Imperial College London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Analysing property sales data using Data Science

Author:
Wenxiang Luo

Supervisor:
Chiraag Lala

Submitted in partial fulfillment of the requirements for the MSc degree in MSc Computing of Imperial College London

June 2022

Contents

Chapter 1

Introduction

Nowadays, there are a significant amount of data generated every second. The daily lives of humans are producing it, and some other fields are generating numerous data, such as research, health care, commercial activities, and environmental information from all kinds of sensors. With these considerable amounts of data, acquiring the relationship between some aspects of records or obtaining the patterns behind these data would benefit the whole world. For instance, new causes of diseases might be discovered, and technology development could be accelerated.

However, this extensive data can be one of the main obstacles for analysis as it is approximately impossible for humans to obtain insights into the data manually. Under this circumstance, artificial intelligence (AI), a technique that empowers the computer to imitate human intelligence and manner, could be one of the methods to mitigate this problem. It can extract patterns from large datasets and use them to make predictions based on future data and even distinguish which parts of data cause the results.

In this project, some AI techniques will be used on property and demographic data to gain insights and understand the factors that impact a homeowner's propensity to sell. The factors could be the distance to schools, hospitals, or supermarkets, the accessibility to public transport, and the types of the properties (flats or houses). It could be highly advantageous to estate agents who would discover homeowners with more potential to become clients and provide them with business.

Chapter 2

Literal Review

Python is one of the most popular programming languages in the world as it is easy to develop, and there are extensive packages for various functionalities. In this project, Python and its several packages would be used for loading data, preprocessing data, and building and evaluating machine learning models.

2.1 Machine Learning

Machine Learning (ML), a subset of AI, is a technique that the computer can learn and improve from data without explicit programming. The reason for using ML is that its performance is sometimes better than the traditional approach. For example, ML techniques would simplify the solution to a problem that comprises a long list of rules (spam mail detection).

ML can be divided into three categories, and supervised learning is one of them. In supervised learning, the dataset contains features (input to the model) and targets (ground truth of the output), and the model's parameters are randomly initialized. Then the features are passed to the model, and the differences between the current output and the ground truth are used to update the parameters until the differences are acceptable.

In this project, a supervised learning model will be implemented for data analysis, and the steps are listed below.

1. Data Preprocessing: Some data from the dataset might be missing, and these values should be treated appropriately before passing to the model.

2. Standardization: In real lif, different features usually have different ranges, and this will cause a problem in ML, which is that high magnitude features would have more weight than low magnitude features (Fandango, 2017). Standardization, which could scale all the features to the same magnitude, is one of the solutions.

3. Feature encoding: ML models require numerical values, whereas the categorical features in the dataset do not meet the requirement. Therefore, these features should be converted into numerical values.

4. Training & Testing: The parameters of the model are updated, and the loss is expected to converge during training. The performance of the model is validated when testing.

2.2 NumPy

Numerical Python (***NumPy***) is a scientific computing package utilizing an optimized C/C++ API to use less computation time than pure Python calculations (McKinney, 2012). It provides some data structures, one of the most important structures is ***ndarray***. Unlike Python lists, NumPy arrays are homogenous, meaning all the elements in them are of the same type, and their sizes are fixed (Fandango, 2017). Moreover, NumPy provides various built-in functions for different purposes, such as statistics, linear algebra, transforms, and element-wise operation.

In this project, ***Numpy*** will be used as part of data preprocessing. For example, converting categorical data into integers.

2.3 Pandas

The ***pandas*** is a fast and compelling package capable of handling data of various types (numerical values, strings, and time) and from multiple sources (CSV, Excel, and MqSQL database). One of the ***pandas*** data structures is ***DataFrame*** which is appropriate to handle tabular data whose columns are of different types. Also, it could manage various operations, such as manipulating missing values, creating pivot tables (summary of the table), and grouping data from different columns (Fandango, 2017).

In this project, ***pandas*** would be used for data loading and preprocessing. First, data is loaded, and it is represented as ***DataFrame***. Then the attributes of the data, such as distribution, should be inspected. Finally, the data is preprocessed.

2.4 Scikit-learn

Scikit-learn is a Python ML package which provides various techniques for data mininig, modeling and analysising. It could be commonly used in multiple ML tasks, such as classification, regression and clustering. In this project, the package will be used for analysising the performance of the models.

2.5 PyTorch

PyTorch was developed by Facebook, and it is one of the popular libraries for building ML models. It is also the basis of numerous packages that the developers could take advantage of (Godoy, 2021). Moreover, this library could perform auto differentiation by using graphic processing unit (GPU) acceleration, which results in less training and evaluation time.

In this project, some of the components provided by ***PyTorch*** will be utilized.

- ***nn.Sequential*** is the container that contains all the layers of the ML model
- An appropriate loss function would be selected to compute the loss and perform backpropagation.
- One of the optimizers would be used to update the model's parameters.

2.6 Matplotlib

Matplotlib is one of the Python plotting packages that can plot different types of figures, such as histograms, pie plots, and line plots. In this project, this library will be primarily used for data visualization. For instance, the distribution of the raw data should be visualized to determine the procedures of preprocessing and plotting the training loss and validation loss during model evaluation.

Chapter 3

Project Plan

3.1 Data Preprocessing

3.1.1 Handling Missing Values

In this project, two methods will be attempted to preprocess the missing values.

1. Drop missing values: This method deletes the messing values for data analysis.

2. Fill the missing values: In this approach, the missing values are filled with mean, median, or default values, for example, zero or some constants.

3.1.2 Standardization

In this project, two different approaches will be attempted.

1. Standard scaling: By applying this method, the mean of the feature is removed and then divided by the standard deviation.

2. Min-max scaling: For this approach, the minimum and maximum of the raw data are used to transforme it into a specific range.

3.1.3 Feature Encoding

In this project, label encoding will be applied for this goal. It would convert the categorical values into a sequence of integer values.

3.1.4 Separating Dataset

The origional dataset should be shuffled and divided into three parts which will be used for training, validation and testing.

3.2 Build Model with PyTorch

3.2.1 Model architecture

- **Basic Model**: The number of the input neural of the model should be the same as the number of features, then there are hidden layers. Finally the output layer is a sigmoid function as the this model should outputs the possibility of the selling.
- **Advanced Model**: The advanced model contains more conponents, such as dropout or batch normalization, these additional layers might reduce overfitting and enhance accuracy.

3.2.2 Training

1. Use ***DataLoader*** to feath baches of data for training, and the batch size could be set to 128 if there are sufficient memory.
2. The loss finction used for model training is ***MSELoss*** as the output is a continuous value.
3. The optimizer for updating parameters could be ***SGD*** or ***Adam***.
4. Train the model iteratively and the number of iteration (epochs) will be set to 500 and will be updated if necessary.

3.2.3 Hyperparameters Tuning

- Learning rate: The inintial learning rate will be set to 0.001 and it will be increased/decreased depending on the performance. Moreover, learning rate schedulers, for example, ***LambdaLR*** and ***StepLR*** could be applied during training.
- Number of epochs: If the initial value cause overfitting or underfitting then the epochs should be decreasd or increased.
- Activation functions: The inintial activation function will be ReLU, but other activation functions, such as TanH and Parametric ReLU will also be tested to enhance the performance.

3.3 Evaluation

After constructing and training models, the next step is testing the performance of these models. The mean squared error (MSE) could be used to test the performance numerically, the less the MSE, the better the performance. Moreover, a line plot contains the training loss and validation loss against epochs should be generated, this plot could be used to determing which the model is overfitting/underfitting.

Finally, an optimal best model is selected and it is tested using the test dataset to produce the final result.

Bibliography

Fandango, A. (2017). *Python Data Analysis*. Packt Publishing Ltd. pages 2, 3

Godoy, D. V. (2021). *Deep Learning with PyTorch Step-by-Step: A Beginner's Guide*. pages 4

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.". pages

McKinney, W. (2012). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.". pages 3