

**CAPSTONE PROJECT**

Assignment 4

**Analyzing and predicting rent prices in Canada**

*Professor: Samer Al-Obaidi*

*Submitted on 28th July 2024*

**Group: Ottawa**

**Team Members**

|  |  |
| --- | --- |
| **NAME** | **Student ID** |
| Tanya Nimesh | N01579781 |
| Yshika Shrikhande | N01545656 |
| Apeksha Hipparagi | N01579432 |
| Shrutika Desai | N01580471 |
| Gurleen Banga | N01579137 |

|  |  |  |
| --- | --- | --- |
| **Table of Content** | | |
| **S no.** | **Content** | **Page Number** |
| 1. | Introduction |  |
| 2. | Implementing the solution |  |
| 3. | Outcome Testing and reviewing |  |
| 4. | Optimization |  |
| 5. | References |  |

|  |  |  |
| --- | --- | --- |
| **Group contribution** | | |
| **S no.** | **Content** | **Stakeholder** |
| 1. | Implementation |  |
| 2. | Testing and reviewing |  |
| 3. | Optimization |  |

**Introduction:**

The goal of this report is to develop an accurate system for analyzing and forecasting rent prices across Canada. Leveraging data from the Canadian Government website, our team is creating a new IT architecture, as the existing system does not grant access to external parties. The implementation process involves several key steps, including data collection, data integration, and the application of machine learning techniques. This comprehensive approach ensures that our solution is well-integrated with the new IT infrastructure. The following sections detail the implementation process, including reflections and observations gathered throughout the project's development.

**1.1 Implementation of the Solution.**

Amid the vibrant and ever-changing landscape of Canada's real estate market, stakeholders sought a sophisticated tool to help them understand the intricate web of rent prices across various regions and housing types. The mission of this project was straightforward: delve into historical rent prices in regions with populations exceeding 10,000, predict future rent trends, and deliver actionable insights for real estate professionals, policymakers, and business owners alike. To tackle the pressing issues of rent affordability and the dynamics of market demand and supply, we envisioned a robust solution. The solution contains a relational database, a time series model, and an interactive dashboard. It equips stakeholders with practical, data-driven insights into affordability, market conditions, and future rent price predictions. The solution is a Regression model along with an interactive dashboard on Power BI. The interactive data dashboard will be an assessment and visualization interface of various rental housing results for the stakeholders.

Key Features of our Business Solution:

* Affordability Metrics Visualization: Rent prices compared with households’ average income across the various geographical areas.
* Regional Comparison: Highlight regions with rent prices significantly higher than the national average.
* Impact Assessment: Visualize the impact of rising rent prices on various demographic groups.
* Vacancy Rates Analysis: Show the vacancy rates and job availability in different regions.
* New Constructions Tracking: Monitor the number of new rental housing units being produced.

**Detailed Solution Design steps**

**Data Collection and Pre-processing:**

We gathered the historical rent price data (2020-2023) from The Canada Mortgage and Housing Corporation (CMHC) website. For cleaning and pre-processing the dataset, we used PowerBI. Power BI is great because it can connect to various data sources like databases, spreadsheets, and cloud services, making it easy to bring in raw data for cleaning and transformation. Inside Power BI, there is a tool called Power Query that lets you clean, reshape, remove duplicates, split columns, and apply transformations using a visual interface, so you don’t need to be a coding expert. One of the best parts is that once you have set up, you are cleaning and preprocessing steps in Power BI, and you can automate them. This means your data will automatically refresh at regular intervals or whenever new data is added. This ensures that every time you analyze your data, you are working with the most up-to-date and cleanest information available.

**Model Development:**

Categorical Features: One-hot encoding is used to convert categorical variables into a binary format.

Numerical Features: Standard scaling is used to normalize numerical data.

Model Selection: We looked at several different models to figure out which one worked best with our data. We tested out the **Gradient Boosting Regressor, K-Nearest Neighbors (KNN) Regressor, Random Forest Regressor, CatBoost Regressor, and LightGBM Regressor**. Each model went through thorough training and testing, where we split our data into 70% for training and 30% for testing to make sure the models were tested on new data they had not seen before. To see how well they did, we used Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) metrics, which helped us see how accurate and dependable our predictions were.

It seemed that the Random Forest Regressor could be among the best choices for further selection because of the trees which give accurate estimations. Its accuracy and assessment speed were favorable because one hundred estimators were employed. The seed number was selected to be a fixed random state of 42; this was because using the same seed number in the code produced the same sequence of random numbers as before, this was helpful in checking solutions against each other. After the chosen model was prepared, the prediction was done. According to the developed model, new data for the years 2025, 2026, and 2027 were created, and various scenarios with and without interaction of features for these years were made. For preparation of this new data with the same sequences for making the predictions, the same procedures were applied. This worked like an expert consultant, and thus, with the assistance of the model, forecasts of the rent prices in these years were given. For better understanding, visuals were made to allow people to easily decipher the various data available. These visualizations pointed out trends and change anticipation, providing the general view for politicians and owners of existing enterprises. These images offered understanding about future trends and displayed the future for those who made big decisions.

The following are the benefits of using Random Forest Regressor.

* Accuracy: In the examined models, the Random Forest Regressor has the lowest value of RMSE therefore this model is least deviant from actual values and better determines patterns of data set than others.
* Robustness: From the above explanation, one can deduce that Random Forest is robust and can work effectively with almost any input data type and dimensions. Moreover, it minimizes overfitting and increases the theory’s generalization because it includes several decision trees.
* Interpretability: Random Forests also provide feature importance making it easier to identify which features have the most significant impact on the predictions even though the model is not as interpretable as the linear model.
* Scalability: Random Forests do not have problems with scalability, especially if they are designed to work on parallel computations as training is computationally intensive. However, Random Forests are slightly easier to use than some of the other complicated models like Gradient Boosting because they involve lesser parameter optimization.

**Steps for Client Company Personnel to Use the Developed Solution Successfully**

1. Access to Data and Data Integration

Update the database often with the most recent information on rent prices and pertinent indicators. In Power BI, set up and keep an eye on automated data refresh schedules to make sure the dashboard always shows the most recent data.

1. Using the Interactive Dashboard

* Affordability Metrics Visualization: Use the dashboard's navigation tools to navigate to the affordability section to view the pertinent metrics visualization. Then, analyze the data by contrasting rent costs and average household income in different regions of the country to determine affordability.
* Impact Assessment: Use interactive filters to modify demographic characteristics and examine impacts based on selected criteria. Visualize the effects of rising rent costs on various demographic groups.
* Regional Comparison: Navigating to the relevant dashboard section will get you access to the regional comparison tools. Assess regions to find locations where rent is noticeably more than the national average, then compare and visualize the results.
* Track Construction: Keep an eye on the quantity of brand-new rental apartments being built, and chart emerging trends in the industry.

1. Predictive Model Utilization

* Access Predictive Insights: navigate to the area where future rent trends are provided by prediction models.
* Understand Predictions: Analyse the projected rent amounts for 2025, 2026, and 2027.

1. Generating Reports

To help with decision-making, use the dashboard's reporting tools to create custom reports based on needs or certain criteria. You can then export these reports and distribute them to the appropriate stakeholders.

1. Regular Training and Updates

Make sure all users receive frequent training so they can keep informed about new features and industry best practices.

1. Monitoring and Maintenance

Ensure the information displayed is accurate and reliable by doing regular data quality checks and monitoring the dashboard's and the underlying database's performance on a regular basis.

1. Utilizing Visualizations for Decision Making

Make use of visualizations to comprehend patterns and anticipated shifts, offering policymakers and company owners a road map. Make well-informed decisions on rent pricing strategies, policy development, and company planning by utilizing the dashboard's findings.

1. Feedback from User

Provide a feedback system so that users may report problems or offer dashboard enhancement suggestions.

**1.2 Outcome testing and reviewing**

The Random Forest Regressor, Gradient Booster, Catboost Regressor, and LightGBM Regressor that are used to build models have been thoroughly tested. With the lowest root mean square error (RMSE) of all the studied models, the Random Forest Regressor can identify data patterns most effectively and generate the most accurate predictions. Among the models examined, the Random Forest Regressor results in accurate predictions with the lowest RMSE value of 89.258, as previously mentioned. This model testing provided us with insights to help us opt for the most effective model.

RMSE enables easy interpretation since the outputs are displayed in unit of the response variable. We tried predicting Rent values for the years 2025, 2026, 2027 by the following 4 predictive models and found the following RMSE:

|  |  |
| --- | --- |
| Model | RMSE |
| Random Forest Regressor | 89.259 |
| Gradient Booster | 153.938 |
| CatBoost Regressor | 99.157 |
| LightGBM Regressor | 101.721 |

Taking all the values above into consideration, the RMSE is the lowest at **89.259** among all the models where the **Random Forest Regressor** was applied.

**Comparing the outcomes of different approach:**

This ensured that our forecasts were exact and did not allow for variations apart from the level of complexity that was understood during the preprocessing phase and evaluation of several models. It provided at least a good structure, which was open for suggestion and further enhancement if better information was obtainable. Thus, our approach gave more detailed information because more complicated forms of relations within a dataset are considered in contrast to simpler forms like linear ones. The model's performance was improved by automated feature preprocessing, which decreased human error and guaranteed consistent transformations.

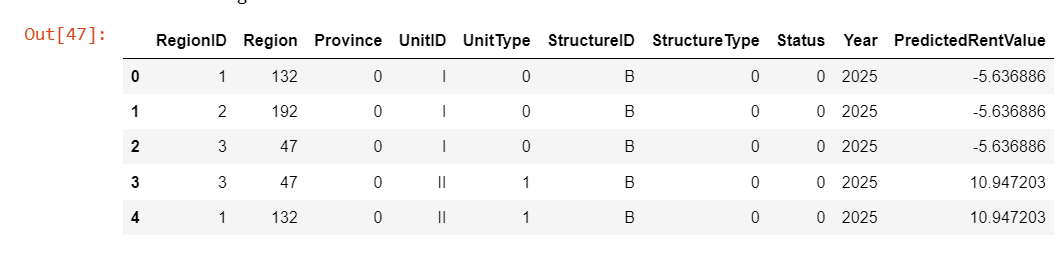
• Simple Linear Regression: Compared to ensemble techniques like Random Forest and Gradient Boosting, it may be more difficult to construct and may not be able to capture complicated relationships in the data.

• Single Model Approach: Performance may be less than ideal if a single model is used without comparison to other options. Robustness is ensured by our approach to analyzing several models.

• Manual Feature Engineering: Automated feature preprocessing using pipelines reduces human error and guarantees consistent application of transformations compared to manual feature engineering.

**Outputs for the code before adding parameters:**

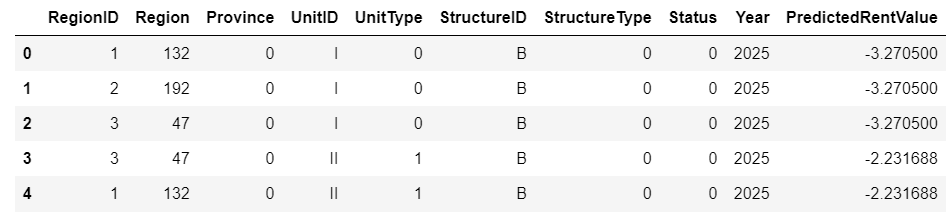
CatBoost Regressor Output



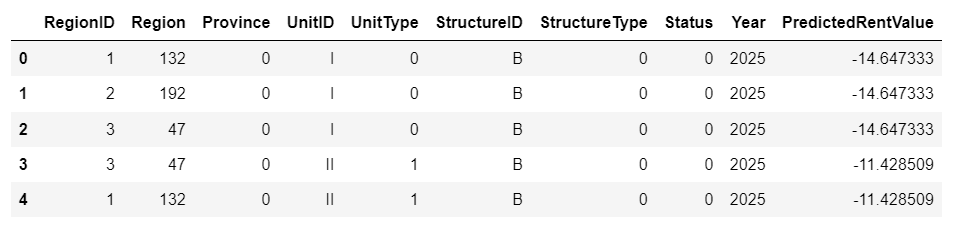
The predicted rent values in the above image show the predicted rent value for regions 1, 2 and 3 with UnitType I is -5. 636886. Both regions 1 and 3 have a predicted rent value of 10 when the type of unit involved is 10. 947203. Some of the predicted rent values for the regions are negative, which is not reasonable. It is possible to face a large difference between the value predicted for one UnitType (for example, -5. 636886 for UnitType I) and that for another one (10. 947203 for UnitType II) while the real rent trends can be different. CatBoost model have over fitted or under fitted the training data thus bad performance on new data.

The CatBoost regressor output did not match the desired output. Therefore, Random Forest regressor which is more reliable and appropriate for predicting because of its general feature is used for final predictions.

LightGBM Regressor Output

The predicted rent values, shown in the Fig 1.2 in UnitType I have predicted rent value of -3.270500 for Units in different regions of 1, 2, and 3. Region 1 and 3 with UnitType II is expected to have a value of -2.231688. Just like with the Catboost Regressor, there are negative predicted rents for all the regions. Compared to others, LightGBM incorporates new hyperparameters that require great attention to get the best results, and the whole process is very tedious. Additionally, LightGBM is a boosting method and involves building models sequentially, which can cause overfitting, if not checked correctly, particularly with noisy data sets. For the above-explained reasons and invalid outputs, this regressor was dismissed.

Gradient Boosting Regressor

Each row is a record, and looking at the ‘PredictedRentValue’ column, the values for rent prediction are negative; for example -14. 647333. But this is not possible as in investment the rent cannot be negative or it is not possible to pay back more than what was initially invested. This has inconsistency suggesting that the Gradient Boost Regressor model, is not performing up to expectations.

Thus stated, grading boosting is less interpretable than Random Forest because of the sequential nature of the algorithm as well as the interactions between boosts. In applying boosting, Gradient Boosting takes more memory space than the other algorithms because of the boosting technique used. Random Forest being parallel can be less computation intensive, or use less memory.

**1.3 Optimization**

During the testing and review of the RMSE for the Random Forest Regression model, several errors were encountered. The table below summarizes the differences between the initial code and the final corrected code, along with the errors faced:

|  |  |  |
| --- | --- | --- |
| **Code** | **Initial** | **Final** |
| Data Splitting | X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42) | X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42) |
| Model Evaluation Output | print(f"Root Mean Squared Error: {rmse}") | print(f"Random Forest Regressor Root Mean Squared Error: {rmse}") |
| New Data Preparation | unique\_combinations = df.drop(columns=['RentID', 'RentValue', 'Year']).drop\_duplicates() | unique\_combinations = df.drop(columns=['RentID', 'RentValue', 'Year']).drop\_duplicates() |
| Data Transformation and Prediction | new\_data\_preprocessed = model.named\_steps['preprocessor'].transform(new\_data)  predictions = model.named\_steps['regressor'].predict(new\_data\_preprocessed) | new\_data\_preprocessed = model.named\_steps['preprocessor'].transform(new\_data)  predictions = model.named\_steps['regressor'].predict(new\_data\_preprocessed) |

**Key Differences and Issues:**

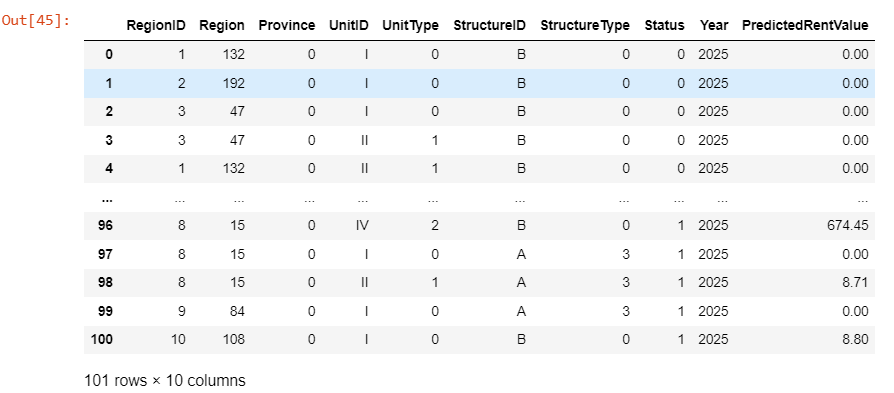
* Test Size:

The initial code was used as 20% test size, while the final code used as 30% test size. This change should not affect significantly the predicted values but might affect the evaluation metric (RMSE) slightly.

* Model Evaluation Output:

The final code specifies that the RMSE is for the Random Forest Regressor, which is a minor difference in terms of code functionality.

During the testing and review of the RMSE for the Random Forest Regression model, several errors were encountered. The table below summarizes the differences between the initial code and the final corrected code, along with the errors faced:

**Initial Code**  


**Final Code**

