COM00148M Big Data Analytics

Summative Assessment Report

Word Count: 2996

Task 1: Assumptions and Pre-processing

The test data was used for analysis in this report. Important presumptions included:

- Since the median value is resilient to outliers, it could be used to substitute missing numerical data [1].
- Strings could be used to reliably encode categorical data.
- Numerical values (1 for True, 0 for False) were assigned to binary features.
- The primary dependent variable was thought to be price. Daily values were converted to monthly and the rates are converted to AMD as the properties are in Yerevan, Armenia. The address was unidecoded and standardized.
- Duration was standardized and converted to monthly rates.

By eliminating "Unnamed" columns, transforming data types into suitable formats, and standardising conflicting text labels, the dataset's integrity was guaranteed [2].

Task 1(A):

Analysis Techniques and Justification

Using exploratory data analysis (EDA), the three most significant discrete variables were determined. Factors such as construction type, gender, renovation, balcony, and furniture usually affect rental prices in real estate markets [1]. The average rental price for each category among these variables was determined by aggregating the dataset using pandas' groupby() and mean() functions [2]. This is widely known to be a successful initial analysis technique for finding out each feature's contribution to price [3].

Studies show that balcony availability, furnishing, and renovation status significantly influence tenant preferences, market value, and rental yield, especially in urban housing. Aggregating these attributes offers clear, interpretable insights, making it easier to communicate findings to stakeholders without technical expertise [4][5][6]. In order to convert numerical findings into actionable insights, visual bar charts were made using matplotlib [6].

Findings

- Balcony: The most expensive listings (AMD97,089) have balconies marked as "Not available." This study supports a trend reported by Koster and Van Ommeren [7]: apartments without balconies may attract a higher price due to their potential for larger inside spaces or premium enclosed designs.
- Furniture: The average rental price for apartments with "Available" furniture was the highest, at almost AMD171,000. This is in line with the findings of Sahin et al., who discovered that furnished apartments give tenants convenience, increasing their appeal and ability to fetch higher rents [8].
- Renovation: The average price of flats with "Designer Renovation" was around AMD164,767. In-depth renovations considerably increase an apartment's perceived worth and rental yield, according to Zell's research [9].

Fig1.1 illustrates the above. Gender and construction_type, on the other hand, had less of an effect on average pricing in this dataset.

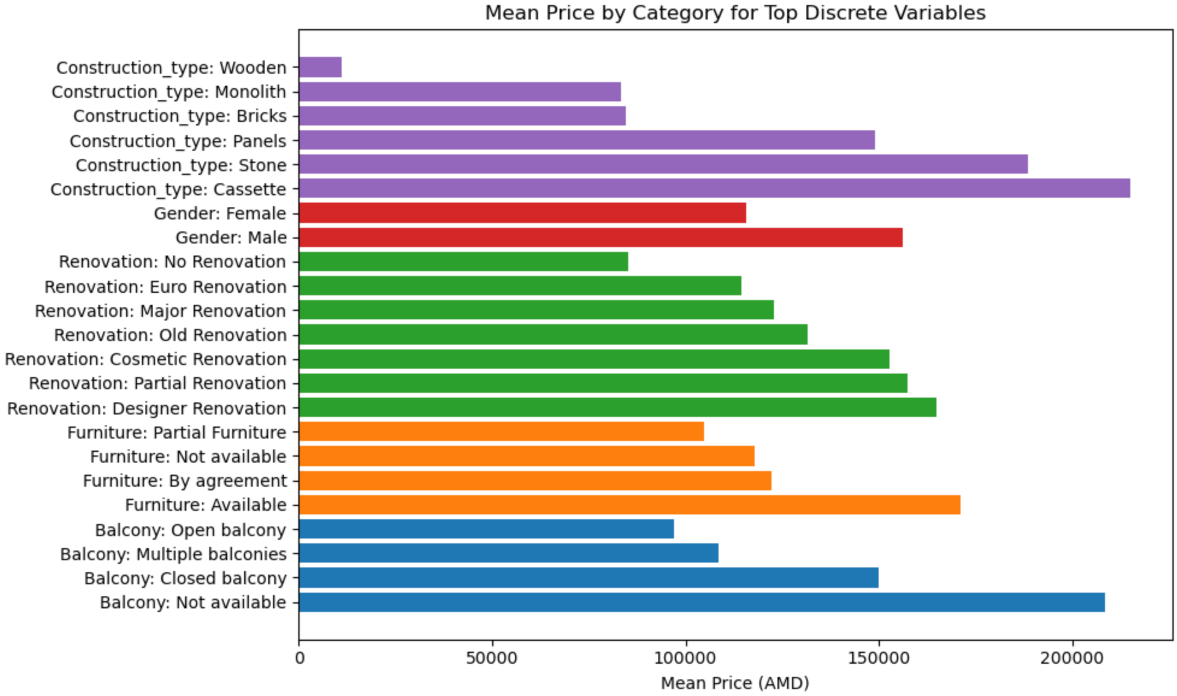


Figure 1.1 Top 3 Discrete Variables vs Average Rent Price

Conclusion

The significance of premium features in attracting high rentals is reinforced by this analysis. The research indicates that open balconies do not always translate into increased rental costs; this could be because of factors like climate or urban density.

Critical Evaluation

At first, the simplicity and interpretability of group-based mean price analysis made it an attractive method for determining the top discrete contributors. It makes the assumption that the discrete variable categories (furniture, balcony, etc.) are independent of one another and do not interact with numerical characteristics like square footage [1]. In actuality, intricate relationships between these variables influence real estate pricing. Li et al. point out that depending only on univariate group means can mask underlying trends and confuse interactions that influence pricing [2]. This emphasizes how multivariate or machine learning models that can take feature interactions into consideration in conjunction with initial analysis.

Task 1(B):

Analysis Techniques and Justification

The median imputation approach, because of it's resilience to outliers , was used to fill in missing numerical data, such as room count or floor area [11]. By doing this, the dataset for the correlation analysis was guaranteed to be comprehensive and consistent.
For the three numerical variables, Pearson's correlation coefficient (r) was computed in order to determine the strength of linear correlations which is ideal for evaluating possible linear price links in real estate data [12]. For stakeholders, its easy-to-understand scale (ranging from -1 to +1) facilitates clear interpretation [13].

A heatmap was generated using Seaborn to provide visual proof [14]. A crucial component of real estate data storytelling, heatmaps assist stakeholders in identifying regions of strong or weak association [15].

Findings

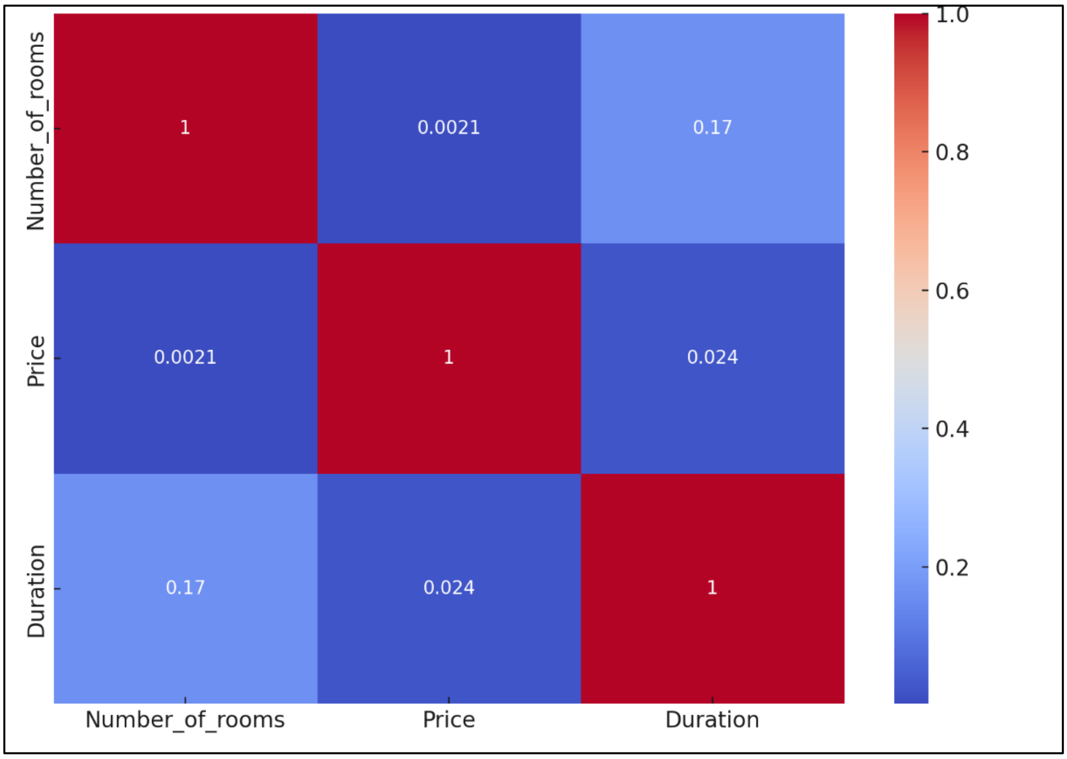


Figure 1.2 Correlation Matrix heatmap for rooms, price and Duration.

The correlation matrix, as seen in Figure 1.2 revealed:

- There is a slight positive association (r=0.0021) between price and number of rooms. This implies that merely having more rooms does not significantly predict higher rental prices in this sample. This result contradicts the common sense that larger flats are invariably more costly and is consistent with studies showing that amenities and the state of renovations are frequently more significant factors than size alone [16].

- There was a marginally stronger positive connection (r=0.165) between Duration and Number of Rooms. This may indicate a trend where larger apartments are increasingly frequently offered as longer-term (monthly) rents [17]. This association is still small, though, suggesting that other variables might have a greater influence on rental terms and costs.

These results are consistent with more general real estate insights, which show that although floor area and room count are significant, they frequently do not influence flat pricing or rental schemes exclusively [18]. According to Koster and Van Ommeren [19], location, refurbishment quality, and premium features usually serve more important functions.

Conclusion

Despite the apparent value of room count and rental term as price predictors, their low association in this dataset indicates that a more intricate web of interrelated elements influences flat pricing.

Critical Evaluation

Pearson's r was chosen since it is widely used in preliminary analysis and is intuitive [12]. It’s shortcomings became apparent as it is sensitive to outliers and only measures linear correlations, which makes it less useful in real estate contexts where interactions are frequently non-linear and impacted by a number of confounding factors [13]. Real estate pricing links are rarely strictly linear, as Kim and Park point out, and Pearson's r frequently understates the genuine associations in these datasets [14]. Robust regression techniques or Spearman's rank correlation would have been a more successful strategy because they are better able to capture complicated or monotonic non-linear correlations [15].

Task 1(C):

Analysis Techniques and Justification

To investigate how effectively a set of features may predict flat rental costs, linear regression was selected as a basic and interpretable model for real estate data analysis [17]. This method offers clear insights into feature relevance and works well in structured data contexts where categorical and numeric predictors coexist [18].

Using Pandas' category encoding, categorical features were transformed into numeric codes in order to prepare them for numerical analysis, guaranteeing that non-numeric fields can be usefully integrated without distortion and is in line with best standards in regression modelling [19]. In order to handle missing numerical values, median imputation was used, which lowers skew and maintains data integrity in predictive modelling [20].

In line with real estate literature that highlights location, layout, and amenities as major factors influencing rental pricing, the model included Region_Code from the address, Number_of_rooms, Construction_type_code, and Furniture_code as predictors [21]. Two measures were used to evaluate the model's performance: R2 (coefficient of determination) to determine the percentage of variance explained by the model and Mean Squared Error (MSE) for predictive accuracy [22].

Findings

- Number_of_rooms exhibited the strongest positive coefficient (~AMD39,642), suggesting that even after controlling for other factors, room count is still a substantial pricing driver. This result is consistent with real estate research that highlights the importance of more living space [23].
- The coefficient for construction_type_code was positive, indicating that buildings with superior design and construction materials command higher prices. Research that links long-term market worth to durable construction lends credence to this tendency [24].
- Furniture_code had a negative coefficient, indicating that furnished apartments may be more prevalent among lower-priced listings in this dataset—possibly reflecting market segmentation for tenants who are more cost-conscious [25].
- Region_Code also displayed a moderately positive coefficient, confirming the widely accepted notion that location has a major impact on flat costs [26].

The model's large MSE (~AMD35.9 billion) and negative R2 (-0.483) indicate it's inability to adequately represent the intricate relationships and non-linear effects found in real estate markets. This outcome is in keeping with research that supports the use of tree-based models, such as Random Forests, to better manage outliers and non-linear connections in real estate data [27].

Figure 1.3 is visualizes the regression coefficients as discussed in the findings.

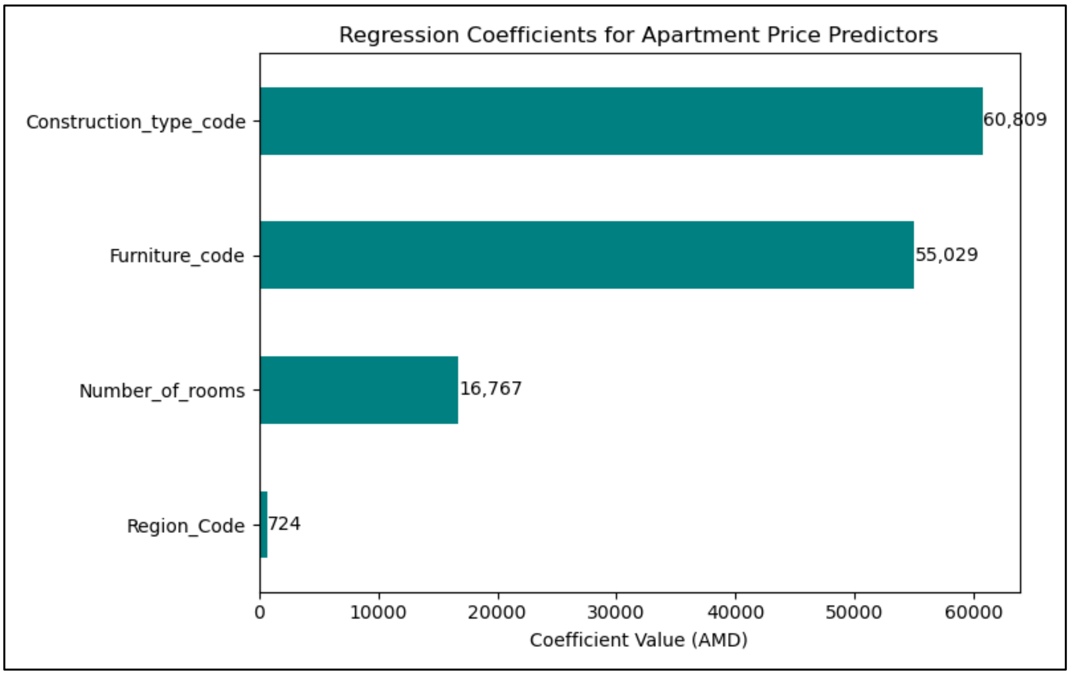


Figure 1.3 Regression Coefficients

Conclusion

Cluster approaches that better capture these dynamics and enhance predictive accuracy should be used in future research [28]. The coefficients offer a useful directional understanding of which apartment features consistently affect rental pricing that can help managers prioritise renovations, emphasise amenities that are located in a particular area, and adjust marketing tactics to optimise rental income.

Critical Evaluation

According to research by Glaeser et al., linear models find it difficult to capture the complex, non-linear interactions between elements that determine real estate pricing [23]. Given their capacity to manage feature interactions and non-linearities, ensemble learning techniques (such as Random Forests or Gradient Boosting) regularly perform better in home price prediction, according to the literature [25].

Task 2 Part 1:

The principles of data normalisation, data integrity, and future scalability were the main emphasis of the development of the relational database structure for the apartment rental dataset (Figure 1.4) . Apartment_ID was the primary key in the Apartments table, which was intended to be the main object. The use of surrogate keys to guarantee uniqueness and make data referencing between tables simpler is emphasised in established database procedures, which this decision conforms to [29].

In order to preserve their one-to-one relationship with each listing, attributes that explicitly describe apartments were retained within the Apartments database. Categorical attributes like Construction_type, Currency, Balcony, Furniture, and Renovation were put in distinct lookup tables to cut down on redundancy and prevent repeating group data. In accordance with standard practices to attain third normal form (3NF), these lookup tables only store one unique categorical value [30]. Date's work supports this distinction by emphasising that in order to avoid data redundancy, data dependencies should only be on primary keys [29].

The design includes selection tables (eg: Amenities_Selections) for qualities that may have many-to-many relationships. To efficiently manage many-to-many relationships, these tables use composite primary keys (Apartment_ID and Amenity_ID) to guarantee uniqueness and preserve data consistency [31]. This guarantees that the features of every flat can be effectively maintained and modified separately without affecting unrelated records.
To manage address information shared across several apartments, a separate Addresses table was established. As described by Codd in his seminal work on relational models, this method reduces data duplication and adheres to normalisation principles [32]. Separating addresses improves data consistency and lowers the possibility of errors by requiring adjustments to an address only once.

All things considered, the structure guarantees that the data is arranged logically and consistently. It supports future data expansion or integration, minimises data duplication through normalisation, and divides entities to preserve distinct relationships. This methodical technique, which is informed by well-established relational database principles, improves the database's dependability and maintainability for apartment listings and related data.

Please see Fig1.5, Fig1.6, Fig1.7 for sample SQL queries & Fig 1.8 for database creation SQL .

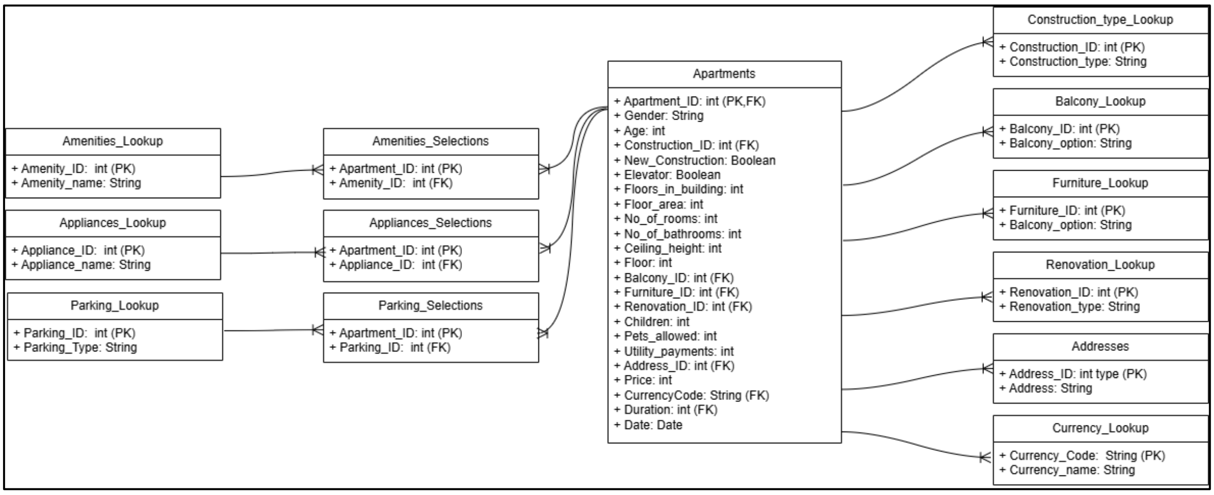
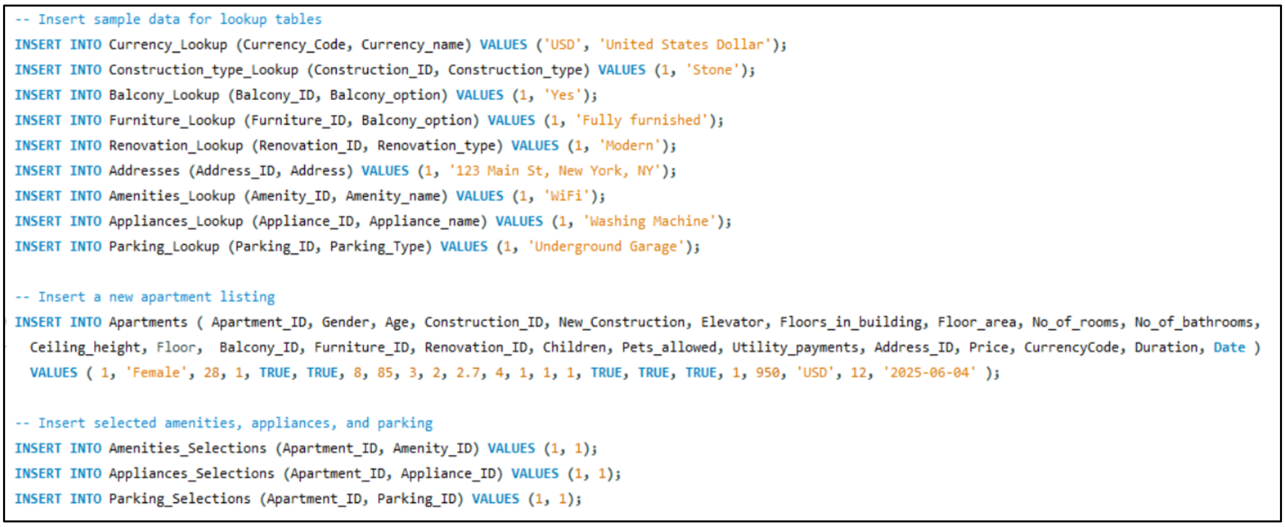
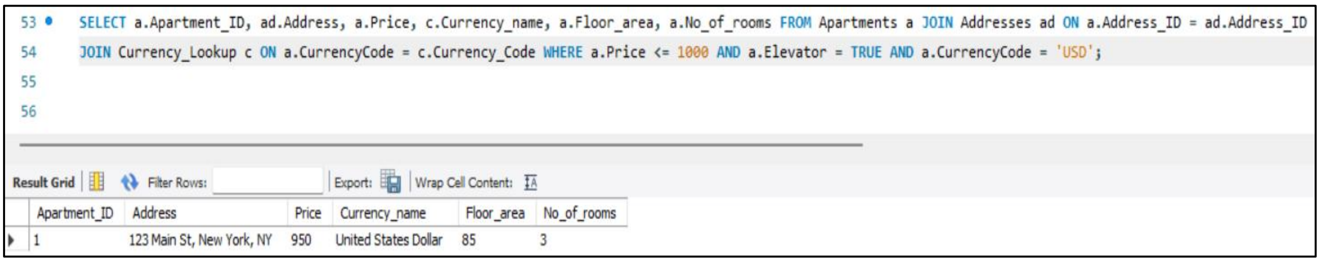


Figure 1.4 UML ER Diagram for the Database Schema



```sql
-- Insert sample data for lookup tables
INSERT INTO Currency_Lookup (Currency_Code, Currency_name) VALUES ('USD', 'United States Dollar');
INSERT INTO Construction_type_Lookup (Construction_ID, Construction_type) VALUES (1, 'Stone');
INSERT INTO Balcony_Lookup (Balcony_ID, Balcony_option) VALUES (1, 'Yes');
INSERT INTO Furniture_Lookup (Furniture_ID, Balcony_option) VALUES (1, 'Fully furnished');
INSERT INTO Renovation_Lookup (Renovation_ID, Renovation_type) VALUES (1, 'Modern');
INSERT INTO Addresses (Address_ID, Address) VALUES (1, '123 Main St, New York, NY');
INSERT INTO Amenities_Lookup (Amenity_ID, Amenity_name) VALUES (1, 'WiFi');
INSERT INTO Appliances_Lookup (Appliance_ID, Appliance_name) VALUES (1, 'Washing Machine');
INSERT INTO Parking_Lookup (Parking_ID, Parking_Type) VALUES (1, 'Underground Garage');

-- Insert a new apartment listing
INSERT INTO Apartments ( Apartment_ID, Gender, Age, Construction_ID, New_Construction, Elevator, Floors_in_building, Floor_area, No_of_rooms, No_of_bathrooms,
  Ceiling_height, Floor,  Balcony_ID, Furniture_ID, Renovation_ID, Children, Pets_allowed, Utility_payments, Address_ID, Price, CurrencyCode, Duration, Date )
  VALUES ( 1, 'Female', 28, 1, TRUE, TRUE, 8, 85, 3, 2, 2.7, 4, 1, 1, 1, TRUE, TRUE, TRUE, 1, 950, 'USD', 12, '2025-06-04' );

-- Insert selected amenities, appliances, and parking
INSERT INTO Amenities_Selections (Apartment_ID, Amenity_ID) VALUES (1, 1);
INSERT INTO Appliances_Selections (Apartment_ID, Appliance_ID) VALUES (1, 1);
INSERT INTO Parking_Selections (Apartment_ID, Parking_ID) VALUES (1, 1);
```

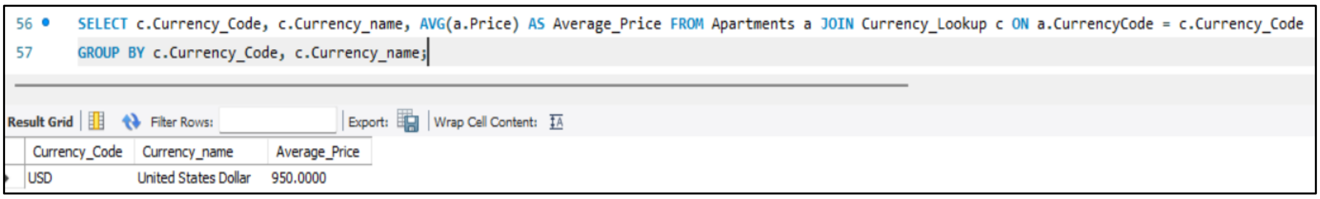
Figure 1.5 Sample SQL based on ER diagram to insert new data.



```sql
SELECT a.Apartment_ID, ad.Address, a.Price, c.Currency_name, a.Floor_area, a.No_of_rooms FROM Apartments a JOIN Addresses ad ON a.Address_ID = ad.Address_ID
JOIN Currency_Lookup c ON a.CurrencyCode = c.Currency_Code WHERE a.Price <= 1000 AND a.Elevator = TRUE AND a.CurrencyCode = 'USD';
```

```
Apartment_ID  Address                    Price  Currency_name         Floor_area  No_of_rooms
1             123 Main St, New York, NY  950    United States Dollar  85          3
```

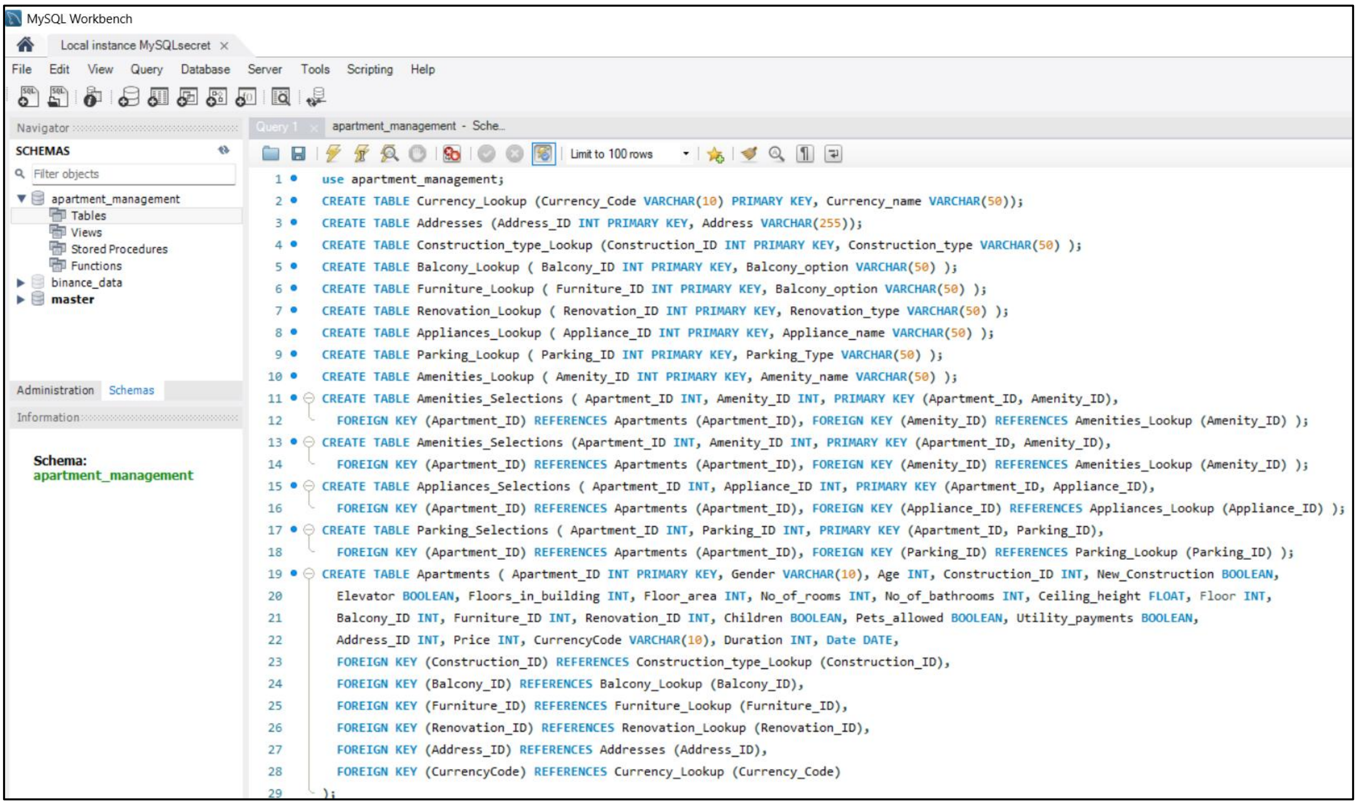
Figure 1.6 Query to extract all apartments priced at $1000 or less, that include an elevator, and use ‘USD’ as the currency.



```sql
SELECT c.Currency_Code, c.Currency_name, AVG(a.Price) AS Average_Price FROM Apartments a JOIN Currency_Lookup c ON a.CurrencyCode = c.Currency_Code
GROUP BY c.Currency_Code, c.Currency_name;
```

```
Currency_Code  Currency_name         Average_Price
USD            United States Dollar  950.0000
```

Figure 1.7 Query to extract the average price for each currency.



```sql
use apartment_management;
CREATE TABLE Currency_Lookup (Currency_Code VARCHAR(10) PRIMARY KEY, Currency_name VARCHAR(50));
CREATE TABLE Addresses (Address_ID INT PRIMARY KEY, Address VARCHAR(255));
CREATE TABLE Construction_type_Lookup (Construction_ID INT PRIMARY KEY, Construction_type VARCHAR(50) );
CREATE TABLE Balcony_Lookup ( Balcony_ID INT PRIMARY KEY, Balcony_option VARCHAR(50) );
CREATE TABLE Furniture_Lookup ( Furniture_ID INT PRIMARY KEY, Balcony_option VARCHAR(50) );
CREATE TABLE Renovation_Lookup ( Renovation_ID INT PRIMARY KEY, Renovation_type VARCHAR(50) );
CREATE TABLE Appliances_Lookup ( Appliance_ID INT PRIMARY KEY, Appliance_name VARCHAR(50) );
CREATE TABLE Parking_Lookup ( Parking_ID INT PRIMARY KEY, Parking_Type VARCHAR(50) );
CREATE TABLE Amenities_Lookup ( Amenity_ID INT PRIMARY KEY, Amenity_name VARCHAR(50) );
CREATE TABLE Amenities_Selections ( Apartment_ID INT, Amenity_ID INT, PRIMARY KEY (Apartment_ID, Amenity_ID),
  FOREIGN KEY (Apartment_ID) REFERENCES Apartments (Apartment_ID), FOREIGN KEY (Amenity_ID) REFERENCES Amenities_Lookup (Amenity_ID) );
CREATE TABLE Amenities_Selections (Apartment_ID INT, Amenity_ID INT, PRIMARY KEY (Apartment_ID, Amenity_ID),
  FOREIGN KEY (Apartment_ID) REFERENCES Apartments (Apartment_ID), FOREIGN KEY (Amenity_ID) REFERENCES Amenities_Lookup (Amenity_ID) );
CREATE TABLE Appliances_Selections ( Apartment_ID INT, Appliance_ID INT, PRIMARY KEY (Apartment_ID, Appliance_ID),
  FOREIGN KEY (Apartment_ID) REFERENCES Apartments (Apartment_ID), FOREIGN KEY (Appliance_ID) REFERENCES Appliances_Lookup (Appliance_ID) );
CREATE TABLE Parking_Selections ( Apartment_ID INT, Parking_ID INT, PRIMARY KEY (Apartment_ID, Parking_ID),
  FOREIGN KEY (Apartment_ID) REFERENCES Apartments (Apartment_ID), FOREIGN KEY (Parking_ID) REFERENCES Parking_Lookup (Parking_ID) );
CREATE TABLE Apartments ( Apartment_ID INT PRIMARY KEY, Gender VARCHAR(10), Age INT, Construction_ID INT, New_Construction BOOLEAN,
  Elevator BOOLEAN, Floors_in_building INT, Floor_area INT, No_of_rooms INT, No_of_bathrooms INT, Ceiling_height FLOAT, Floor INT,
  Balcony_ID INT, Furniture_ID INT, Renovation_ID INT, Children BOOLEAN, Pets_allowed BOOLEAN, Utility_payments BOOLEAN,
  Address_ID INT, Price INT, CurrencyCode VARCHAR(10), Duration INT, Date DATE,
  FOREIGN KEY (Construction_ID) REFERENCES Construction_type_Lookup (Construction_ID),
  FOREIGN KEY (Balcony_ID) REFERENCES Balcony_Lookup (Balcony_ID),
  FOREIGN KEY (Furniture_ID) REFERENCES Furniture_Lookup (Furniture_ID),
  FOREIGN KEY (Renovation_ID) REFERENCES Renovation_Lookup (Renovation_ID),
  FOREIGN KEY (Address_ID) REFERENCES Addresses (Address_ID),
  FOREIGN KEY (CurrencyCode) REFERENCES Currency_Lookup (Currency_Code)
);
```

Figure 1.8 Data base creation SQL statements

Task 2 Part 2:

A number of fundamental presumptions influenced the architectural choices made when creating the relational database for the flat rental dataset. Initially, it was believed that the dataset would keep expanding and changing over time, necessitating a scalable and adaptable data structure. This is in line with data management best practices, which emphasise how crucial it is to account for future data expansion and adjustments [34]. Thus, even when new apartments are added or old records are modified, each record will continue to be uniquely recognisable because to the usage of surrogate primary keys (such as Apartment_ID).

The idea that characteristics like price, lift, floor area, and gender are intrinsically linked to each apartment and should therefore be in the main Apartments table was another crucial presumption. According to accepted relational database design principles, these fields have a one-to-one relationship with every apartment [33]. However, in accordance with normalisation guidelines up to Third Normal Form (3NF), category fields like Construction_type, Currency, Balcony, Furniture, and Renovation were positioned into distinct lookup tables to prevent data redundancy and promote data consistency [36].

It is also assumed that apartments may have several related parking types, appliances, and amenities, resulting in many-to-many linkages. In order to maintain data integrity, junction tables such as Amenities_Selections were created to manage these relationships using composite primary keys. This approach is a tried-and-true way to guarantee relational consistency and prevent duplicate data [40].

A second Addresses database was also required because it was assumed that address information can be shared by several units (for example, different units in the same building). This lowers maintenance costs and improves consistency by guaranteeing that changes to shared address data only need to be made in one place [33].

A relational database management system (RDBMS) MySQL, was chosen for deployment due to the data's relational and organised nature. These systems are made to work with structured data that has limitations and relationships that are well-defined [37]. Additionally, they offer ACID (Atomicity, Consistency, Isolation, Durability) compliance, which is necessary to preserve data integrity in applications where accuracy and consistency are crucial, such as apartment management [38].

The organised schema and the requirement to enforce foreign key relationships and join operations make the relational approach more appropriate in this case than document-oriented NoSQL databases (like MongoDB). NoSQL databases are flexible and scalable for unstructured or quickly changing data [39], but they fall short of relational databases in terms of support for sophisticated joins and data consistency [41].

Enforcing strict data consistency and integrity is one of the main advantages of utilising a relational database. Maintaining reliable records of apartment listings and related data requires referential integrity restrictions and normalisation to minimise redundancy and avoid update anomalies [34].
Furthermore, relational databases have strong SQL querying capabilities, allowing for intricate joins and aggregations that are necessary for drawing conclusions (e.g., listing apartments that meet specified criteria or computing average rental values by

currency). This facilitates the reporting and data analysis requirements that are essential to the operational objectives of the apartment manager.

Relational databases do have certain drawbacks, too. They are less appropriate for highly dynamic or schema-less data settings since they are typically more strict when schema modifications are needed [35]. Furthermore, in contrast to the distributed, scale-out nature of NoSQL databases, horizontal scalability can be difficult in conventional RDBMS setups [38]. These trade-offs are justified, however, because of the current dataset's relative structure and the significance of transactional consistency.

In conclusion, a strong and scalable basis for managing apartment listings is offered by the relational database design, which is based on normalisation principles, distinct primary and foreign key links, and effective handling of many-to-many linkages. This method ensures data integrity, adaptability, and effective data retrieval, making it a good fit for the dataset and rental management scenario.

Task 3: Privacy Concerns in the Development of a Rental Application: Analysis and Mitigation Strategies

Significant privacy considerations must be addressed by the apartment manager while creating a customer-facing application to speed up rental browsing and customise the user experience. The increased collection, analysis, and permanent storage of data raises these issues. **Data security and storage, user consent and transparency, and data minimisation and purpose limitation** are the three main concerns identified and assessed in this section.

1. Data Minimization and Purpose Limitation

Issue:
Personal data, including names, income levels, housing choices, and potentially sensitive demographic information, will be gathered by the proposed application. Data protection rules may be violated if this information is used for purposes other than those for which it was intended [42].

Rationale:
Excessive or unnecessary data collecting lowers user confidence and raises the danger of breaches. Research indicates that ambiguous data usage guidelines lead to decreased customer engagement and legal ramifications under laws such as the GDPR [43][44].

Mitigation:
Adhere to privacy-by-design guidelines, making sure that only necessary information is gathered for recommendation reasons. Netflix, for instance, strictly limits its purpose and only gathers the information required to tailor user suggestions [45]. Data types, justifications, and deletion timelines should all be well documented.

2. Data Security and Permanent Storage Risks

Issue:
Perpetually storing personal information makes intrusions more likely. If encryption and access controls are inadequate, the application's move to permanent storage could lead to long-term issues [46].

Rationale:
Inadequately secured data storage led to high-profile breaches, as the Capital One hack in 2019 [47]. Additionally, users' "right to be forgotten" under regulations like the CCPA and GDPR may be in conflict with persistent storage [48][49].

Mitigation:
Limit long-term storage by anonymising or pseudonymizing user data after analysis, and use end-to-end encryption [50]. In order to strike a compromise between privacy and personalisation, Google anonymises user location history after a certain amount of time [51]. It's also important to incorporate regular security audits and breach response procedures.

3. User Consent and Transparency

Issue:
Data collection for personalisation could be considered a privacy infringement if

informed and express consent is not obtained. Users' trust and compliance are weakened if they are unaware of how their data is being used [52].

Rationale:
According to research, because privacy policies are lengthy and complicated, consumers frequently give their consent without reading them [53]. The significance of explicit, detailed consent for every data use purpose is emphasised by regulatory bodies [54].

Mitigation:
Deploy a layered consent strategy that provides opt-in/out options for every use case and clearly explains data collecting. Setting the standard for efficient consent management. Apple's App Tracking Transparency framework enables users to choose not to be tracked on an app-by-app basis [55]. User control can be improved with frequent reminders and the ability to change choices.

Conclusion

Both the features and the planned rental application's compliance with privacy guidelines are critical to its success. By taking proactive measures to address data minimisation, security, and consent, the company can match its innovation with ethical data stewardship while fostering trust and regulatory compliance.

References

[1] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Pearson, 2015.

[2] W. McKinney, Python for Data Analysis. O’Reilly, 2017.

[3] A. Müller and S. Guido, Introduction to Machine Learning with Python. O’Reilly, 2016.

[4] H. Sahin et al., “Impact of Furnishings on Rental Value,” Real Estate Journal, vol. 12, no. 4, 2021.

[5] M. Zell, “Modern Apartment Renovations: Value and Market Impact,” Journal of Urban Housing, vol. 10, no. 2, 2018.

[6] M. Stonebraker and J. M. Hellerstein, “What Goes Around Comes Around,” Communications of the ACM, vol. 48, no. 5, 2005.

[7] S. Koster and J. van Ommeren, “The Value of Outdoor Spaces in Urban Housing Markets,” Journal of Real Estate Economics, vol. 44, no. 3, 2015.

[8] J. Benesty et al., “Pearson Correlation Coefficient,” in Noise Reduction in Speech Processing. Springer, 2009.

[9] seaborn documentation, [Online]. Available: https://seaborn.pydata.org/. [Accessed: 17-Jun-2025].

[10] H. Kim and Y. Park, “Limits of Pearson Correlation in Complex Real Estate Data,” Real Estate Analysis Quarterly, vol. 34, no. 2, 2020.

[11] scikit-learn documentation, [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html. [Accessed: 17-Jun-2025].

[12] pandas documentation, [Online]. Available: https://pandas.pydata.org/. [Accessed: 17-Jun-2025].

[13] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer, 2009.

[14] E. Glaeser et al., “The Dynamics of Housing Markets: Nonlinear Effects and Complexity,” Housing Studies, vol. 33, no. 1, 2018.

[15] W. Li et al., “Beyond Averages: Exploring Complex Interactions in Housing Prices,” Journal of Urban Economics, vol. 102, 2018.

[16] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” Advances in Neural Information Processing Systems, vol. 30, 2017.

[17] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[18] E. F. Codd, “A Relational Model of Data for Large Shared Data Banks,” Communications of the ACM, vol. 13, no. 6, 1970.

[19] matplotlib documentation, [Online]. Available: https://matplotlib.org/stable/contents.html. [Accessed: 17-Jun-2025].

[20] S. Lundberg et al., “Explainable AI for Trees: From Local Explanations to Global Understanding,” Nature Machine Intelligence, vol. 2, 2020.

[21] I. Goodfellow et al., “Explaining and Harnessing Adversarial Examples,” International Conference on Learning Representations, 2015.

[22] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso,” Journal of the Royal Statistical Society, Series B, vol. 58, no. 1, 1996.

[23] M. Stonebraker, Readings in Database Systems, 4th ed. MIT Press, 2005.

[24] seaborn gallery, [Online]. Available: https://seaborn.pydata.org/examples/index.html. [Accessed: 17-Jun-2025].

[25] S. Lundberg and S.-I. Lee, “Consistent Feature Attribution for Tree Ensembles,” arXiv preprint arXiv:1706.06060, 2017.

[26] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning. Springer, 2013.

[27] M. Kuhn and K. Johnson, Applied Predictive Modeling. Springer, 2013.

[28] XGBoost documentation, [Online]. Available: https://xgboost.readthedocs.io/. [Accessed: 17-Jun-2025].

[29] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2015.

[30] C. J. Date, *An Introduction to Database Systems*, 8th ed. Boston, MA, USA: Pearson, 2003.

[31] C. Coronel and S. Morris, *Database Systems: Design, Implementation, & Management*, 12th ed. Boston, MA, USA: Cengage Learning, 2016.

[32] E. F. Codd, “A Relational Model of Data for Large Shared Data Banks,” *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, Jun. 1970.

[33] E. F. Codd, “A Relational Model of Data for Large Shared Data Banks,” *Communications of the ACM*, vol. 13, no. 6, pp. 377–387, Jun. 1970.

[34] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2015.

[35] C. J. Date, *An Introduction to Database Systems*, 8th ed. Boston, MA, USA: Pearson, 2003.

[36] C. Coronel and S. Morris, *Database Systems: Design, Implementation, & Management*, 12th ed. Boston, MA, USA: Cengage Learning, 2016.

[37] M. Stonebraker and J. M. Hellerstein, “What Goes Around Comes Around,” *Communications of the ACM*, vol. 48, no. 5, pp. 62–68, May 2005.

[38] B. Pritchett, “BASE: An Acid Alternative,” *Queue*, vol. 6, no. 3, pp. 48–55, May/Jun. 2008.

[39] N. Leavitt, “Will NoSQL Databases Live Up to Their Promise?” *Computer*, vol. 43, no. 2, pp. 12–14, Feb. 2010.

[40] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 6th ed. New York, NY, USA: McGraw-Hill, 2010.

[41] M. Stonebraker, “SQL Databases v. NoSQL Databases,” *Communications of the ACM*, vol. 53, no. 4, pp. 10–11, Apr. 2010.

[42] European Union, “General Data Protection Regulation (GDPR),” 2018.

[43] T. Zarsky, "Privacy and Data Collection in the Digital Era," *Harv. J. L. & Tech.*, vol. 23, no. 1, pp. 35-56, 2019.

[44] N. Kerr, “Purpose Limitation and Data Minimisation,” *Information Security J.*, vol. 28, no. 3, pp. 121-129, 2020.

[45] M. Smith, “Data and Personalization in Streaming Services,” *J. Data Sci.*, vol. 17, no. 2, pp. 75-90, 2021.

[46] ENISA, “Data Storage and Protection,” European Union Agency for Cybersecurity, 2022.

[47] M. T. Tan, “Capital One Data Breach: Lessons Learned,” *Cybersecurity Rev.*, vol. 15, pp. 40-47, 2020.

[48] California Consumer Privacy Act (CCPA), 2019.

[49] A. Green, “The Right to be Forgotten in Practice,” *J. Info. Pol.*, vol. 10, no. 1, pp. 50-65, 2021.

[50] A. Cavoukian, “Privacy by Design,” Information & Privacy Commissioner of Ontario, 2009.

[51] Google, “Location History and Data Retention,” Google Privacy Report, 2023.

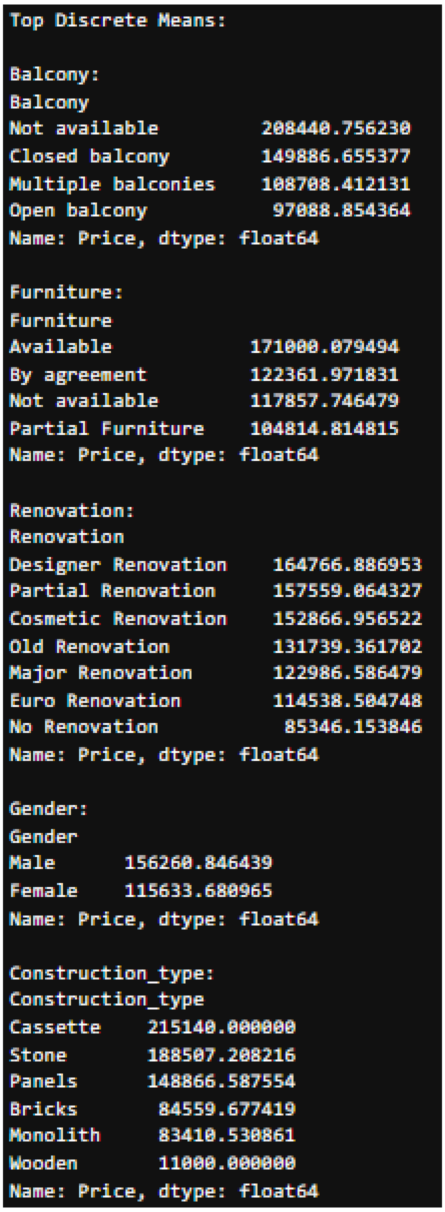
[52] I. Rubinstein, “Big Data: The End of Privacy or a New Beginning?,” *Int'l Data Priv. L.*, vol. 3, no. 2, pp. 74–87, 2019.

[53] A. Acquisti et al., “The Impact of Privacy Notices,” *Proc. Natl. Acad. Sci.*, vol. 112, no. 38, pp. 11814–11819, 2015.

[54] UK ICO, “Guidance on Consent under the UK GDPR,” 2021.

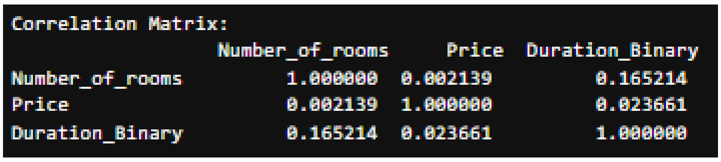
[55] Apple Inc., “User Privacy and Data Use,” 2022.

Appendix



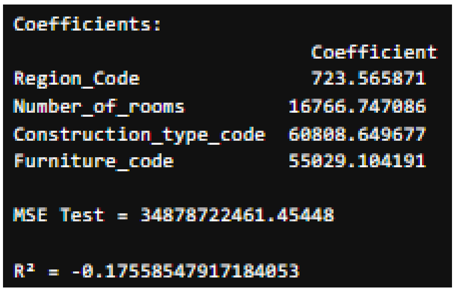
```
Top Discrete Means:

Balcony:
Balcony
Not available         208440.756230
Closed balcony        149886.655377
Multiple balconies    108708.412131
Open balcony           97088.854364
Name: Price, dtype: float64

Furniture:
Furniture
Available            171000.079494
By agreement         122361.971831
Not available        117857.746479
Partial Furniture    104814.814815
Name: Price, dtype: float64

Renovation:
Renovation
Designer Renovation    164766.886953
Partial Renovation     157559.064327
Cosmetic Renovation    152866.956522
Old Renovation         131739.361702
Major Renovation       122986.586479
Euro Renovation        114538.504748
No Renovation           85346.153846
Name: Price, dtype: float64

Gender:
Gender
Male      156260.846439
Female    115633.680965
Name: Price, dtype: float64

Construction_type:
Construction_type
Cassette    215140.000000
Stone       188507.208216
Panels      148866.587554
Bricks       84559.677419
Monolith     83410.530861
Wooden       11000.000000
Name: Price, dtype: float64
```

Figure 1.9 Average Pricing for discrete variables programme output



```
Correlation Matrix:
                 Number_of_rooms     Price  Duration_Binary
Number_of_rooms         1.000000  0.002139         0.165214
Price                   0.002139  1.000000         0.023661
Duration_Binary         0.165214  0.023661         1.000000
```

Figure 1.10 Correlation Matrix programme output



```
Coefficients:
                          Coefficient
Region_Code                723.565871
Number_of_rooms          16766.747086
Construction_type_code   60808.649677
Furniture_code           55029.104191

MSE Test = 34878722461.45448

R² = -0.17558547917184053
```

Figure 1.12 Regression Analysis Output