ID5059 - Individual assignment

Chrissy Fell

ID5059 - Knowledge Discovery & Data Mining

Coursework Assignment 1 - Individual

Deadline: Friday 23rd February 2024 (week 6), 9pm

Credit: 50% of coursework mark, 20% of overall module mark

Learning Objectives

On successful completion of this assignment you should be able to:

- investigate the properties of a real-world dataset
- prepare a real world dataset for analysis
- construct several simple machine learning models aimed at predicting a categorical attribute from other attributes
- perform a simple evaluation of the results of those models
- report your solution

Requirements

It is not necessary to obtain the best possible performance by searching far and wide for the most state-of-the-art algorithm. You should instead use a reasonable model and performance measure similar, if not identical, to those discussed in our lectures and readings. Which model you use, and how you evaluate its performance, is up to you. **If your model performs poorly by your selected metric, do not worry. Your goal is to find a sensible approach and to produce clear, concise, understandable code and text documenting your effort.** Do not attempt to code everything from scratch. You are expected to use packages discussed in the lectures and readings, or similar. However, you should understand, and be capable of explaining, the packages you use.

Data

You will use data derived from the Flight status prediction dataset on Kaggle, posted by Rob Mulla. **Do not download the data to a lab machine or School server**. This data has been extracted from the Marketing Carrier On-Time Performance database of the TranStats data library. More information on the attributes can be found on both Kaggle and the TransStats website.

The full dataset contains data for 5 years (2018-2022) each of which has at least 3 million rows, and the file size is 8GB. There isn't space on School systems for everyone to have their own separate copy. Instead, the dataset and various subsets are provided at the following path accessible on the Linux Lab PCs and CS teaching server:

- The directory 4_huge has been split 80/20 into training (6GB) and testing (1.4GB) files.
- The directory 3_large contains a subset of the same data as the original file approximately 1GB in total. It is also split 80/20 into training and testing files.
- The directory 2_medium contains subsets of the large files (100MB).
- The directory 1_small has one file containing a subset of the medium data for all years (10MB).

All of these can be accessed via the file system from a School lab machine or server. There is a starter notebook (HTML version), showing how to do this in Python, that you can adapt if you wish (in which case you will need to download a copy of the notebook before you can change it). Similarly a starter Rmarkdown is available for R.

Instructions

The task is to predict if, before a flight takes off, it will suffer some type of disruption. By disruption we mean cancellation (the attribute *Cancelled*), diversion (the attribute *Diverted*) or a delay of more than 15 minutes (the attribute *ArrDel15*). In the data available on the CS teaching server these have been combined into the attribute *Disruption*.

This data covers the period of the COVID-19 pandemic in 2020 and 2021, during which there was severe disruption to air travel. You should visualise the impact of COVID-19 during the data exploration and if needed include suitable attributes in your model.

The data is recorded after the plane has landed. Since the task is to predict if a future flight will be disrupted you should only use attributes that are available before the plane departs. You *MUST NOT* use the following columns from the Kaggle dataset in your model (all these columns have already been removed from the huge and large datasets available on the school server, the Cancelled, Diverted, ArrDel15 columns remain in the small and medium datasets):

DepTime, DepDelayMinutes, DepDelay, ArrTime, ArrDelayMinutes, AirTime, ActualElapsedTime, DepDel15, DepartureDelayGroups, TaxiOut, WheelsOff, WheelsOn, TaxiIn, ArrDelay, ArrivalDelayGroups, DivAirportLandings, Cancelled, Diverted, ArrDel15

You will need to:

- Split into training and test sets (for small or medium files; this is already done for you for the large and huge files, to avoid you having to make separate copies).
- Considering the descriptions, data types, missing data and number of unique values of the attributes choose a small number of attributes (no more than 10) that could plausibly predict disruption.
- Explore and visualise your chosen attributes, you may use feature engineering to create up to 3 new attributes.
- Perform data cleaning and prepare the data for modelling.
- Select and train no more than 3 models and fine tune the most promising model.
- Evaluate performance on the test data using an appropriate measure.
- Present your findings in a one page summary report aimed at readers who are not machine learning experts.

It's strongly recommended that you start with one of the smaller data files, and gradually scale up to the larger ones once you have established your process. If you find that you can't feasibly process the full dataset this won't be a problem, as long as you clearly explain what you have done. Be aware, though, that the smaller files are extracted from the larger ones at random, they may not be representative of the larger data.

You can use either R or Python. If you have a strong reason for wanting to use something else, talk to the lecturers. Unless you have obtained explicit permission in advance, other languages will not be accepted.

Key Points

- Your solution should follow a logical process, for example the machine learning project structure from the lectures and course text.
- We are not looking for the best performing models: we are looking to see if you can build a sensible model and a sensible evaluation of its performance.
- We are also seeing if you can clearly document your effort, interpreting the output you create, explaining and justifying the choices you make in the process.
- Your solution shouldn't need to include more than a couple of hundred lines of code, though you won't be penalised for more.
- If you are struggling to make something work with the volume of data present, you can use one of the smaller subsets. But explain what you have done, and why it is sensible.
- Presentation counts for both code and report.
- A concise notebook (Jupyter, Rmarkdown or Quarto) complete with markdown annotations or some equivalent will earn more marks than an enormous raw text file full of opaque and poorly commented code.
- The CS student handbook contains writing guidance and coursework report writing tips.
- The summary report should present your solution as described in the lectures and course text. For example, explaining how your model meets the task (look at the big picture), the performance of the model on data unseen during training, the reasons for your major decisions, the details of your final model and any insights gained either from the issues encountered or into which factors are most important.
- Two models using the same algorithm can give very different results depending on the data (observations and features) used for training and hyperparameters selected. When describing a final machine learning model you should aim to give enough details so that it could be reproduced.

- If you do not understand something or have questions, you are encouraged to discuss it with your peers (say, via the Teams channel) or the module staff. However, the deliverables that you submit must, of course, comply with the policy on Good Academic Practice.

Submission

A single zip file containing the following must be submitted via MMS by the deadline. Submissions in any other format will be rejected. **You are reminded of the importance of re-downloading and checking immediately after submission.**

1. The code of your solution, preferably in a notebook (e.g. Jupyter (.ipynb), Rmarkdown(.rmd) or Quarto(.qmd)) with markdown annotations, or something similar built to be read with a web browser or PDF reader.
2. A clear and concise summary report (**one page maximum**) in a PDF file.

Assessment Criteria

This assignment will be marked on the standard 20-point scale using the following mark descriptors:

- **0** Nothing submitted.
- **1-3** Little evidence of any significant attempt to complete the work.
- **4-6** No substantial relevant material submitted, or no evidence of significant progress on any of the basic ML process elements.
- **7-10** A reasonable attempt at most of the basic ML process elements including summary report, perhaps major flaws in the process or limited explanation and analysis justifying choices.
- **11-13** A competent attempt at most of the basic ML process elements and summary report, poor presentation of code or summary report, explanation of choices and results unclear or shows confusion, uses limited amount of data.
- **14-16** A clearly presented and explained notebook and summary report, demonstrating significant attempts at all of the basic ML process elements, and success in most elements with adequate explanation of choices and results.
- **17-18** An excellent and clearly presented notebook and summary report, demonstrating successful implementation of all the basic ML process elements with well explained decisions and analysis of results.
- **19-20** An exceptionally clearly presented notebook and summary report, demonstrating extensive implementation of all the basic ML process elements, very well explained decisions and analysis of results. Assignment conveys significant insight into the task and important features in success of the model.

Lateness

The Computer Science standard penalty for late submission applies (Scheme A: 1 mark per 24 hour period, or part thereof).

Good Academic Practice

The University policy on Good Academic Practice applies. This is an individual assignment. Any aspects of the submission that are not entirely your own work must be explicitly acknowledged.