem”.

Problem Statement

In this Kaggle competition, Red Hat seeks an optimal algorithm for using information about a given action and information about a customer to predict the customer’s behavior with regard to that action. A completed product will take the form of a csv with two items per row - an `action_id` from the test set, and a predicted outcome from the set $(0, 1)$.

I am going to take the following approach to completing this task:

1. Seed a PostgreSQL database with the three csv files.
2. One-Hot Encode the data and write this data to a separate table

3. Pull rows from the One-Hot Encoded table to pass through a Reinforcement Learner
4. Create, Update, and Store the parameters of the Reinforcement Learner
5. Use the Reinforcement Learner to run a set of predictions on Test Data.

Note that while the Kaggle Challenge includes a set of test-data, for the purposes of this study we will be holding a separate test set aside that we are able to run our own local accuracy metrics.

Metrics

The quality of a solution to this task will be measured using the following test error metric

$$\mathrm{Ave}(I(y_i \neq \hat{y}_i))$$

Here, I is an indicator function which yields 0 if the predicted outcome ($\hat{y}_i$) matches the actual outcome (y_i). While the size of the dataset (over 2 million rows in the action set) makes this problem atypical, it is at the end of the day, a binary classifcation problem. As such this simple metric is sufficient to measure our accuracy.

II. Analysis

(approx. 2-4 pages)

Data Exploration

The data to be used here consists of three datasets:

- `people.csv` sample
- `act_train.csv` sample
- `act_test.csv` sample

We will do the following to analyze the datasets.

1. define the basic structure - rows, columns, data types
2. identify unique labels for each column and the counts for each label
3. identify duplicate records, if they exist
4. search for NULL data
5. run `AGGREGATE HISTOGRAM`

The structure of these datasets has been identified in order to prepare tables in our database to hold the data. The full structure can be viewed in the seeding file here. Samples

To summarize the data, we can say this about each table.

people

Basic attribute query against people table `SELECT COUNT (DISTINCT #COLUMN#), MAX(#COLUMN#), MIN(#COLUMN#), AVERAGE(#COLUMN#) from people`

|: column :|: count :|: max :|: min :|: mean :|: notes :| |———-|———|-|-|-|-| | people_id| 189118 | | | |textprimary key | |ppl_date| 1196 | | | |timestamp| |ppl_group_1| 34224 | | | |text| |ppl_char_1| 2 |type 2 |type 1 | |text| |ppl_char_2| 3 |type 3 |type 1 | |text| |ppl_char_3| 43 |type 9 |type 1 | |text| |ppl_char_4| 25 |type 9 |type 1 | |text| |ppl_char_5| 9 |type 9 |type 1 | |text| |ppl_char_6| 7 |type 7 |type 1 | |text| |ppl_char_7| 25 |type 9 |type 1 | |text| |ppl_char_8| 8 |type 8 |type 1 | |text| |ppl_char_9| 9 |type 9 |type 1 | |text| |ppl_char_10| 2 | | | 0.25094385515921276663 |boolean| |ppl_char_11| 2 | | | 0.21550037542698209583 |boolean| |ppl_char_12| 2 | | | 0.24034729639695851267 |boolean| |ppl_char_13| 2 | | | 0.36507365771634640806 |boolean| |ppl_char_14| 2 | | | 0.25980075931429054876 |boolean| |ppl_char_15| 2 | | | 0.26951427151302361489 |boolean| |ppl_char_16| 2 | | | 0.28207785615330111359 |boolean| |ppl_char_17| 2 | | | 0.29196057487917596421 |boolean| |ppl_char_18| 2 | | | 0.18762360008037310039 |boolean| |ppl_char_19| 2 | | | 0.28465825569221332713 |boolean| |ppl_char_20| 2 | | | 0.22911621315792256686 |boolean| |ppl_char_21| 2 | | | 0.28503368267430916148 |boolean| |ppl_char_22| 2 | | | 0.29105637749976205332 |boolean| |ppl_char_23| 2 | | | 0.29849088928605420954 |boolean| |ppl_char_24| 2 | | | 0.19044723400205162914 |boolean| |ppl_char_25| 2 | | | 0.32778476929747565012 |boolean| |ppl_char_26| 2 | | | 0.16702799310483401897 |boolean| |ppl_char_27| 2 | | | 0.23805243287259806047 |boolean| |ppl_char_28| 2 | | | 0.28888841887075793949 |boolean| |ppl_char_29| 2 | | | 0.16834463139415602957 |boolean| |ppl_char_30| 2 | | | 0.20693429499042925581 |boolean| |ppl_char_31| 2 | | | 0.27858268382702862763 |boolean| |ppl_char_32| 2 | | | 0.28490677777895282311 |boolean| |ppl_char_33| 2 | | | 0.21784282828710117493 |boolean| |ppl_char_34| 2 | | | 0.35648113875992766421 |boolean| |ppl_char_35| 2 | | | 0.21027612390147949957 |boolean| |ppl_char_36| 2 | | | 0.34370075825674975412 |boolean| |ppl_char_37| 2 | | | 0.28545141128819044195 |boolean| |ppl_char_38| 101 |100.0 |0.0 |50.3273987669074 |real‘ |

`action`

Basic attribute query against `action` table `SELECT COUNT (DISTINCT #COLUMN#), MAX(#COLUMN#), MIN(#COLUMN#), AVERAGE(#COLUMN#) from action`

: column :	: count :	: max :	: min :	: mean :	: notes :
`people_id`	189118				`text` foreign key on `people`
`act_id`	2695978				`text` primary key
`act_date`	411				`timestamp`
`act_category`	7	type 7	type 1		`text`
`act_char_1`	51	type 9	type 1		`text`
`act_char_2`	32	type 9	type 1		`text`
`act_char_3`	11	type 9	type 1		`text`
`act_char_4`	7	type 7	type 1		`text`
`act_char_5`	7	type 7	type 1		`text`
`act_char_6`	5	type 5	type 1		`text`
`act_char_7`	8	type 8	type 1		`text`
`act_char_8`	18	type 9	type 1		`text`
`act_char_9`	19	type 9	type 1		`text`
`act_char_10`	6969	type 999	type 1		`text`
`act_outcome`	2			0.44395439657287086690	`boolean`

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section: - *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?* - *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?* - *If a dataset is **not** present for this problem, has discussion been made about the input space or input data for your problem?* -

Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)

Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section: - *Have you visualized a relevant characteristic or feature about the dataset or input data?* - *Is the visualization thoroughly analyzed and discussed?* - *If a plot is provided, are the axes, title, and datum clearly defined?*

Algorithms and Techniques

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section: - *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?* - *Are the techniques to be used thoroughly discussed and justified?* - *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

Benchmark

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section: - *Has some result or value been provided that acts as a benchmark for measuring performance?* - *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section: - *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented? - Based on the* ***Data Exploration*** *section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected? - If no preprocessing is needed, has it been made clear why?*

Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section: - *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data? - Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution? - Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section: - *Has an initial solution been found and clearly reported? - Is the process of improvement clearly documented, such as what techniques were used? - Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section: - *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate? - Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data? - Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results? - Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section: - *Are the final results found stronger than the benchmark result reported earlier? - Have you thoroughly analyzed and discussed the final solution? - Is the final solution significant enough to have solved the problem?*

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section: - *Have you visualized a relevant or important quality about the problem, dataset, input data, or results? - Is the visualization thoroughly analyzed and discussed? - If a plot is provided, are the axes, title, and datum clearly defined?*

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section: - *Have you thoroughly summarized the entire process you used for this project? - Were there any interesting aspects of the project? - Were there any difficult aspects of the project? - Does the final* *model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section: - *Are there further improvements that could be made on the algorithms or techniques you used in this project?* - *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?* - *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting, ask yourself. . .

- Does the project report you've written follow a well-organized structure similar to that of the project template?
- Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your analysis, methods, and results?
- Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?
- Is the code that implements your solution easily readable and properly commented?
- Does the code execute without error and produce results similar to those reported?