Machine Learning Career Track Capstone Proposal:

Clustering Successful Suburbs and Predicting Location in Melbourne Housing

Drew Lehe

Introduction:

Real Estate is a great topic for data science because

1) It's the largest industry in USA, the world's largest economy, and many other economies
2) There's a plethora of rich data on housing available
3) Built environments have a large effect on people's lifestyle, and, because housing is such a large sector of the economy, it's very influential on economies

Melbourne, Australia experienced a steady housing bubble from 2005 to 2019, fueled by a surge of foreign investment, domestic speculators, and local restrictions on land use that can cause shortages. According to the International Monetary Fund, Australia, at one point, had the 3rd-highest housing price-to-income ratio in the world, with price levels well above its historical average.

In their paper "Cluster Analysis for Neighborhood Change," Reibel and Regelson cluster neighborhoods of Los Angeles to track demographic shifts in neighborhoods. They use KMeans, but I'll introduce a slightly more advanced model, the Gaussian Mixture Model (for clustering). Then,

Problem Statement:

Reports are now that the bubble has burst and Australian housing prices are falling. Ideally, Australians could avoid both expensive rents and lost investment by averting a bubble altogether. Identifying what caused rising property prices will help in this endeavor. Because of the intricate and unique nature of cities, it's essential we build a flexible model that can transfer to multiple cities and their datasets. With my skill in data mining, I hope to examine residential real estate data and find out:

-What creates a high-value suburb? Is it close to the city center and densely-populated, or is it remote and sparse? What kind of housing is built there?

I'd like to use Gaussian Mixture Modeling to create several clusters of Melbourne suburbs to find out what is driving this trend. Through analyzing the different clusters, we can find out which attributes the priciest suburbs had.

One beneficial use is that, with good clusters, we can track the changes of these neighborhoods through time and observe the effects. This is very useful for demographers, sociologists, marketers and real estate developers.

Next, I'll take a list of features and predict which city a house is in. This is important for the same groups of researchers because public data on cities is often messy or incomplete, and wielding it can be a prohibitive engineering task. Hopefully, I can prove that decision trees are a better method than more purely quantitative algorithms such as linear classifiers or K-Nearest Neighbors.

Dataset and Inputs:

The data is the Melbourne Housing Market dataset from Kaggle. Features I'll examine are distance from the city center, year the house was built, price, number of rooms, number of bathrooms, number of car spaces, type of housing, size of the house and size of the building. I'll also investigate the *types* of housing there to see if areas with more multifamily have lower prices. There was not any extreme class imbalance in housing variety, though the slight majority of housing units were 'h' (single-family detached). Some suburbs had only one or a few observations, but I didn't eliminate them from the data because that would make the prediction/clustering problem too easy to solve.

However, I'll eliminate a few of them for my machine learning models. My Gaussian Mixture Model includes square meters, year, and price. My decision tree includes square meters, year the house was built, distance from the city center, and size of the lot.

Notably, some columns are not explained on the site, so I chose not to include them in my model. However, I *did* include them in my PCA to examine their significance and avoid overlooking something that could be consequential.

Evaluation Metrics:

I'll evaluate the quality of each cluster with the silhouette score and create separate graphs of silhouette scores for each cluster, for several amounts of clusters. For the boosted tree model of location and price prediction, I'll use the F1-score for the classification component, and silhouette score for the clustering component.

However, the important data story here is: what *kind* of housing was in each suburb? Did the high-priced suburbs have denser housing? Were there taller buildings? Or did the suburbs with pricey housing have restrictive land-use policies that only allowed low-density housing?

Project Design:

I want to create an efficient system of finding patterns in city data.

Cleaning: First, I'll load the data and clean it; outliers and unusual observations are easiest to clean with Excel, but easier to find through plotting in Python.

EDA: Using Python, I'll construct several correlograms to find any interesting patterns in the data and make note of them. Using pandas' built-in methods like .head() or .describe() helps get a preliminary feel for the data.

PCA: Next, use scikit-learn's PCA command to examine all numerical features, and write functions that graph feature weights and biplot all features in the 1st two principal component dimensions. Interpreting the graphs will help find relationships among my variables.

Clustering: Now that I have clean data I understand, I'll create a *separate* DataFrame of the suburbs, using the features I determined were important. This is where it gets interesting. First I'll cluster using KMeans (as a benchmark), graph the results, then use Gaussian Mixture Modeling to create more

detailed clusters. To determine the number of clusters, I'll plot an elbow chart and then a silhouette plot.

K-Nearest Neighbors Benchmark: I'd like to use this algorithm to make an important point in data science modeling: if you're classifying an arbitrary metric like city borders, something purely "numeric" like KNN will be a poor predictor. I run KNN to predict which city a house is in, and it returns 20% accuracy (F1-score).

Machine Learning with Boosted Decision Trees: To prove my point, I'll compare it to scikit-learn's untuned DecisionTreeClassifier, which achieves an F1 of 72.5%. Then I use XGBoost and GridSearchCV to find the best parameters and work up to 76.5% accuracy.

Solution Statement:

By understanding what causes inflated prices (before bursting), we can save investors and homeowners billions in losses. Experiments like this can guide municipal governments in creating policy: perhaps if denser construction ("upzoning" in planning jargon) were allowed, demand could be eased and prices would not inflate.

Marketers and other GIS mappers who rely on housing data can use this tool to impute missing data for customers and determine where to send ads. Sociologists can track changes in similar neighborhoods over time.