Group Proposal

In this project, we will evaluate the performance and predictive power of models that has been trained and tested on data collected from houses in suburbs of Boston, Massachusetts. Models trained on this data that is seen as a good fit could then be used to make certain predictions about a house– in particular, its monetary value. The models would be proved to be invaluable for someone like a real estate agent who could make use of such information on a daily basis.

If you're studying data science, you probably heard of the Boston housing dataset, which is a very ubiquitous dataset. This dataset contains information about 506 census tracts of Boston from the 1970 census. And each of the 506 entries represent aggregated data about 13 features and 'MEDV' as response, including average number of rooms per dwelling, pupil-teacher ratio, per capita crime rate and so on, for homes from various suburbs in Boston, Massachusetts.

Like 'hello world', the Boston Housing Dataset has become part of a common vocabulary. And it will remain so, not only because thoroughly labeled datasets for machine learning are still not that easy to find, but because using the same dataset for decades to test different algorithms has allowed scientists to control for that variable and highlight the differences in algorithm performance. However, the Boston housing dataset is small, especially in today's age of big data. If the amount of data is too small, the model is likely to appear overfitting, resulting in poor model performance. In order to reduce the impact of this problem on the training model, we intend to optimize the data preprocessing and model selection by, for example, reducing the number of features, cross-validation, and reducing the complexity of the model.

Since this problem is a standard regression problem, we will use sklearn.neural_network.MLPRegressor to build NN model of it combining what we learned in class. In our initial plan, our network will not be too complex. Just one hidden layer is enough and we will use logistic function as our transfer function. In the meantime, we will try to use different transfer function to evaluate the performance of our model. MSE is calculated as our loss. And it's one of main part to evaluate our model. What's more, considering the small amount of data we have, we will use some ensemble learning methods to strength our model. Like random forest and boosting methods.

Considering the amount of data and our model part are just traditional machine learning model and shallow neural network. We can just use python with cpu to achieve this. By the way, we use Google Colaboratory, which is faster and have free Gpu.

Our main reference books are Python Machine Learning and Neuron Network Design. In addition to that, We looked through a lot of API reference of sklearn. According to Python Machine Learning, we determine the models that can be used for regression predict problems. For the neuron network model, we mainly refer to Neuron Network Design. For coding problems, our main reference website is scikit-learning, whose URL is https://scikit-learn.org/.

As for metrics used in judging the performance of the model, we mainly mean square error (MSE), R square (R^2) and loss function. Mean square error is the average of errors squared. Thus, higher errors will result in a higher MSE as squaring the error doubles the error factor. The closer the value of MSE is to 0, the better the model fitting. R square can explain the variability of the dependent variables. The value range of R^2 is 0 to 1. The closer the value of R^2 is to 1, the better the model fitting. The loss function is used to measure the degree of inconsistency between the model's predict value and the true value. It is a non-negative real-valued function. The smaller the value of the function, the better the robustness of the model. In our neuron network, we will use the square loss as the metrics.

Finally, a rough schedule for completing the project is listed as following:

6.5	Group meeting – Discuss the whole schedule
6.6-6.9	Find datasets
6.10	Group meeting – Decide dataset
6.11-17	Data preprocessing and train model (Use as many methods as possible)
6.18	Group meeting – See results and decide which methods to use
6.19-6.23	Improve model based on discussion
6.24	Group meeting – Summarize results
6.25-6.29	Write final project and prepare for presentation