Analyzing Declining Rental Housing Demand: Identifying Problems and Solutions Using Data Mining Techniques

Content:

Task 1: Rental “demand” investigation:

Assumptions:

In formulating the assumptions for the analysis, initial considerations centered around the belief that attributes such as "rent" and "sqfeet" would serve as primary determinants influencing property demand. Additionally, it was anticipated that factors like "wheelchair_access" and "cats_allowed" might have weaker individual influences on demand. Consequently, attribute engineering strategies were planned to explore whether combining these discrete attributes could enhance their predictive power. These assumptions laid the groundwork for a structured approach to understanding the intricate dynamics of housing demand and informed subsequent preprocessing.

Preprocessing:

In preprocessing the datasets, a Python application (Appendix A.1) was built using Tkinter [1] and Pandas [2] libraries in order to facilitate initial cleaning before further processing in Weka. For the training dataset, the application efficiently replaced 0,1 and missing values in the "rent" column with the average rent value using the engineered attribute 'price_per_sqft' for each region, ensuring data consistency and accuracy; The same steps were taken for missing values in the “sqfeet” attribute and removal of 20 instances with square footage under 100 square feet for data validity. Despite discovering 6000 duplicate entries, these instances were retained due to unique IDs and uncertainty about their nature. Notably, instances with values of "n" and "0" in the "demand" attribute, as well as those with extra space before the "no" value, were replaced with "no". Furthermore, instances with missing values or values other than "yes" or "no"(e.g., Invalid URLs) in the "demand" attribute were removed, resulting in the elimination of 689 instances. The test data underwent similar preprocessing, with the exception of artificially calculating values. These steps ensured data quality and prepared it for analysis, aligning with the investigation's objectives [3][4].

In the pursuit of enhancing the dataset for housing demand prediction, attribute engineering strategies were employed creating the attributes 'total_rooms' combining bedrooms and bathrooms, 'pet_friendly_score' reflecting pet-friendliness by aggregating individual pet allowances, and 'accessibility_score' combining wheelchair access and electric vehicle charging. Furthermore, the description attribute was replaced with word counts to explore its relationship with demand (Appendix A.1). These machine learning and data mining-based methodologies enrich the dataset for accurate housing demand predictions [5].

During the preliminary examination of the training dataset within the Weka environment, a salient disparity was observed: merely 2.5 percent of the dataset contained the "no" value for the demand attribute, indicating a pronounced imbalance in the dataset. To address this issue, a strategic combination of down-sampling and up-sampling techniques was adopted [6][7]. Initially, the NumericToBinary filter was applied to convert numeric variables such as

"cats_allowed" into binary form, ensuring compatibility with subsequent processing steps. Following this, the Synthetic Minority Over-sampling Technique (SMOTE) filter, integrated into the Weka package manager, was employed to augment instances with "no" values for demand from 941 to 7528 artificially [6]. To mitigate potential errors associated with up-sampling, the NumericTransform filter, with the Math.floor method, was applied to attributes such as "bedrooms" and "bathrooms". Subsequently, the SpreadSubSample filter, configured with a distribution spread of 3, was utilized to reduce the number of instances with "yes" values for demand to 22584 instances, effectively balancing the dataset [8]. Lastly, the Randomize filter was applied to shuffle the instances randomly across the dataset, ensuring uniform distribution and mitigating potential biases [9]. This meticulous preprocessing approach aimed to alleviate the imbalanced class distribution while preserving the integrity and representativeness of the dataset for subsequent analysis and modeling tasks (Appendix A.2).

For each of the three tasks, further specific preprocessing steps were required, leading to the creation of three distinct sets of ARFF files excluding the main set. The subsequent sections of the report will detail the preprocessing methods undertaken for each task, providing insights into the tailored approaches adopted to address the analytical requirements.

Task 1.a. Analysis of Discrete Variables in Predicting Property Demand:

The analysis aimed at identifying discrete variables with predictive potential for both "low demand" and "high demand" properties utilizing four distinct methods: Support Vector Machines (SMO), chosen for their ability to handle high-dimensional data and classify instances accurately [10], J48 decision trees, for uncovering interpretable patterns within the data [3], Random Forest, to give a point of comparison with enhanced generalization [11], and Naive Bayes as a probabilistic classifier, complementing the methodology mix [12]. The combination of these methods allowed for a comprehensive exploration of discrete variables' predictive power, ultimately illuminating nuanced relationships between property attributes and demand [13-15]. Primarily, all non-discrete attributes were removed from the datasets (refer to arffs/task1.a). Following the training of each method and evaluation on the test set, it became evident that and Random Forest initially exhibited superior performance on the training data, correctly classifying 99.69 percent of instances (Appendix A.3) compared to 99.14 percent on J48 and SMO and 98.14 percent on Naïve Bayes [13]. However, closer scrutiny revealed a degree of overfitting when applied to the test set, prompting a more cautious interpretation of its efficacy.

In contrast, SMO and J48 consistently demonstrated robust prediction results, boasting a 99 percent success rate and great precision and recall [10] (Appendix A.4). Subsequent scrutiny of SMO's results highlighted the attributes "bedrooms," "total_rooms," and "bathrooms" as having the highest weights (Appendix A.5) [14]. These variables were selected for further analysis, with J48 decision trees revealing that the number of bedrooms emerged as the primary predictive factor for demand (Appendix A.6). Houses with more than three bedrooms exhibited a notable negative demand, followed by the total number of rooms as the secondary determinant.

Additionally, a comprehensive analysis using the CorrelationEval method unveiled weak positive correlations between demand and both the “pet_friendly_score” and the description word count (Appendix A.7). This multi-method approach not only provided nuanced insights into the predictive potential of discrete variables but also illuminated the interplay and relationships among these variables and the demand attribute [15].

While the analysis successfully identified discrete variables associated with predicting low demand properties, determining their potential to predict high demand properties involves a more nuanced understanding of market dynamics, buyer preferences, and external factors impacting demand. Variables like location, proximity to amenities, market trends, economic conditions, and demographic shifts may exert a more substantial influence on predicting high demand properties.

Task 1.b. Correlation between demand, rent and type:

The task aimed to determine the presence of a correlation, either positive or negative, between the demand for a property and its rent and type attributes. To conduct the analysis, all attributes except rent, type, and demand were removed from the dataset. One-hot encoding was applied to the "type" attribute using the nominalToBinary filter, to transform categorical variables into binary vectors and facilitate the evaluation of correlations between each type and demand (refer to arffs/task1.b).

Subsequently, the correlationAttributeEval with the ranker method was employed as the evaluator in the select attributes tab [15]. The results indicated that the type attribute, particularly the house type, exhibited the highest correlation with demand, with a coefficient of 0.67(Appendix A.8). Following this, the apartment type demonstrated a correlation coefficient of 0.59, indicating a significant association with demand. Conversely, the correlation coefficient for rent was 0.27, suggesting a moderate correlation with demand [16].

Further analysis through visualization revealed that demand exhibited a positive correlation with the apartment type and a negative correlation with the house type and rent (Appendix A.9). Notably, the property type emerges as a significant determinant of demand [17].

This implies that apartments are more likely to be associated with higher demand, whereas houses tend to be linked with lower demand. Furthermore, these findings align with the observation that properties with more than three bedrooms predict negative demand. The interplay between the number of bedrooms and property type adds complexity to the dynamics, suggesting that larger homes, particularly houses, might contribute to decreased demand.

Task 1.c. Analysis of Optimal Square Footage Range for High Demand Properties:

The task aimed to identify the optimal range of property size ("sqfeet") for generating high demand. For preprocessing, the Discretize filter[18] was employed with the useEqualFrequency option, dividing "sqfeet" values into ranges. The initial objective involved determining the optimum number of bins by evaluating the Information Gain using the InfoGainAttributeEval, considering values from 3 to 15 bins.

The results indicated that the optimal number of bins, yielding the highest Information Gain without the risk of overfitting, was found to be 6, with a score of 0.443(refer to arffs/task1.c). Subsequently, the J48 classifier was applied, and the resulting decision tree analyzed (Appendix A.10). Notably, the tree predicted any "sqfeet" value below 1709 (the first 5 bins) as indicative of high demand [19].

The performance metrics of the J48 classifier [3] provided insightful metrics indicating that the classifier achieved a relatively high level of accuracy (87.58%) in predicting high demand based on the defined "sqfeet" ranges.

However, upon closer examination of the visualization of the bins, it was observed that Bin 5, covering the range from 1254 to 1709, exhibited an equal distribution between negative and positive demand(Appendix A.10). This, coupled with a thorough review of the instances in the dataset, led to the conclusion that the optimal range for "sqfeet" associated with high demand would likely fall between 200 and 1200.

In summary, the integrated approach involving Information Gain evaluation, J48 classification, and visualization resulted in a nuanced understanding of the optimal "sqfeet" range for generating high demand, providing valuable insights for decision-making in the real estate domain.

Task 2: Storing data and possible solutions

Part 1: Designing a relational database

In designing the relational database to store the provided 'Housing' dataset, several considerations were taken into account to ensure efficient data management, scalability, and adherence to normalization principles.

One key feature of the database design and ER diagram (Appendix B.1), is the separation of entities such as Location, Laundry Option, and Parking Option [20]. This decision was motivated by the need to reduce redundancy and improve data integrity. Upon analyzing the original dataset, it was observed that around 20% of the data contained duplicates, suggesting the presence of multiple housing units in the same buildings. By separating the Location entity, we can effectively capture the unique geographic coordinates (latitude and longitude) for each distinct location, accommodating multiple housing units within the same vicinity while maintaining data integrity [21].

Furthermore, separating Laundry Option and Parking Option into their own entities (Appendix B.1) enhances scalability and flexibility [23]. As these attributes may have multiple values and can vary across different housing units, maintaining them as separate entities allows for easier management and modification. For instance, new laundry or parking options can be added without altering the core structure of the database, ensuring adaptability to changing requirements (Appendix B.2).

The normalization process was pivotal in refining the database structure and minimizing data redundancy. Specifically, attributes were organized into distinct entities to eliminate data anomalies and ensure data consistency [22]. For example, in order to find specific housing units having a normalized database facilitates data retrieval by writing queries where you address entities separately in order to find relevant entries such as finding a unit that allows cats and dogs with rent equal to or less than 1000 dollars in the state of California (Appendix B.3).

Moreover, careful consideration was given to selecting primary keys for each entity. Primary keys, such as 'id' in the House entity and composite keys for the Location entity, were chosen to uniquely identify each record within the table. This ensures data integrity and enables efficient indexing and querying operations [24]. One example of this is the query for average rent of each state which demonstrates the efficiency of using primary and foreign keys in organizing our selection and efficient data retrieval (Appendix B.4).

In summary, the relational database design prioritizes data integrity, scalability, and normalization principles. By separating entities, selecting appropriate primary keys, and adhering to normalization standards, the database structure provides a robust framework for storing and managing housing data effectively, facilitating informed decision-making in the domain.

Part 2: Scaling the database

In response to the housing manager's international expansion goals, adopting a cloud-based relational database management system (RDBMS) emerges as a robust solution for efficiently handling megabyte-scale datasets [25]. This proposal advocates for leveraging Amazon Web Services (AWS) RDS [26], coupled with PostgreSQL [27], a powerful open-source RDBMS renowned for its reliability, scalability, and extensive feature set.

The AWS RDS solution with PostgreSQL offers scalability by providing scalable compute and storage resources, enabling the housing manager to adjust capacity based on workload demands [28]. For instance, as the dataset grows with property listings, including rent, square footage, and amenities, PostgreSQL's scalability ensures seamless handling of these variables across various international locations.

PostgreSQL's ACID compliance ensures data integrity and reliability, critical for managing housing data across diverse global offices. With features like transactions and constraints, PostgreSQL enforces data consistency and integrity at the database level. In a dynamic market, frequent updates to property listings necessitate a robust transactional system [29]. PostgreSQL's support for transactions ensures atomic modifications, maintaining data consistency across various locations [36].

Real-time event processing can be seamlessly integrated with PostgreSQL using AWS services such as Amazon Kinesis Data Streams [30] or AWS Lambda [31]. To illustrate, suppose the automated analysis detects a significant increase in the count of rental properties exceeding a predetermined threshold in a particular region. With the integration of AWS services and PostgreSQL, the system can swiftly recognize this event and trigger immediate actions. AWS Lambda functions can process the analysis results in real-time, determining if the threshold has been surpassed [32]. If so, the system can automatically generate and send notifications to relevant stakeholders, alerting them to the surge in demand. This enables the business to promptly respond to changing market dynamics and make informed decisions to capitalize on emerging opportunities or address potential challenges.

PostgreSQL's robust feature set and strong community support make it an ideal choice for managing housing data. Its support for complex data types, indexing mechanisms, and transaction management capabilities provide the necessary tools for efficient data management and analysis [33].

When comparing the proposed cloud-based RDBMS solution with alternative approaches, such as flat-file systems and traditional relational databases, notable differences emerge. Unlike flat-file systems, which often lack scalability and robust data management features, the AWS RDS with PostgreSQL solution offers dynamic scalability and comprehensive data management capabilities tailored to the housing dataset's complexities. Similarly, while containerized RDBMS

solutions offer portability and resource efficiency [34]., they may not match the scalability and reliability provided by cloud-based RDBMS platforms like AWS RDS.

On the other hand, compared to big data solutions like Hadoop and Kafka, the cloud-based RDBMS approach offers a more structured and familiar environment for data management and analysis. While Hadoop excels in distributed data processing and storage, its learning curve and infrastructure requirements may pose challenges for organizations not well-versed in big data technologies. Similarly, Kafka's real-time event processing capabilities are powerful, but its integration with traditional RDBMS systems may require additional complexity and overhead [35].

In conclusion, the deployment of AWS RDS with PostgreSQL presents a compelling solution for the housing manager's international expansion efforts. By leveraging cloud-based infrastructure and a proven open-source RDBMS, the solution offers scalability, reliability, and real-time capabilities tailored to the challenges of handling megabyte-scale datasets. With the ability to manage data efficiently, ensure reliability, and react to real-time events, the housing manager can navigate the complexities of the global rental market effectively and drive business growth.

Task 3: Considering web-based application:

Three Privacy Issues in the Development of a Public-Facing Application:

Data Protection and GDPR Compliance: The housing manager faces a critical challenge in ensuring the lawful collection, storage, and processing of personal data in accordance with data protection laws, notably the General Data Protection Regulation (GDPR) [37]. Evidence from real-world situations highlights the consequences of non-compliance, including substantial fines and reputational damage. For instance, British Airways and Marriott International faced hefty fines—£20 million and £18.4 million, respectively—for GDPR breaches related to inadequate security measures leading to customer data exposure [36][37].

The housing manager must navigate principles such as data minimization, purpose limitation, and robust data security to mitigate these risks. Implementing data anonymization techniques and conducting regular compliance audits can enhance data protection practices. A comparison with other sectors, such as healthcare and finance, underscores the commonality of data protection challenges and the need for stringent compliance measures [38][39].

Data Security and Breach Risks: Storing clients' personal details in a digital environment exposes the housing manager to heightened risks of data breaches and unauthorized access. Recent incidents, such as the Equifax data breach in 2017, exemplify the severe consequences of compromised data security, including identity theft and financial loss for individuals [40]. Equifax, a consumer credit reporting agency, faced a significant backlash and legal repercussions, emphasizing the urgency of implementing stringent security measures.

The housing manager must prioritize encryption, access controls, regular security audits, and staff training to fortify the confidentiality, integrity, and availability of the stored data [41][42]. By comparing security frameworks across industries, such as the Payment Card Industry Data Security Standard (PCI DSS) in finance, the housing manager can adopt best practices and tailor security measures to mitigate breach risks effectively [43].

Consent and Transparency: Securing clear and informed consent from potential clients before collecting personal data is imperative to uphold privacy standards. Recent examples, such as the Facebook-Cambridge Analytica scandal, underscore the repercussions of inadequate transparency and consent practices [44]. Facebook faced severe backlash for mishandling user data without explicit consent, leading to unauthorized profiling and manipulation of user behavior.

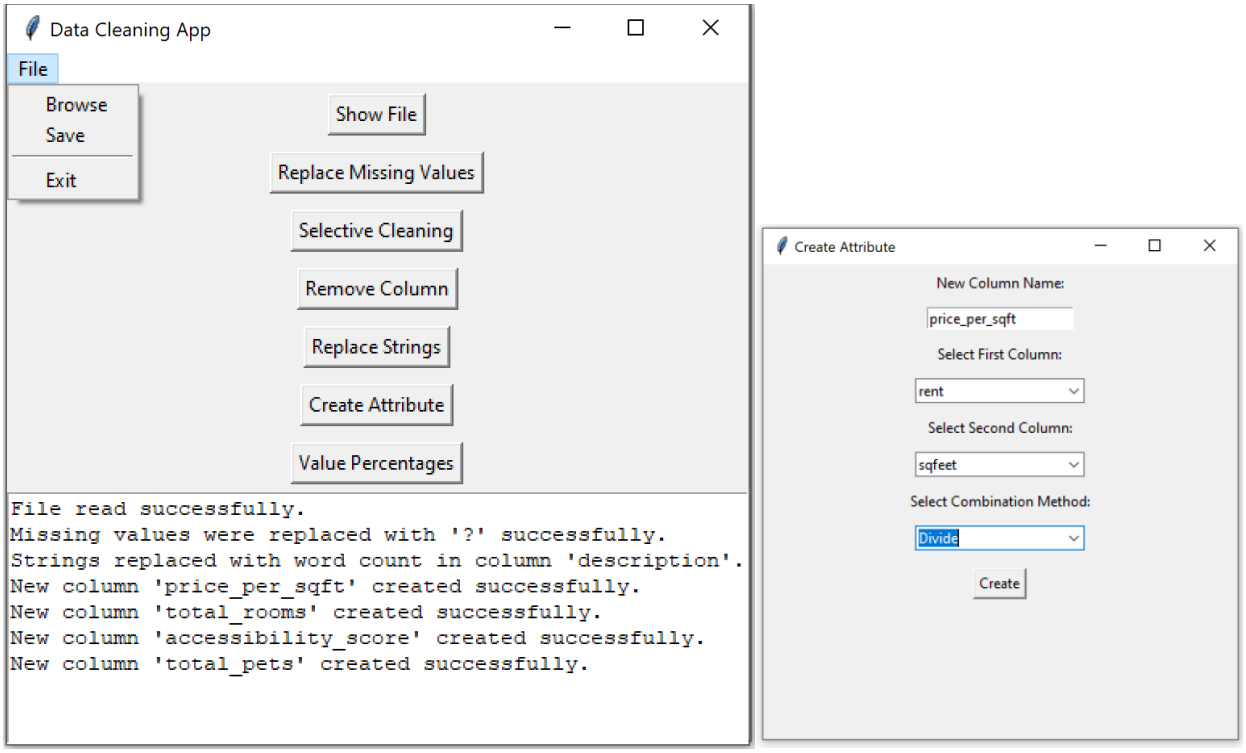
Drawing from these incidents, the housing manager must adopt transparent information practices, detailing data usage, access, and individual rights. Implementing user-friendly mechanisms for consent withdrawal and the exercise of data rights is essential for building trust and complying with privacy expectations [45]. Comparative analyses with industries like e-

commerce and social media elucidate common challenges and innovative solutions for enhancing consent mechanisms and promoting user privacy [46][47].

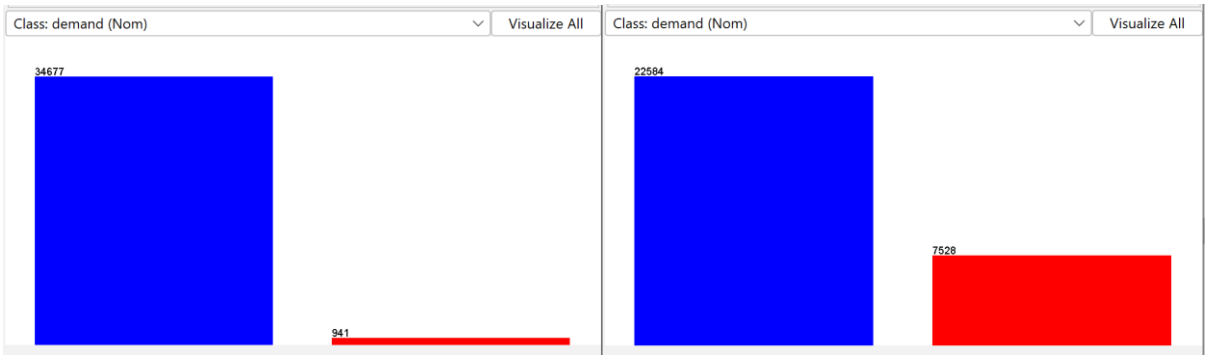
In comparison with similar scenarios and issues, other organizations in the real estate industry have faced similar challenges when developing public-facing applications and capturing personal data. By adopting best practices in data protection, security, and transparency, organizations can mitigate privacy risks and build trust with their clients, ultimately enhancing the success and sustainability of their business operations.

Appendices:

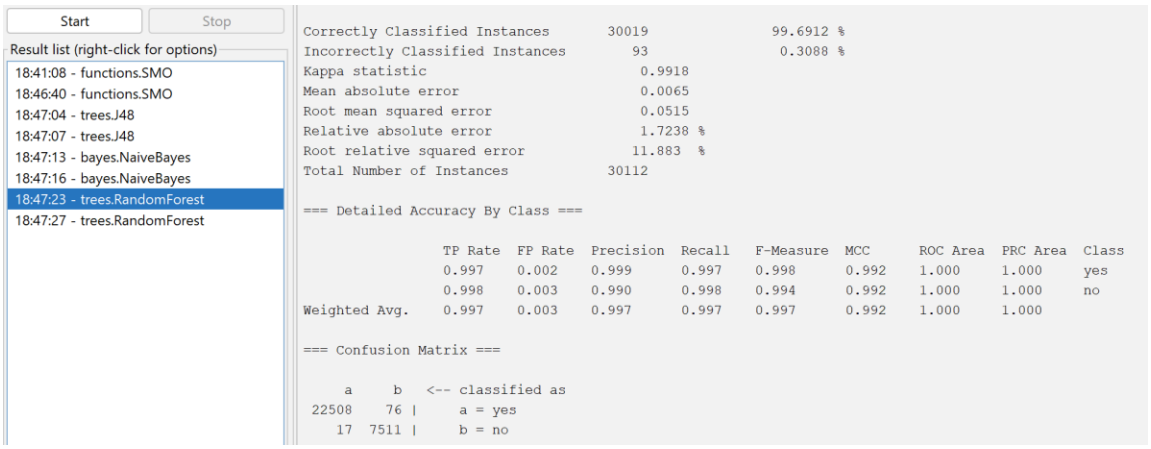
Appendix A: Rental “demand” investigation



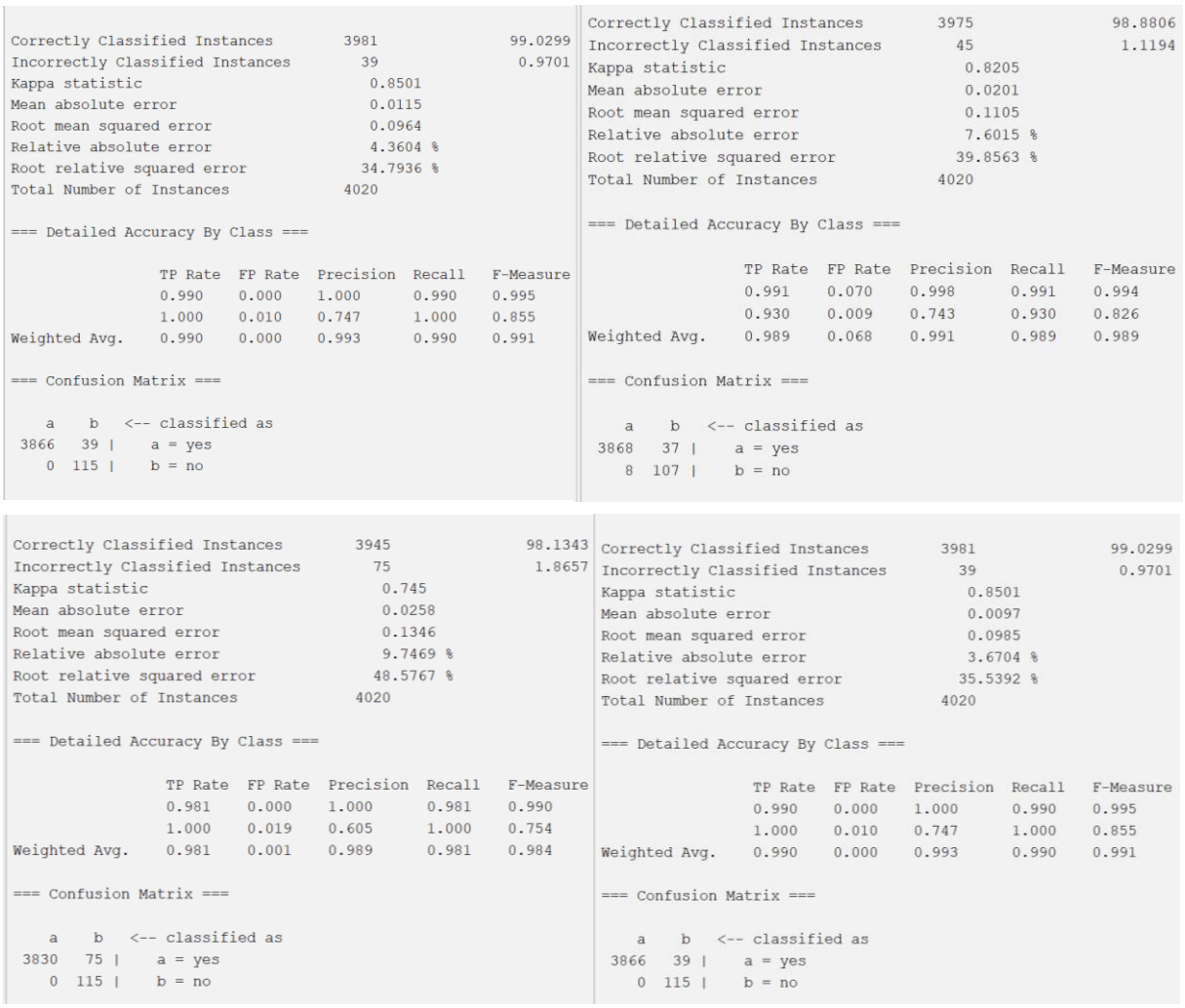
A.1 Data Cleaning App created for initial cleaning and attribute engineering on csv files on right there is the Create Attribute window creating the rent_per_sqft attribute by combining the two attributes “rent” and “sqfeet” using the division method.



A.2 The demand distribution attribute before(left) and after(right) applying Up sampling and Down sampling for data balancing.



A.3 Random Forest classifier results on the training dataset.

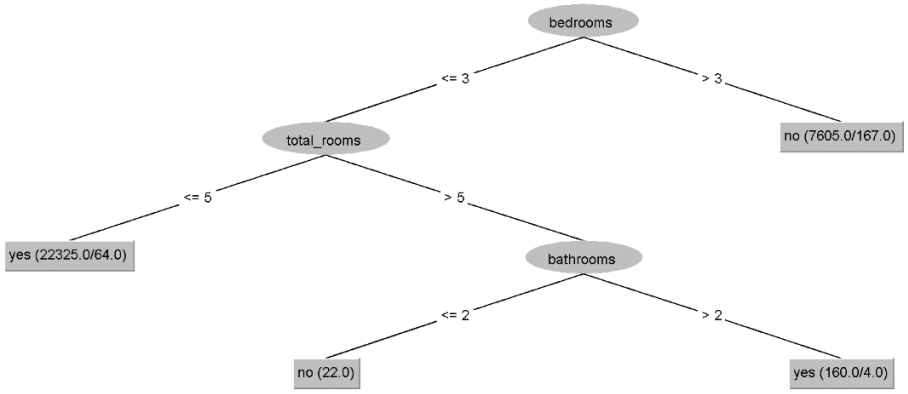


A.4. Comparison of 4 classifier results on the test dataset showing the lower performance of Random forest(top left) and Naïve Bayes(top right) compared to J48(lower left) and support vector machine(lower right).

```
Machine linear: showing attribute weights, not support vectors.

       14.2597 * (normalized) bedrooms
 +     -1.3017 * (normalized) bathrooms
 +      0.0004 * (normalized) cats_allowed=1
 +      0.0002 * (normalized) dogs_allowed=1
 +      0.0001 * (normalized) smoking_allowed=1
 +     -0.0001 * (normalized) wheelchair_access=1
 +     -0.0007 * (normalized) electric_vehicle_charge=1
 +     -0.0001 * (normalized) comes_furnished=1
 +      0      * (normalized) pet_friendly_score
 +      0.0007 * (normalized) accessibility_score
 +      2.388  * (normalized) total_rooms
 +     -0.0007 * (normalized) description
 -      6.782
```

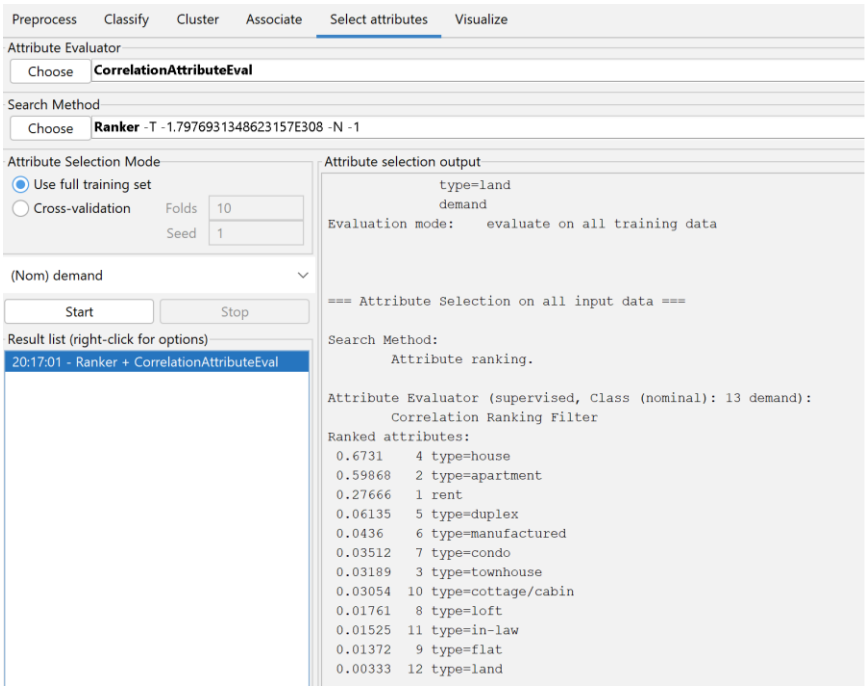
A.5 Attribute weights for discrete variables after training the data on SMO classifier showing bedrooms, bathrooms and total number of rooms having the highest impact on demand prediction.



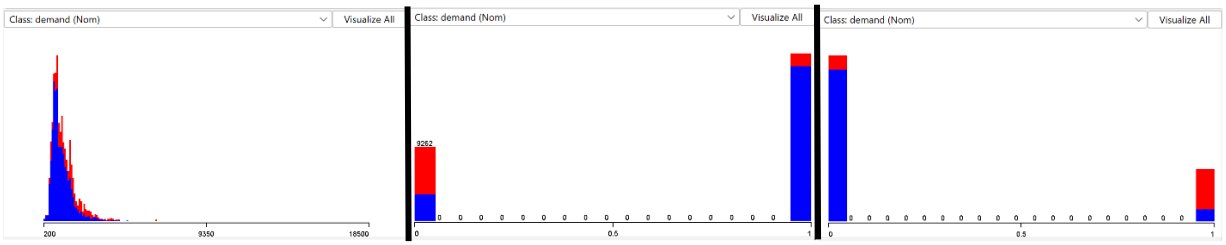
A.6 Visual representation of the J48 tree with the bedroom, bathroom, and total_rooms attributes, highlighting the number of bedrooms as the main predictor.

```
Attribute Evaluator (supervised, Class (nominal): 13 demand):
        Correlation Ranking Filter
Ranked attributes:
 0.8004   1 bedrooms
 0.7516  11 total_rooms
 0.444    2 bathrooms
 0.23     9 pet_friendly_score
 0.2003   3 cats_allowed
 0.1676  12 description
 0.1503   4 dogs_allowed
 0.1009  10 accessibility_score
 0.0998   6 wheelchair_access
 0.0548   7 electric_vehicle_charge
 0.0348   5 smoking_allowed
 0.0197   8 comes_furnished
```

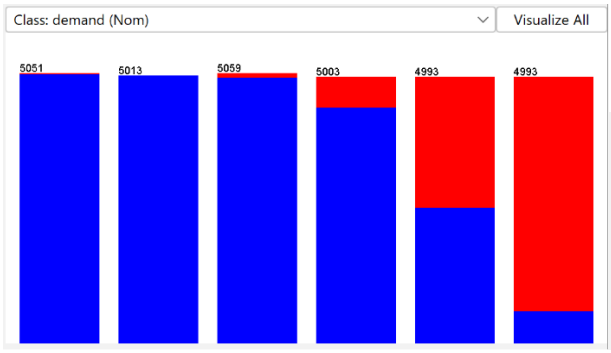
A.7. Correlation evaluation between demand and discrete variables highlighting the significant correlations with number of rooms and weak correlations with pet_friendly_score and number of words in the property description.



A.8 Correlation results with demand as the class attribute showing significant correlation with the types “house” and “apartment” and a moderate correlation with the rent attribute.



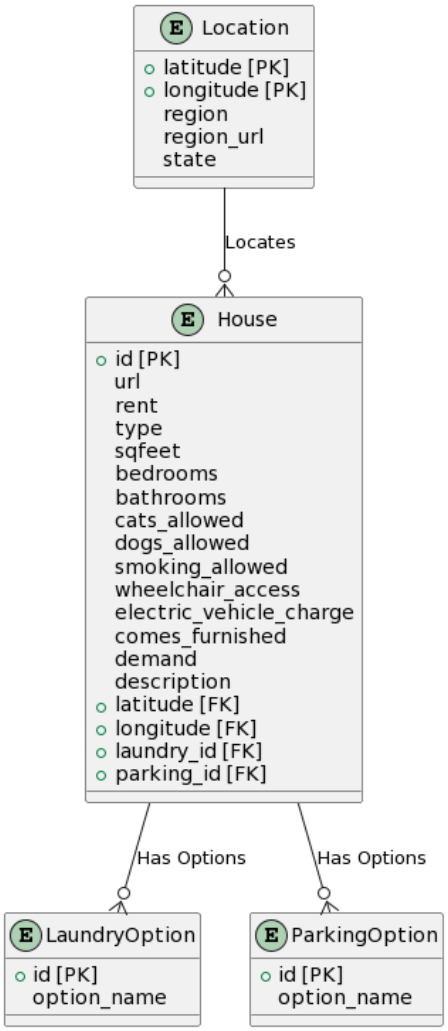
A.9 distribution of demand in the three attributes rent(left), type=apartment(middle) and type=house(right) the blue highlight signifies the value “yes” for demand and the red highlight signifies the value “no”.




```
sqfeet = '(-inf-705]': yes (5051.0/21.0)
sqfeet = '(705-889]': yes (5013.0/4.0)
sqfeet = '(889-1040]': yes (5059.0/80.0)
sqfeet = '(1040-1254]': yes (5003.0/584.0)
sqfeet = '(1254-1709]': yes (4993.0/2448.0)
sqfeet = '(1709-inf)': no (4993.0/602.0)
```

A10. the distribution of demand values of each node in the attribute “sqfeet” converted to nominal format using discretize(left) and J48 tree classified on square feet range nodes(right).

Appendix B: Storing data and possible solutions



B.1 ER Diagram in UML format highlighting the Normalization process through separation of entities and key designation.

```sql
-- Insert data into the Location table
INSERT INTO Location (latitude, longitude,
region, region_url, state)
VALUES
(33.4226, 86.7065, 'birmingham',
'http://example.com/philly', 'al');

-- Insert data into the House entity with
references to Location ,LaundryOption and
ParkingOption
INSERT INTO House (id, url, rent, type, sqfeet,
bedrooms, bathrooms, cats_allowed,
dogs_allowed, smoking_allowed,
wheelchair_access, electric_vehicle_charge,
comes_furnished, demand, description, latitude,
longitude, laundry_option_id,
parking_option_id)
VALUES (274532,'http://example.com/house20',
1200, 'Apartment', 1000, 2, 1, 1, 1, 0, 1, 0,
1, 'yes', 'Cozy apartment in the heart of the
city.', 33.4226, 86.7065, 1, 1);
```

B.2 SQL queries for entering a new line of data

```sql
SELECT description
FROM House
WHERE rent <= 1000
AND cats_allowed = 1
AND dogs_allowed = 1
AND (latitude, longitude) IN (
    SELECT latitude, longitude
    FROM Location
    WHERE state = 'ca');
```

B.3 SQL Query for Extracting the ‘description’ for Properties Meeting Criteria.

```sql
SELECT state, AVG(rent) AS average_rent
FROM House
JOIN Location
ON House.latitude = Location.latitude AND
House.longitude = Location.longitude
GROUP BY state;
```

B.4 SQL Queries extracting the average rental value for each state.

References:

[1] Python Software Foundation. (n.d.). Tkinter, Python's standard GUI (Graphical User Interface) library. [Online]. Available: https://docs.python.org/3/library/tkinter.html [Accessed: Feb. 02, 2024].

[2] McKinney, W., et al. (2010). Panda's library, providing data structures and data analysis tools for Python programming. [Online]. Available: https://pandas.pydata.org/ [Accessed: Feb. 02, 2024].

[3] D. Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, 1993.

[4] J. Han, M. Kamber, J. Pei, "Data Mining: Concepts and Techniques," Elsevier, 2011.

[5] R. O. Duda, P. E. Hart, D. G. Stork, "Pattern Classification," Wiley, 2012.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp

[7] Santos, G. (2020). Upsampling with SMOTE for Classification Projects. Towards Data Science. [Online]. Available: https://towardsdatascience.com/upsampling-with-smote-for-classification-projects-e91d7c44e4bf [Accessed: Feb. 05, 2024].

[8] Batista, G. E. A. P., et al. "A study of the behavior of several methods for balancing machine learning training data." ACM SIGKDD Explorations Newsletter 6.1 (2004): 20-29.

[9] Han, J., et al. "Data Mining: Concepts and Techniques." Elsevier, 2011. [4] Kubat, M., & Matwin, S. "Addressing the curse of imbalanced training sets: one-sided selection." ICML (1997): 179-186.

[10] J. C. Platt, "Fast Training of Support Vector Machines Using Sequential Minimal Optimization," in Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999, pp. 185-208.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5-32, 2001.

[12] P. Domingos and M. Pazzani, "On the optimality of the simple Bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2-3, pp. 103-130, 1997.

[13] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R news*, vol. 2, no. 3, pp. 18-22, 2002.

[14] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121-167, 1998.

[15] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, "Data Mining: Practical Machine Learning Tools and Techniques," *Morgan Kaufmann Publishers Inc.*, 2016.

[16] G. James, D. Witten, T. Hastie, and R. Tibshirani, "An Introduction to Statistical Learning: with Applications in R," *Springer*, 2013.

[17] A. Webb, "Statistical Pattern Recognition," *John Wiley & Sons*, 2002.

[18] Weka. (n.d.). "Discretize." [Online]. Available: https://weka.sourceforge.io/doc.dev/weka/filters/unsupervised/attribute/Discretize.html [Accessed: Feb. 08, 2024].

[19] Rajalakshmi, A., Vinodhini, R., & Fathima Bibi, K. (Year). "Data Discretization Technique Using WEKA Tool." Rajah Serfoji Govt. College (Autonomous), Thanjavur-5.

[20] C. J. Date, "An Introduction to Database Systems," Addison-Wesley, 2003.

[21] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems," Pearson, 2016.

[22] W. Kent, "A Simple Guide to Five Normal Forms in Relational Database Theory," Communications of the ACM, vol. 26, no. 2, pp. 120-125, 1983.

[23] W. Lemihau, "Principles of Database Management," XYZ Publishers, 2018.

[24] Celko, J. (2007). "Choosing the Right Primary Key for Your Database Tables." Database Journal. [Online]. Available: https://www.databasejournal.com/features/mssql/article.php/3626056/Choosing-the-Right-Primary-Key-for-Your-Database-Tables.htm [Accessed: Feb. 12, 2024].

[25] TechTarget. (2024). "Cloud Database Comparison: AWS, Microsoft, Google, and Oracle." SearchDataManagement. [Online]. Available: https://www.techtarget.com/searchdatamanagement/tip/Cloud-database-comparison-AWS-Microsoft-Google-and-Oracle [Accessed: Feb. 12, 2024].

[26] Amazon Web Services. (n.d.). Amazon Relational Database Service (RDS). Retrieved from https://aws.amazon.com/rds/

[27] PostgreSQL. (n.d.). PostgreSQL: The world's most advanced open-source relational database. Retrieved from https://www.postgresql.org/

[28] Amazon Web Services. (n.d.). Amazon Kinesis Data Streams. Retrieved from https://aws.amazon.com/kinesis/data-streams/

[29] Amazon Web Services. (n.d.). AWS Lambda - Serverless Compute. Retrieved from https://aws.amazon.com/lambda/

[30] Stonebraker, M., & Rowe, L. (2019). Readings in Database Systems (5th ed.). The MIT Press.

[31] Date, C. J. (2003). An Introduction to Database Systems (8th ed.). Pearson Education.

[32] Kafka, A., Garg, N., & Mittal, N. (2013). Learning Apache Kafka (1st ed.). Packt Publishing.

[33] Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009). Database Systems: The Complete Book (2nd ed.). Prentice Hall.

[34] Sharma, R. (2018). PostgreSQL High Performance Cookbook: Over 100 recipes to design, implement, and manage successful PostgreSQL database solutions (3rd ed.). Packt Publishing.

[35] European Union (2016). "General Data Protection Regulation (GDPR)." Official Journal of the European Union, 59(1), 1-88. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679 [Accessed: Feb. 16, 2024].

[36] Information Commissioner's Office (2019). British Airways fined more than £183m over data breach. [Online]. Available: https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2019/07/british-airways-fine-reduced-to-20m/ [Accessed: Feb. 16, 2024].

[37] Information Commissioner's Office (2019). Statement: Marriott International, Inc. [Online]. Available: https://ico.org.uk/about-the-ico/news-and-events/news-and-blogs/2019/07/statement-marriott-international-inc/ [Accessed: Feb. 17, 2024].

[38] HealthITSecurity (2021). HIPAA Compliance Checklist. [Online]. Available: https://healthitsecurity.com/features/hipaa-compliance-checklist [Accessed: Feb. 18, 2024].

[39] Information Commissioner's Office, "Data protection impact assessments," ICO, [Online]. Available: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/accountability-and-governance/data-protection-impact-assessments/ [Accessed: Feb. 19, 2024].

[40] The United States House of Representatives (2018). Data Breach: Equifax. [Online]. Available: https://oversight.house.gov/sites/democrats.oversight.house.gov/files/Equifax%20Report.pdf [Accessed: Feb. 22, 2024].

[41] National Institute of Standards and Technology, "Framework for Improving Critical Infrastructure Cybersecurity," NIST, April 2018. [Online]. Available: https://www.nist.gov/publications/framework-improving-critical-infrastructure-cybersecurity [Accessed: Feb. 24, 2024].

[42] Financial Services Information Sharing and Analysis Center (2021). Cybersecurity Framework for Financial Services. [Online]. Available: https://www.fsisac.com/standards/framework [Accessed: Feb. 25, 2024].

[43] Payment Card Industry Security Standards Council (2021). PCI Security Standards. [Online]. Available: https://www.pcisecuritystandards.org/ [Accessed: Feb. 27, 2024].

[44] Cadwalladr, C., & Graham-Harrison, E. (2018). Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. The Guardian. [Online]. Available: https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election [Accessed: Feb. 29, 2024].

[45] Facebook Newsroom (2018). An Update on Our Plans to Restrict Data Access on Facebook. [Online]. Available: https://about.fb.com/news/2018/04/restricting-data-access/ [Accessed: Feb. 22, 2024].

[46] Winshuttle (2021). GDPR Compliance in E-commerce: Best Practices for Data Privacy. [Online]. Available: https://www.winshuttle.com/learn/gdpr-compliance-in-e-commerce-best-practices-for-data-privacy/ [Accessed: Feb. 22, 2024].

[47] Krasodomski-Jones, A., & Parker, G. (2020). The Online Privacy Paradox. Demos. [Online]. Available: https://demos.co.uk/wp-content/uploads/2020/06/The-Online-Privacy-Paradox-Demos.pdf [Accessed: Feb. 22, 2024].