Individual Project

This short project tests your ability to construct and evaluate machine learning models. You are asked to produce a simple machine learning model and a measure of its performance. 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.

Deadline

Deliverables must be submitted by Friday 6th of March (in week 6). Starting on the 7th of March, marks M as a function of time t decay exponentially, $M(t) \propto \exp(-t/\tau)$, with $\tau = 4$ days.

Data

You will use the UK government's land registry data. The task is to predict how much a property will sell for. You can obtain the data at:

https://data.gov.uk/dataset/4c9b7641-cf73-4fd9-869a-4bfeed6d440e/hm-land-registry-price-paid-data

The site has many files: please follow the "Show more" link under "data links", and then download the "2015 FULL Price Paid Data-Single file 1995-2015" (CSV) file, which contains all registered purchases since 1995 as comma separated values. The data file is about 4 GB in size. A description of the data's columns can be found at:

https://www.gov.uk/guidance/about-the-price-paid-data#explanations-of-column-headers-in-the-ppd

Instructions

Each row in the file contains a property that was purchased, the price that was paid, and features of the property and purchase. Your model should **predict the price that was paid** from

at most three simple features: the **lease duration**, the **property type**, and **whether or not the property is in London**. In the data:

- Rows are separated by newlines, while the columns are separated by commas.
- Column 2 contains the **price that was paid**.
- Column 5 contains the **property type** (meanings of the codes can be found in the link above).
- Column 7 contains the **lease duration**.
- Column 12 contains the town or city in which the property was located. **You can judge a property to be in London if this field contains the word "London"**.
- Ignore the remaining columns.

Testing and Training

Any purchases prior to 2015 are to be used as training data, while those made in 2015 are to be used as test data. You can either do this split dynamically, or split the one large file into two files as a pre-processing step.

Key points

- I am not looking for a model that performs well: I am looking to see if you can build a sensible model and a sensible evaluation of its performance, and also if you can clearly document your effort.
- You should submit a solution with no more than a couple hundred lines of code.
- If you are struggling to make something work with the volume of data present, you can subsample (for instance, look at a month or a year's worth of data). But explain what you have done, and why it is sensible.
- If you are having trouble extracting features, can you submit an evaluation of a sensible baseline on the test data?
- You can use any programming language you like to solve the problem: pick a language suited to the task, and one you are comfortable with, but your code must be presentable and understandable. **Presentation counts.** A concise jupyter notebook complete with markdown annotations or some equivalent will earn more marks than an enormous raw text file full of opaque and poorly commented code.
- If you do not understand something or have questions, you are encouraged to discuss it with your peers (say, via piazza) or myself. However the deliverables that you submit must be your own original work.

Deliverables

Please upload two things via moodle:

1. The code of your solution, preferably in a jupyter notebook with markdown annotations, or something similar built to be read with a web browser or PDF reader.
2. A brief clear and concise single-paragraph summary describing your model, your measure of its performance, and your result, in a single-page PDF file.

Marks will be deducted for code or summaries that lack sensibility, clarity, or brevity. Again: your goal is a sensible model, a sensible measure of its performance, and a clear and concise summary of your effort.

Do not upload any of the data, or your model's predictions.