KDD Cup 98 Challenge

Task

For a direct mailing campaign organised by a non-profit organisation (http://kdd.ics.uci.edu/databases/kddcup98/epsilon_mirror/cup98doc.txt), build statistical models that:

1. Identify the recipients that will engage with the campaign.
2. Maximise the campaign's revenue.

My solutions to these tasks are in scripts `donors.py` and `profits.py`, respectively. The technical report is at `report.html`. All the code and the report are available in a github repository (https://github.com/rebordao/kdd98cup).

Comments About The Project

My goal was to implement a system that would fullfil the objectives above and therefore drive up the organisation's revenue.

Often I write two types of reporting; one technical and the other more high-level that caters for clients, business analysts or project managers. In this report I describe the project in technical terms and thus its content, format and emphasis are directed for a suitable audience.

Also sorry for not having graphs to support this report but I didn't have enough time to do some nice plots and tables. If I had more time to work on this I would have done the following as well:

- Spend more time with the Feature Selection.
- Do cross-validation grid-search for Model Selection.
- Had used an IPython Notebook to illustrate the Exploratory Analysis; the Feature Selection; the Model Selection; and the results.
- Do a business action plan supported by the results of this analysis.

The file `donors.py` and `README.md` are good starting points to understand my solution and its steps.

Please focus in my solution to task 1 (`donors.py`) because in problem 2 I was into quick and dirty mode. However, in this report I indicate what can be improved.

Dataset

This dataset was used in the KDD Cup 98 Challenge (http://www.sigkdd.org/kdd-cup-1998-direct-marketing-profit-optimization). It was collected by a non-profit organisation that helps US Veterans. They raise money via direct mailing campaigns.

See the documentation and the data dictionary (https://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html) for more information.

The profits when targeting the entire testset are $10,560. The cost of sending each mail is $0.68.

Size

- 191779 records: 95412 training cases and 96367 test cases
- 481 attributes
- 236.2 MB: 117.2 MB training data and 119 MB test data

My Solutions

They are structured around the following steps:

1. Data Importation
2. Exploratory Analysis
3. Data Munging
4. Feature Selection
5. Model Selection
6. Training
7. Testing
8. Model Evaluation and Comparison

In my solution to task 1 I follow this procedure.

In my solution to task 2 first I predict who is a donor, and then - using just those samples - I train a classifier that predicts how much the person donated. Then I mail all the ones where the prediction is higher than $0.68.

I used only the training cases that were provided and made my training and test sets out of that file. Thus my train and test sets together have 95412 cases.

System Architecture

```
.
├── README.html
├── README.md
├── config.yml
├── data
│   ├── cup98LRN.csv
│   └── cup98lrn.zip
├── donors.py
├── lib
│   ├── __init__.py
│   ├── analyser.py
│   ├── importer.py
│   ├── preprocessor.py
│   ├── utils.py
├── profits.py
├── report.html
└── report.md
```

The main files are `donors.py` and `profits.py`. The project's configuration is at `config.yml` and all the auxiliary classes and their methods are in `lib`.

Strategy, Comments, Decisions and Results

My comments to the steps mentioned in section My Solution are:

Data Importation

What I Did

- I built a class for all importations in the project at `lib/importer.py`.
- I had some issues importing the raw data because lines ending in `^@` break the importation of the file. A workaround is to use `error_bad_lines = False` as an argument of `pandas.read_csv()`.

Comments

- All the hard-coded settings sit in `config.yml` and are read at this stage.
- Later I'll refactor `importer.get_raw_dat()` such that it imports the data directly from the compressed file instead from the `csv` file.

Exploratory Analysis

What I Did

- Checked the dimensionality of the raw data and how many missing values are per variable.
- Looked at the data, analysed its variables, type and meaning.
- Looked at the distribution of the target variables.
- Computed the Pearson's correlation between the targets and the predictors.
- Computed a set of descriptive statistics (mean, median, count, std, min, max, percentiles, etc).
- Checked the documentation to understand how the variables are categorised.
- If I had more time I would had checked how donations are distributed among age groups and per gender.

Comments

- The dataset contains only 5% of donors.
- The donations are usually smaller than $20.
- This data is quite noisy, high dimensional and with lots of missing values. Feature selection and preprocessing will be vital for good modelling.
- There are records with formatting errors.
- There is an inverse relationship between the probability to donate and the amount donated.

Data Munging

What I did

- Identified redundant variables based on:
    - low variance,
    - low sparsity,
    - linear dependency to other variables,
    - common sense.
- The previous step is made via a method at `Analyser.get_redundant_vars()`.
- Performed dimensionality reduction by dropping those vars/columns.
- Imputed the data by filling in the missing values with the mean if the variable is a numeric type or with the most common term if the variable is an object type. Made a method for this at `Preprocessor.fill_nans()`.
- Shuffled the data.
- Changed categorical variables into a numerical representation.

Comments

- Found several redundant variables but dropping them wasn't enough for significant dimensionality reduction.
- With all the missing values this step is quite sensitive. I hope the data imputation doesn't add too much noise to the variables.
- Could had tried to apply Principal Component Analysis for dimensionality reduction but I didn't have enough time for that.

Feature Selection

What I did

- Got important variables by doing Feature Selection. Made a method for this at `Analyser().get_important_vars()`.
- Tried the following methods for Feature Selection:
  - Correlation-based Feature Selection;
  - Variance-based Feature Selection;
  - Univariate Feature Selection;
  - Tree-based Feature Selection.
- Dropped the non-important features from the data.
- Changed categorical variables into a numerical form.
- Made train and test sets, in its full form and in a balanced version.

Comments

- Even if I tried many methods for Feature Selection I could not spot an optimal set of variables, and this is vital for good performance.
- If I had more time I would had fiddled more with Feature Selection.
- The balanced set is useful because some methods converge faster and better if the training data is balanced.
- For this dataset this seems to be the most important part of the statistical modelling.

Model Selection

What I did

- I tried manually several combinations of parameters of the training methods and watched its impact. Then I chose the ones that seemed performing decently.

Comments

- An obvious choice for this is to do cross-validation grid search to find optimal parameters, but I haven't had time to do it.

Training

What I did

Task 1

- Deployed 4 methods:
  - Method 1 | Decision Tree Model.
  - Method 2 | Random Forest Model (also used Extremely Randomized Trees).
  - Method 3 | Logistic Regression Model.
  - Method 4 | Ensemble Model of the previous 3 methods.

Task 2

- First I predict who is a donor, and then - using just those samples - I train a classifier that predicts how much the person donated. Then I mail all the ones where the prediction is higher than $0.68.
- For predicting the donors I used Logistic Regression and for predicting the donations I used Linear Regression.

Comments

Task 1

- Would be nice to try as well a Naive Bayes Model and a Neural Network Model.
- Choose Method 3 as a baseline and
- tune in the parameters of the training methods of methods 1 and 2.

- Cherry pick the best 3 and build an optimal ensemble method.

Task 2

- Should had used in both classifiers cross validation.
- Need to reevaluate my implementation and to make the training faster.

Testing, Model Evaluation and Results

Task 1

- For each method computed the confusion matrix, accuracy, recall, precision and F1. Made a method for this at `Performance.get_perf()` .

```
              Model Name Performance Metric      Value
           Ensemble Model             recall  56.449376
           Ensemble Model                 F1  12.371761
           Ensemble Model          precision   6.947171
           Ensemble Model           accuracy  59.601303
     Decision Trees Model             recall  49.861304
     Decision Trees Model                 F1   9.951557
     Decision Trees Model          precision   5.527368
     Decision Trees Model           accuracy  54.412641
      Random Forest Model             recall  56.726768
      Random Forest Model                 F1  11.901644
      Random Forest Model          precision   6.648244
      Random Forest Model           accuracy  57.572785
Logistic Regression Model             recall  57.628294
Logistic Regression Model                 F1  13.178971
Logistic Regression Model          precision   7.440236
Logistic Regression Model           accuracy  61.640332
```

- All the methods don’t perform significantly well.
- The best one is Logistic Regression.
- It would had been nice to display for each method its lift, AUC and ROC curves.
- With better Feature and Model Selection the results can be improved.

Task 2

- Uses Pearson’s correlation to evaluate tje performance of my solution.
- The correlation varies between 0.4 and 0.73 because I am evaluating it in a reduced set and didn’t compute a statistical mean of it.
- I think the results can by improved by reevaluating my implementation and the feature selection.

References

The article Learning and Making Decisions When Costs and Probabilities are Both Unknown (http://cseweb.ucsd.edu/~elkan/kddbianca.pdf) (2001) by B. Zadrozny and C. Elkan is an interesting reading that provides good insight into task 2, i.e. maximising the profit of the campaign. However I didn’t have time to replicate their approach.

Author

Antonio Rebordao (https://www.linkedin.com/in/rebordao) 2015