ISYE 6740 – Spring 2021
Final Project Report

Team Member Name
Inah Canlapan

Project Title
Property Style Reclassification

Problem Statement
At RPS Real Property Solutions, we have a property database that houses almost 6 million of Canada's 15 million residential addresses. The majority of this data is scraped from appraisal forms and includes geographical features like city, province, postal code, latitude and longitude and property features like property value, number of bedrooms, number of bathrooms etc. From the property features and other information from the appraisal form, business logic is applied and a property style is assigned.

Table 1: Distribution of Records in Property Database by Property Style

Property Type	% of Properties
1. Single Family Detached	67%
2. Semi-Detached	4%
3. Condo	11%
4. Row	7%
5. Plex	1%
6. Other	10%

There is a subset of these properties that come from a secondary source that only contains geographical features and the property value for the address. If an address from this source matches one already in the database, then that property type is assigned. Otherwise, the address is given a property type of "6. Other".

Table 2: Distribution of Records in Property Database from Secondary Source by Property Style

Property Type	% of Properties
1. Single Family Detached	40%
2. Semi-Detached	2%
3. Condo	8%
4. Row	5%
5. Plex	0%
6. Other	45%

Addresses with a property type of "6. Other" make up 45% of records from the secondary data source and 10% of the entire property database. Business logic cannot be used to classify this data since we don't have any property attributes. My goal for this project is to create a model using geographic features to predict property type so that RPS can use these records in market analysis and in data solutions.

Data Source

Property data is provided by my employer, RPS Real Property Solutions. It includes properties that were appraised from January 2005 to March 2022. The addresses in the data have been geocoded and standardized. Due to the nature of this data, I cannot share the actual datasets but have enclosed a copy of my code for reference.

The property data is separated into the following groups:

1. Training Data – 1.3M addresses
    - 50% of records from the primary source of data, already classified with a property style
    - Data was used to train the models described below in the Methodology section
    - Sample was stratified by Province to ensure that the models are able to predict properties across Canada
2. Test Data – 320K addresses
    - 50% of records from the primary source of data, already classified with a property style
    - Data was used to generate an accuracy score for each of the models, allowing me to choose the best ones
3. Validation Data – 534K addresses
    - From the secondary source of data, records with property style other than "6. Other"
    - Accuracy score calculated from this data was used as a comparison between the final two models
4. Target Data – 696K addresses
    - From the secondary source of data, records with "6. Other" as property style
    - Data was the true target for prediction as these addresses don't have an associated property style
    - A small subset of records were selected and assigned a true property style manually by using online resources in order to calculate an approximate accuracy score for the target data

All datasets were refactored to encode categorical or string fields into integer values.

I had originally chosen 80% of records from the primary source of data to be in the training set but found that my test accuracy was higher than the training accuracy. I adjusted the training size to 50% to correct this.

Table 3: Data Dictionary for Training, Test, Validation and Target Sets

Column Name	Description	Data Type
RecordID	Unique identifier for each record	int
Full_Address	Full address of the property	varchar
Street_Address	Street address of property	varchar
Suite	Suite/Unit number	varchar
Street_Number	Street number	varchar
Street_Name	Street name	varchar
Street_Type	Street type	varchar
Street_Direction	Street direction	varchar
City	City	varchar
Province	Province (or State)	char
Postal_Code	Postal Code (or ZIP Code)	char
Latitude	Latitude	float
Longitude	Longitude	float
Dissemination_Area	Small geographic groupings with an average population of 400 to 700 people	int
Property_Style	Property Style	varchar

Methodology

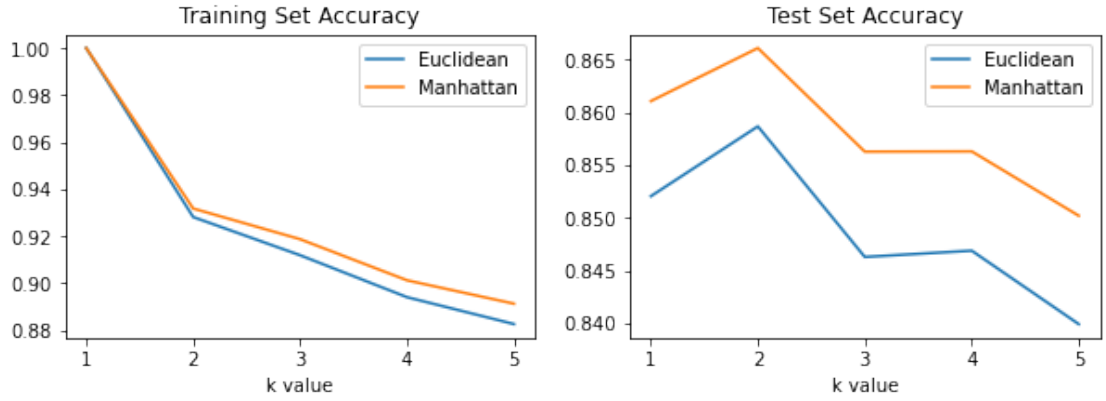
In my original proposal, I had only intended to create two models to compare results and choose the best one:

1. K Nearest Neighbours
    - My hypothesis is that KNN will work well for this data because if an unknown address is geographically close to other addresses that have the same property style, there's a good chance that the unknown address is also of the same property style.
2. Multinomial Logistic Regression
    - I am less confident that this type of model will work well for my data but I would still like to try it so that I can compare the results to KNN.

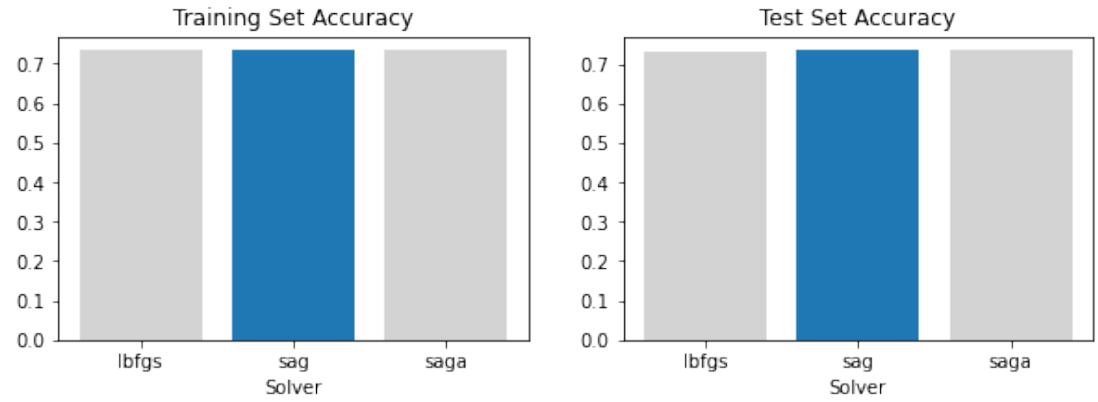
After completing the course material and learning about other models, I created three additional models to compare to KNN and Multinomial Logistic Regression:

3. Random Forest
4. Neural Network
5. Naïve Bayes

To tune the KNN model, I explored small values of k and tested different distance metrics. I chose the model with the highest test accuracy as the final KNN model (k = 2, distance metric = Manhattan).



For the Logistic Regression model, I wanted to investigate the impact of using different solvers. There was very little difference between the accuracy of each solver so I chose the one with the highest test accuracy (solver = sag).

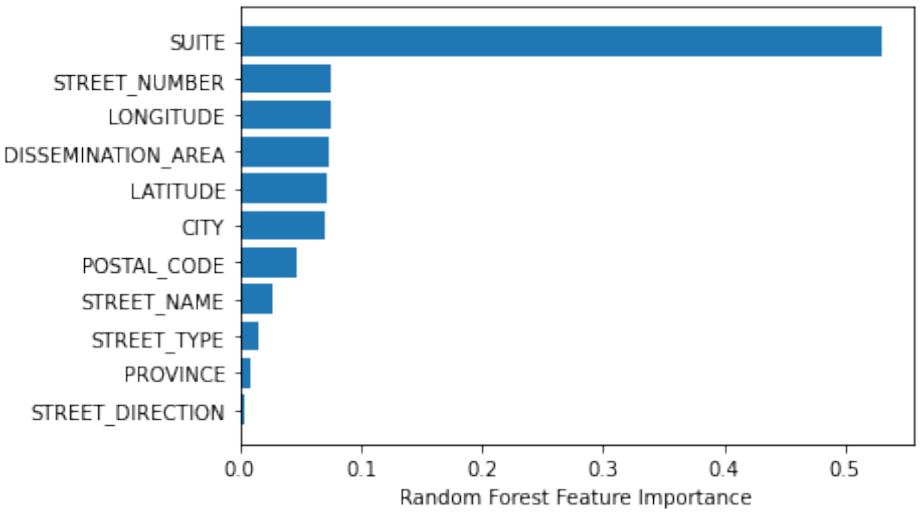


For the Neural Network model, I used GridSearchCV to tune the model parameters:

```
param_grid = {   'hidden_layer_sizes': [(150,100,50), (100,50,30)],
                 'max_iter': [200],
                 'activation': ['logistic', 'relu'],
                 'alpha': [0.0001, 0.05]  }
```

This code took 4 hours to run on my machine and resulted in these ideal parameters:
`{'activation': 'logistic', 'alpha': 0.0001, 'hidden_layer_sizes': (150, 100, 50), 'max_iter': 200}`

For the Random Forest model, I didn't use GridSearchCV to tune the parameters as it is computationally expensive and would have taken a long time. I chose to only set `min_samples_leaf = 100`.



Looking at the feature importance, I was surprised to see that Suite carried so much importance; I expected to see Street Number and Street Name or Latitude and Longitude at the top of the list.

For the Naïve Bayes model, I opted to use GridSearchCV to tune the var_smoothing parameter since there is only one parameter to tune. It was quick to run (around 8 minutes) and resulted in `var_smoothing = 1e-09`.

Evaluation and Final Results

Below is a summary of the training and test accuracy of all the models.

Table 4: Comparison of Model Training and Test Accuracy

Model	Training Accuracy	Test Accuracy	Difference
KNN	92.3%	85.0%	-7.2%
Logistic Regression	73.3%	73.3%	0.0%
Random Forest	85.3%	85.1%	-0.2%
Neural Network	72.8%	72.9%	0.1%
Naïve Bayes	73.5%	73.5%	0.0%

While the Logistic Regression, Neural Network and Naïve Bayes models performed well on the test set compared to the training set, the test accuracy of KNN and Random Forest models were almost 12% higher. In the next round of testing, I chose to compare the accuracy results of the KNN and Random Forest models in order to determine which model to use on the target dataset.

Table 5: Validation Accuracy of Final Two Models

Model	Validation Accuracy
KNN	72.6%
Random Forest	83.9%

Table 6: Validation Accuracy of Final Two Models by Property Type

True Property Type	Address Count	KNN		Random Forest	
		Correct Class	Accuracy	Correct Class	Accuracy
1. Single Family Detached	511,076	469,245	92%	498,695	98%
2. Semi-Detached	27,778	1,494	5%	0	0%
3. Condo	94,345	30,557	32%	75,300	80%
4. Row	56,801	4,239	7%	10,096	18%
5. Plex	3,044	21	1%	17	1%
6. Other	3,316	26	1%	0	0%
Total	696,360	505,582	73%	584,108	84%

If I based my final model selection purely on the validation accuracy, I would have chosen Random Forest to use on the target data. However, since the model fails to classify any Semi-Detached properties, it is difficult explain why that is to non-technical stakeholders and is not acceptable given the use case of the reclassified data.

Instead, KNN was used on the target data to predict the property style.

Table 7: Distribution of Predicted Property Style by KNN on Target Data

Predicted Property Type	Address Count	% of Total
1. Single Family Detached	453,644	85%
2. Semi Detached	10,382	2%
3. Condo	47,734	9%
4. Row	21,572	4%
5. Plex	332	0%
6. Other	509	0%
Total	534,173	100%

In an attempt to validate the results of the KNN model, I chose a small sample of records from the target dataset and manually looked up their true property style using online resources like House Sigma, Zolo, Redfin, Google Maps etc. I selected properties from Toronto because I'm the most familiar with this city and it's easy to find listing data that contains the property style.

Table 8: Accuracy of Toronto Sample

True Property Type	Address Count	KNN	
		Correct Class	Accuracy
1. Single Family Detached	26	19	73%
2. Semi Detached	17	4	24%
3. Condo	57	37	65%
Total	100	60	60%

Future Considerations

Given the high test and validation accuracy performance of the Random Forest model, I would want to explore ways to improve the classification of the Semi-Detached home so that the model may be usable. Perhaps creating models for each province or other geographic regions of Canada would have made it possible for Random Forest to separate Semi-Detached from the rest of the property styles.

It would also be interesting to explore if KNN performed better in rural or urban settings. If, for example, KNN performed better in rural areas and Random Forest performed well in urban areas, then an ensemble model could be used to maximize the performance.

Property value was not used as a feature to define the property style because the value depends on time and current market conditions. For example, a $500K single family detached home in 2012 will cost the same as a one-bedroom condo in 2019 in the same neighbourhood. It would be a useful feature in the future but it is not in the scope of this project.