Rental Housing Data Analysis

Big Data Analytics: Summative Assessment

Department of Computer Science

University of York

March 2024

Word Count: 1513, Task 1

997, Task 2

470, Task 3

Total (2980)

Contents

Task 1

The primary goal of this investigation was to analyse a dataset comprising rental property listings to discern which characteristics influence customer demand utilising the data analysis software 'Weka'.

Initially, the dataset was reviewed in Excel which aimed to distinguish between discrete and continuous variables and identify outliers in critical variables like rent and sqfeet. Scatter plots and histograms aided in understanding the data distribution and the impact of outliers, subsequently addressed using Python for data cleaning [1].

Feature selection was performed early in the process, with less relevant columns like 'ID', 'region URL', 'URL', and, 'description' being removed, as they offered little value when evaluating demand although description could be valuable, a text processor would be needed for the unstructured data. Additionally, data inconsistencies such as 'n0' and 'n' instead of 'no' and 'flat' instead of 'apartment' were corrected and invalid values (e.g., URLs) in demand were removed to standardise the dataset. Furthermore, Python was used for preprocessing tasks, including outlier removal through IQR method and dataset downsampling [2] [3] [4]. Further cleaning ensured data consistency, such as standardising property type names and addressing missing data. There were approximately 30% missing values for parking and 16% for laundry, which can be a significant issue with no robust methods for dealing with it appropriately [5], thus analysis was carried out to observe how the model performance was affected when they were imputed, accepted or removed with Weka's imputation creating the best results. For the test set, missing values were replaced with NaN for nominal attributes and '?' for numeric attributes. Command line Weka rather than GUI Weka was utilised for the data analysis due to the GUI being too restrictive for large data sets.

Additionally, the dataset underwent normalisation and discretisation, dividing continuous variables into bins to improve predictive accuracy and improve machine learning performance [6] [7] [8]. Discretisation and nominalisation were both considered due to the advantages of flexible categorisation, and parameters tested to see which produced more accurate and computer efficient predictive models [9].

(A) Discrete Variables and Housing Demand

Assumptions:

It was assumed that discrete variables such as type and bedrooms would significantly impact customer preferences hence housing demand.

Analysis Techniques:

The study employed classifiers SMO, RandomForest, and OneR and used the four test options: training set data, test set data, 10-fold cross validation, 66% train-test split. The discrete variables' predictive potential was tested by CfsSubsetEval and BestFirst which highlighted "type", "bedrooms", and "bathrooms" as key predictors. Additionally, InfoGainAttributeEval and CorrelationAttributeEval rank "bedrooms" and "bathrooms" highest, indicating these have the most information gain or correlation with demand. Auto-Weka was utilised and predicted OneR as the best classifier for the housing dataset.

Justification of Techniques:

RandomForest was used as the balanced random forest approach for WEKA improves prediction quality for minority classes [10]. SMO was used due to having fast optimisations and allowing for working with large training sets [11]. OneR was chosen as it is promising in sentiment analysis [12]. Moreover, although not always correct, Auto-Weka automatically selects the best learning algorithm and hyperparameter settings for a dataset although hence its use in my analysis [13].

Summary of Results:

The analysis of discrete variables regarding demand reveals insights into factors influencing rental preferences. "Bedrooms," "Bathrooms," and "Parking_options" emerged as significant predictors of demand.

Bedrooms significantly predict demand, with RandomForest showing True Positive (TP) rates of 0.999 for high demand and 0.994 for low demand. False Positive (FP) rates were minimal at 0.006 and 0.001, respectively.

Bathrooms were effectively predicted by the SMO classifier for high demand (TP rate: 0.98, precision: 0.68), while RandomForest was better at identifying low demand scenarios.

Type of property saw strong performance across classifiers on the test set, with a high TP rate of 0.857 but with a notable FP rate of 0.438, indicating a mix of accurate high demand predictions and significant misclassification of low demand properties.

Smoking_allowed showed good classifier performance (TP rate: 0.857) but with a high FP rate of 0.802, suggesting many high demand predictions were incorrect.

Laundry_options had a moderate predictive capability for high demand, especially with SMO (precision: 0.98) but suffered from a substantial FP rate (0.44). OneR provided more balanced results with a lower FP rate.

Parking_allowed showed strong predictive ability for both high and low demand with OneR, indicating a significant role in influencing demand, despite high precision (0.99) for high demand predictions and lower precision (0.07) for low demand predictions.

State demonstrated overfitting in the test set, leading to unreliable predictions. Consistent TP rates suggest some predictive value, but high error metrics and low Kappa Statistics indicate difficulties in using state as a reliable demand predictor.

(B) Correlation between Demand, Rent, and Type

Assumptions:

The analysis assumed a potential correlation between rent, property type, and housing demand, positing that higher rents and specific property types might influence demand negatively or positively.

Analysis Techniques:

Linear regression models and Correlation Attribute Evaluators were employed to explore the relationships between demand, rent, and property type.

Justification of Techniques:

Data normalization was used with the purpose of normalizing variable values to bring the variables to comparability and standardizing their orders of magnitude [14]. Conversion of the nominal variable type to binary facilitated a more detailed correlation analysis. This step was vital for enabling linear regression analysis and correlation coefficient calculation. The linear regression model was selected for its ability to quantify the strength and direction of relationships between continuous and binary variables, offering insights into how rent and type influence demand.

Summary of Results:

Rent shows a weak positive correlation with low demand (coefficient: 0.158), suggesting slightly higher no-demand likelihood as rent increases, albeit weakly.

Property type impacts demand differently: Apartments show a negative correlation with no demand, implying higher popularity, whereas houses positively correlate with no demand, indicating lower demand. Other types like townhouses and condos have weaker or negligible correlations. Duplex and manufactured were inversely related to "low demand" hence are likely to be in demand. Coefficients for property type house are positive, suggesting it has a direct relationship with "low demand". The correlation analysis indicates a nuanced effect of rent and type on demand.

(C) Optimal Range of SqFeet for High Demand

Assumptions:

Initially, it was presumed that customer preferences might gravitate towards properties with larger sqfeet, hypothesizing that space could be a significant determinant of demand.

Analysis Techniques:

Normalisation and discretisation of sqfeet, categorising it into bins. The OneR classifier, applied to training data, 10-fold cross-validation, and a 66% split, effectively identifying demand trends across sqfeet ranges. Visualisation in Weka aided in distinguishing high from low demand zones, leading to the identification of the sqfeet range most conducive to high demand. Weka's Hot Spot analysis highlighted the sqfeet ranged that correlated best with 'demand=yes'.

Justification of Techniques:

Discretisation was selected for its efficacy in simplifying continuous data, transforming sqfeet into manageable categories for analysis. The choice of ten bins was strategic with each bin representing a span of approximately 150sqfeet—neither too narrow to miss subtleties nor too broad to dilute significant trends. Weka's HotSpot was used to identify and predict hotspots, determining which size properties are associated with high demand [15].

Summary of Results:

The analysis revealed a clear preference for properties within certain sqfeet ranges. The most optimal range for demand was identified as bin 5 (796.2 - 951.5sqft), followed by bin 6 and then bin 4. This underscores the market's preference for properties sized between 640-1106.8sqft. This finding mostly aligns with a hot spot analysis, pinpointing the demand for sqfeet to be in the range greater than 0.24 but less than or equal to 0.444, highlighting specific property sizes most conducive to high demand.

Critical Evaluation

The initial phase of this study faced challenges due to dataset imbalance, leading to a critical evaluation of our preprocessing strategies. Initially, Weka's ClassBalancer and SMOTE for class balance were tested but proved inadequate for significantly imbalanced data, where the minority class made up 2% of the dataset, highlighting the complexities of working with weighted data and severe class imbalances [16]. This limitation prompted a pivot to downsampling the majority class however, employing SMOTE before final feature selection inadvertently may have introduced a bias towards predicting 'low' demand [17]. Furthermore, multiple imputation may have created better results than the single imputation carried out could have caused false precision [18]. After reflecting on the need for classifiers rather than a regression model in task 1, realised that a classifier like NaiveBayes would have been better for the analysis [19], especially for the categorical variables, however SMO still performed very well in most analyses.

Addressing the Housing Manager's Question:

The analysis revealed that features like smoking allowance, laundry facilities, and particularly the type of property (e.g., apartments) positively influence tenant interest and drive demand. Key variables "Bedrooms" and "Bathrooms," emerged as dual indicators predicting both high and low demand, despite some inconsistencies in bathroom-related data. The study also pinpointed a preferred property size range of 640-1106.8sqft as optimal for high demand. Focusing marketing and investment on properties within this size range could strategically boost demand, aligning with consumer preferences.

Task 2

Part 1

Whilst designing the relational database for a global rental business, strategic decisions were made to optimise data structure and operational efficiency. At the foundation of this design lies the 'Property' entity, chosen as the core due to its centrality in representing the essence of the business as a housing company. The attributes like URL, 'Description', 'Rent', and various amenities were directly integrated into this entity to facilitate quick, efficient data access, crucial for operational agility for when the database would be created. Furthermore, the database evolution using normalisation saw ‘Location', 'Type', 'Laundry', and 'Parking' into distinct entities was driven by a desire for modular data management, allowing for detailed, flexible categorisation that enhances the database's scalability and usability. Each entity, defined by its unique primary key, supports structured relationships across the database, enabling coherent organisation and easy navigation of related data.

Furthermore, the decision to incorporate 'Region' details within the 'Location' entity, rather than as a separate entity with its own ID and URL, was driven by the desire to streamline queries and minimise complexity. This approach ensures seamless access to regional information alongside location data, preventing unnecessary joins that could impede performance. Likewise, retaining amenities like 'comes_furnished', 'cats_allowed', and 'smoking_allowed' within the 'Property' entity simplifies data retrieval, enabling straightforward queries that align with the business and client needs.

Following the creation of the ER diagram, the focus shifted to building the database within the SQL domain. Writing SQL statements that reflect the database's logical structure and entity interrelations was essential. For example, the SQL command for inserting a new property record illustrates the management of attributes across different entities (*), showcasing the complexity of the core entity with its multiple variables and values. Moreover, SQL queries crafted to extract specific information - such as property descriptions meeting defined criteria or calculating average rental values by state - highlight the use of SQL for data navigation and extraction based on the established relational model (*, *).

In conclusion, the database design is characterised by careful consideration of both structural integrity and operational practicality. 'Property' serves as the database's core entity, with directly integrated attributes for enhanced access efficiency. The creation of separate entities for 'Location', 'Type', 'Laundry', and 'Parking', along with the strategic inclusion of 'Region' within 'Location', exemplifies a balanced approach to data organisation.

Part 2

With the probability of a growing database, scalability is a significant consideration for a global rental business and the adoption of a sophisticated and flexible data infrastructure emerges as a requirement. This infrastructure must adeptly manage expansive volumes of data while enabling swift, real-time operations. Central to this strategy are key entities including ‘Property’, ‘Type’, ‘Location’, ‘Laundry’, and ‘Parking’, coupled with the imperative need for a rapid-response system. A recommended approach for the rental company in its long-term aim of globalisation encapsulates the utilisation of the Hadoop Ecosystem and cloud-based services for scalable storage solutions, as they are generating considerably more data, Apache HBase for immediate data access, and the employment of MapReduce and Apache Spark for effective parallel processing. Additionally, Spark Streaming or Apache Flink is proposed for the real-time processing of data streams.

The Hadoop Distributed File System (HDFS) has the potential to store extensive property data across numerous computers, inherently equipped to manage large datasets by distributing them across a cluster, thus enhancing fault tolerance and ensuring high availability through the replication of data block with massive scalable IO capability. This distributed storage solution is further augmented by cloud-based Big Data Services such as AWS, Google Cloud Platform, and Azure. These services offer managed Hadoop clusters, providing scalability and presenting a cost-effective solution by mitigating the need for direct management of physical infrastructure. However, despite Hadoop excelling at processing vast datasets, its batch processing nature might not align with all real-time analysis requirements, with alternatives like Apache Storm or Apache Samza offering more real-time data processing solutions and typically provide scalable schema-flexible solutions for diverse data types [20]. However, although NewSQL is cost-effective, extensive data processing and storage can incur significant expenses. On-premises solutions or hybrid cloud approaches might provide more control over costs for some businesses. Furthermore, Apache HBase presents an optimal solution for scenarios that demand quick data access and modifications. Operating atop HDFS, HBase enables real-time read/write access, proving particularly advantageous for managing large, sparse datasets. In tandem, the incorporation of Spark Streaming or Apache Flink with Apache Kafka for real-time data streaming and processing efficiently pinpoints significant trends or triggers, thereby enabling the prompt dispatch of alerts or messages. Whilst Hadoop’s MapReduce supports distributed data processing tasks, Apache Spark bolsters this capability with an advanced computational engine capable of in-memory processing, dramatically accelerating analytics tasks. This feature plays a crucial role in underpinning the rapid-response system by facilitating swift insights and actions derived from data analysis.

The adoption of these techniques permits horizontal expansion of the computing cluster in line with increasing data volumes - an essential consideration for a business eyeing global expansion. Systems such as HDFS, HBase, and Kafka are engineered to ensure data safety and operational continuity, which is vital for maintaining service integrity in the face of component failures. Moreover, the proposition for real-time data processing meets the critical need for a rapid-response system, with Kafka, Spark Streaming, or Flink ensuring the business can promptly act on data-derived insights. The utilisation of cloud services for big data infrastructure management translates to notable savings mitigates substantial initial investments in hardware and maintenance. Lastly, the global accessibility facilitated by cloud platforms ensures that international offices can seamlessly store, access, and analyse data without geographical constraints.

In conclusion, the strategic implementation of these technologies equips the rental business with the necessary tools to efficiently manage and analyse big data on a global scale. This not only supports informed strategic decision-making but also enhances responsiveness in a competitive rental market. This strategic application of technology addresses current operational needs and positions the business well to navigate future challenges and opportunities, ensuring sustained growth and competitiveness.

Task 3

In the context of a rental business globalising with an initiative to capture personal details through online forms, providing personalised recommendations to revisiting clients, presents several privacy challenges. This report delineates three privacy issues and outlines mitigation strategies.

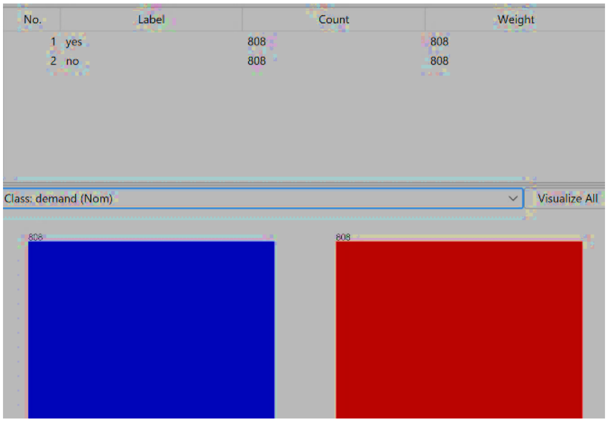
Firstly, with the proliferation of online data collection, personal data breaches pose a significant risk, with a maximum breach size of 200 million expected to grow by 50% over the next five years [21], thus a fundamental concern is ensuring the security of personal data for landlords and tenants. Prioritisation of personal data security is paramount as the leakage of sensitive data due to inadequate security measures could lead to discrimination, fraud, identity theft, and financial loss [22] [23]. To mitigate these risks, there must be improved data security postures [24] with inclusion of advanced encryption protocols like homomorphic encryption, such as which prevents data spoofing [25] by converting data to ciphertext [26], which in turn provides secure and fast data transmissions over networks [27], safeguarding information from potential breaches.

Secondly, the rental company is planning to engage in the longitudinal collection and storage of clients' personal data. This process inherently involves tracking individual information over extended periods, providing critical insights for personalised recommendations but also introduces greater vulnerability to privacy risks. Moreover, the potential misuse of data for purposes not initially consented to by clients, like unsolicited marketing or profiling, is a significant privacy concern. Techniques from the healthcare sector anonymisation, like the (K,C)P-privacy model [28], which prevents identity and attribute disclosure in multidimensional longitudinal data, can be adapted for rental data. This model, coupled with a hybrid anonymisation algorithm, has shown effectiveness in maintaining privacy while allowing for statistical analysis [28].

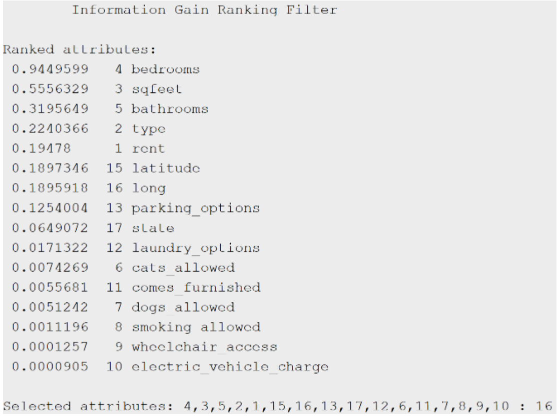
The initiative to leverage data analysis for offering personalised recommendations to clients inherently involves collecting, analysing, and storing personal data. This practice significantly elevates the risk of re-identification, where anonymised data can potentially be cross-referenced with other datasets, revealing individual identities. This concern transcends theoretical speculation, as evidenced by numerous instances where anonymised data was subsequently re-identified, compromising privacy [29]. To mitigate this privacy risk, implementing differential privacy during the data analysis phase is feasible. Differential privacy infuses analysis outputs with noise, rendering it mathematically infeasible to backtrack any individual's data [30]. This technique, applied to analysing customer behaviour for personalised recommendations, ensures the protection of individual identities while exploiting data for business insights. Another strategy is synthetic data generation which statistically mirrors the real dataset but devoid of any actual personal information, synthetic data eradicates re-identification risks [30]. These strategies collectively fortify the privacy framework, safeguarding against re-identification threats while maintaining data utility.

In conclusion, the development of a public-facing application for the rental business necessitates a thorough consideration of privacy concerns. These mitigation strategies lay the foundation for a privacy-respecting web-based application that supports the business's expansion.

Appendix



No.	Label	Count	Weight
1	yes	808	808
2	no	808	808



```
        Information Gain Ranking Filter

Ranked attributes:
 0.9449599    4 bedrooms
 0.5556329    3 sqfeet
 0.3195649    5 bathrooms
 0.2240366    2 type
 0.19478      1 rent
 0.1897346   15 latitude
 0.1895918   16 long
 0.1254004   13 parking_options
 0.0649072   17 state
 0.0171322   12 laundry_options
 0.0074269    6 cats_allowed
 0.0055681   11 comes_furnished
 0.0051242    7 dogs_allowed
 0.0011196    8 smoking_allowed
 0.0001257    9 wheelchair_access
 0.0000905   10 electric_vehicle_charge

Selected attributes: 4,3,5,2,1,15,16,13,17,12,6,11,7,8,9,10 : 16
```

Smoking Variable

Classifier	Evaluation	Demand	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC
SMO	Training Set	High	0.755	0.720	0.512	0.755	0.610	0.039	0.517
		Low	0.280	0.245	0.533	0.280	0.367	0.039	0.517
	Test Set	High	0.857	0.802	0.973	0.857	0.911	0.026	0.528
		Low	0.198	0.143	0.039	0.198	0.066	0.026	0.528

Type Variable

Classifier	Demand	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
OneR	High	0.913	0.438	0.676	0.913	0.777	0.508
	Low	0.562	0.087	0.866	0.562	0.682	0.508

Bedroom Variable

Classifier	Demand	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
RandomForest	High	0.999	0.006	0.994	0.999	0.996	0.993
	Low	0.994	0.001	0.999	0.994	0.996	0.993

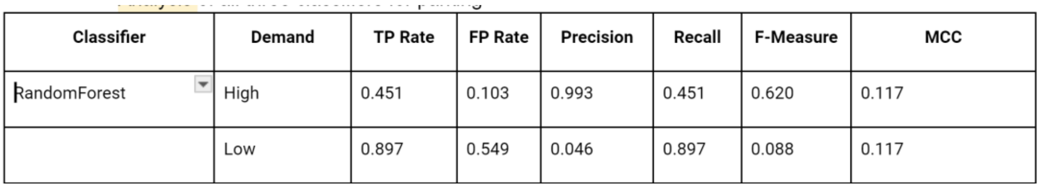
Bathroom variable

Classifier	Demand	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
RandomForest	High	0.548	0.022	0.961	0.548	0.698	0.582
	Low	0.978	0.452	0.684	0.978	0.805	0.582

Laundry Variable

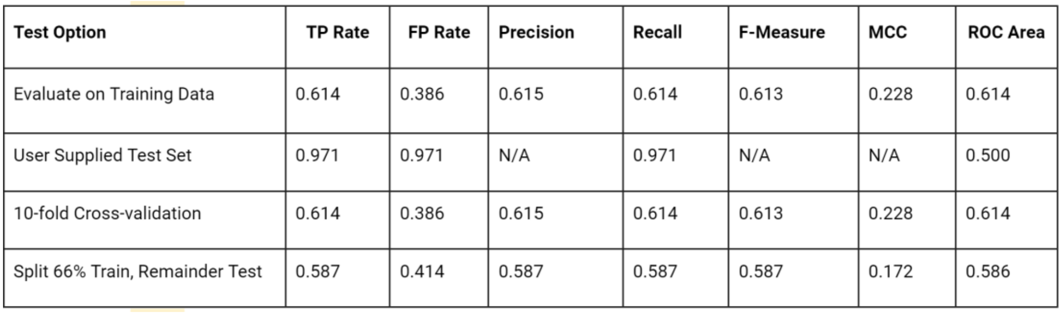
Classifier	Demand	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
OneR	High	0.544	0.129	0.993	0.544	0.703	0.139
	Low	0.871	0.456	0.053	0.871	0.101	0.139

Parking Variable

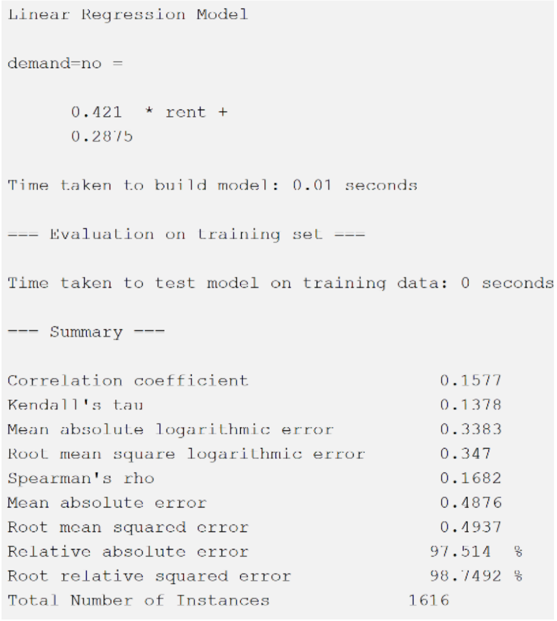


Classifier	Demand	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC
RandomForest	High	0.451	0.103	0.993	0.451	0.620	0.117
	Low	0.897	0.549	0.046	0.897	0.088	0.117

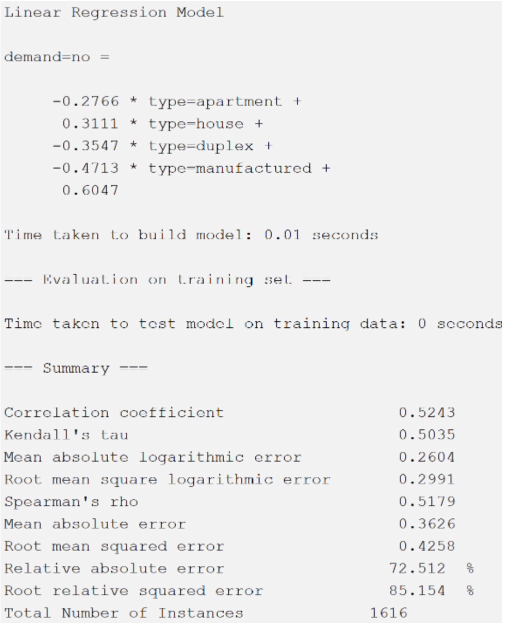
State variable evaluated by OneR classifier with clear overfitting in test set



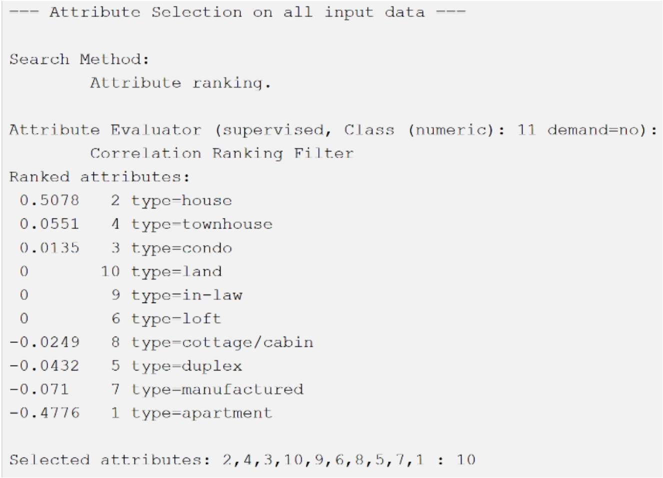
Test Option	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area
Evaluate on Training Data	0.614	0.386	0.615	0.614	0.613	0.228	0.614
User Supplied Test Set	0.971	0.971	N/A	0.971	N/A	N/A	0.500
10-fold Cross-validation	0.614	0.386	0.615	0.614	0.613	0.228	0.614
Split 66% Train, Remainder Test	0.587	0.414	0.587	0.587	0.587	0.172	0.586



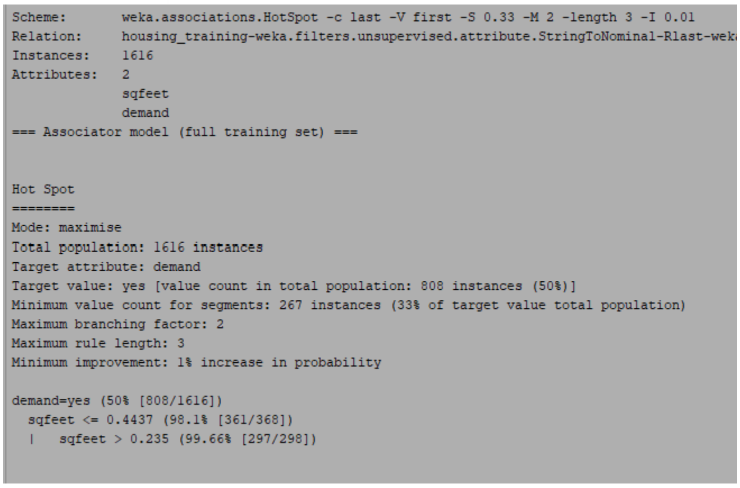
```
Linear Regression Model

demand=no =

      0.421  * rent +
      0.2875

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.1577
Kendall's tau                            0.1378
Mean absolute logarithmic error          0.3383
Root mean square logarithmic error       0.347
Spearman's rho                           0.1682
Mean absolute error                      0.4876
Root mean squared error                  0.4937
Relative absolute error                 97.514  %
Root relative squared error             98.7492 %
Total Number of Instances             1616
```



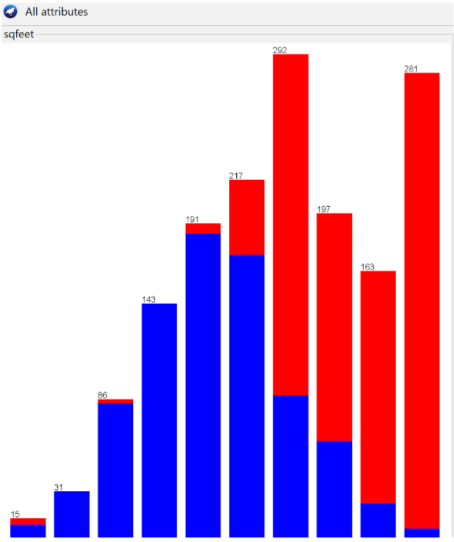
```
Linear Regression Model

demand=no =

     -0.2766 * type=apartment +
      0.3111 * type=house +
     -0.3547 * type=duplex +
     -0.4713 * type=manufactured +
      0.6047

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correlation coefficient                  0.5243
Kendall's tau                            0.5035
Mean absolute logarithmic error          0.2604
Root mean square logarithmic error       0.2991
Spearman's rho                           0.5179
Mean absolute error                      0.3626
Root mean squared error                  0.4258
Relative absolute error                 72.512  %
Root relative squared error             85.154  %
Total Number of Instances             1616
```

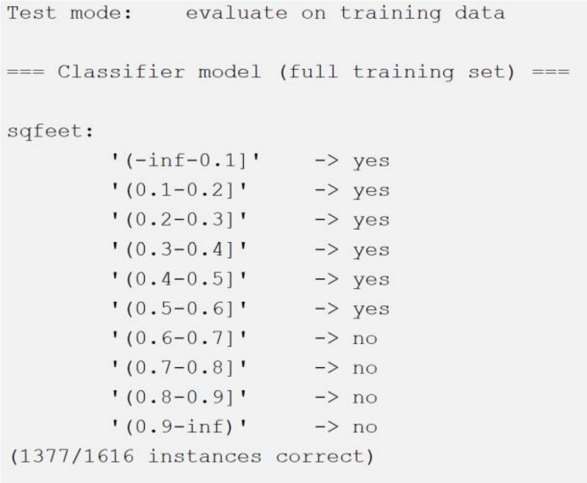


```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (numeric): 11 demand=no):
        Correlation Ranking Filter
Ranked attributes:
 0.5078    2 type=house
 0.0551    4 type=townhouse
 0.0135    3 type=condo
 0        10 type=land
 0         9 type=in-law
 0         6 type=loft
-0.0249    8 type=cottage/cabin
-0.0432    5 type=duplex
-0.071     7 type=manufactured
-0.4776    1 type=apartment

Selected attributes: 2,4,3,10,9,6,8,5,7,1 : 10
```



```
Scheme:       weka.associations.HotSpot -c last -V first -S 0.33 -M 2 -length 3 -I 0.01
Relation:     housing_training-weka.filters.unsupervised.attribute.StringToNominal-Rlast-wek
Instances:    1616
Attributes:   2
              sqfeet
              demand
=== Associator model (full training set) ===


Hot Spot
========
Mode: maximise
Total population: 1616 instances
Target attribute: demand
Target value: yes [value count in total population: 808 instances (50%)]
Minimum value count for segments: 267 instances (33% of target value total population)
Maximum branching factor: 2
Maximum rule length: 3
Minimum improvement: 1% increase in probability

demand=yes (50% [808/1616])
  sqfeet <= 0.4437 (98.1% [361/368])
  |   sqfeet > 0.235 (99.66% [297/298])
```





```
Test mode:    evaluate on training data

=== Classifier model (full training set) ===

sqfeet:
        '(-inf-0.1]'    -> yes
        '(0.1-0.2]'     -> yes
        '(0.2-0.3]'     -> yes
        '(0.3-0.4]'     -> yes
        '(0.4-0.5]'     -> yes
        '(0.5-0.6]'     -> yes
        '(0.6-0.7]'     -> no
        '(0.7-0.8]'     -> no
        '(0.8-0.9]'     -> no
        '(0.9-inf)'     -> no
(1377/1616 instances correct)
```

Bin 1 (inf-0.1): 175.0 - 330.3sqft

Bin 2 (0.1-0.2): 330.3 - 485.6sqft

Bin 3 (0.2-0.3): 485.6 - 640.9sqft

Bin 4 (0.3-0.4): 640.9 - 796.2sqft

Bin 5 (0.4-0.5): 796.2 - 951.5sqft

Bin 6 (0.5-0.6): 951.5 - 1106.8sqft

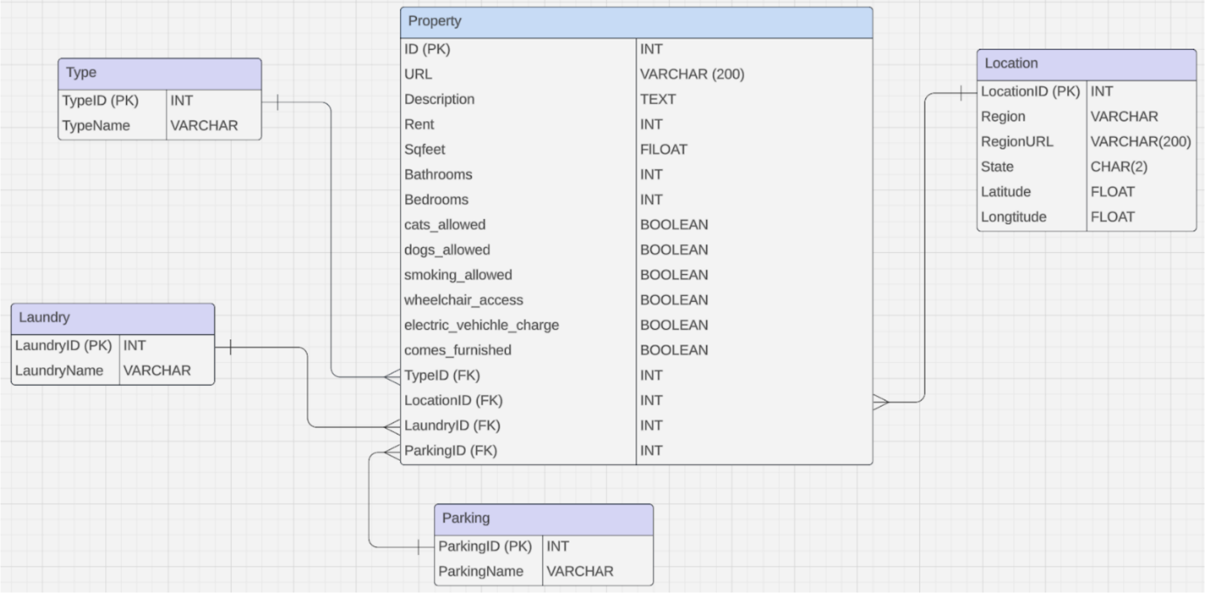
Bin 7 (0.6-0.7): 1106.8 - 1262.1sqft

Bin 8 (0.7-0.8): 1262.1 - 1417.4sqft

Bin 9 (0.8-0.9): 1417.4 - 1572.7sqft

Bin 10 (0.9-inf): 1572.7 - 1728.0sqft

ER diagram



Type

TypeID (PK)	INT
TypeName	VARCHAR

Property

ID (PK)	INT
URL	VARCHAR (200)
Description	TEXT
Rent	INT
Sqfeet	FILOAT
Bathrooms	INT
Bedrooms	INT
cats_allowed	BOOLEAN
dogs_allowed	BOOLEAN
smoking_allowed	BOOLEAN
wheelchair_access	BOOLEAN
electric_vehichle_charge	BOOLEAN
comes_furnished	BOOLEAN
TypeID (FK)	INT
LocationID (FK)	INT
LaundryID (FK)	INT
ParkingID (FK)	INT

Location

LocationID (PK)	INT
Region	VARCHAR
RegionURL	VARCHAR(200)
State	CHAR(2)
Latitude	FLOAT
Longtitude	FLOAT

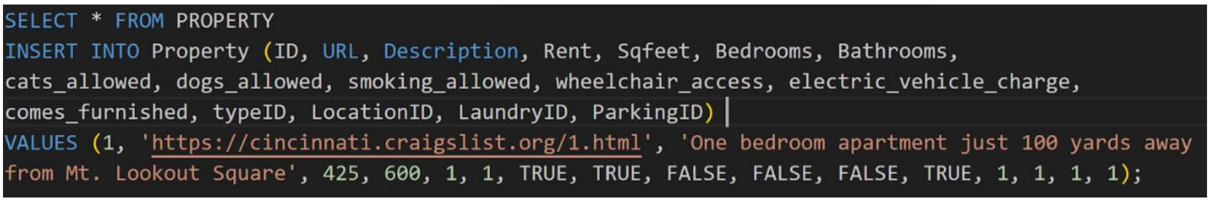
Laundry

LaundryID (PK)	INT
LaundryName	VARCHAR

Parking

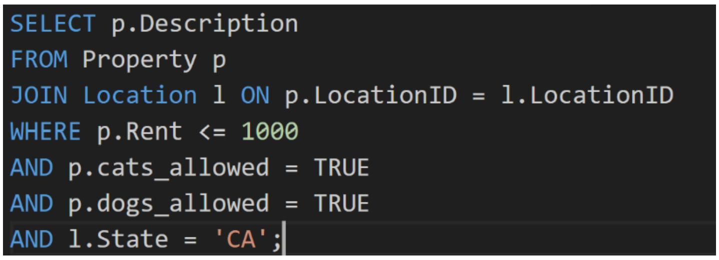
ParkingID (PK)	INT
ParkingName	VARCHAR

Demonstrate the SQL that you would write to enter a new line of data, covering all relevant attributes.



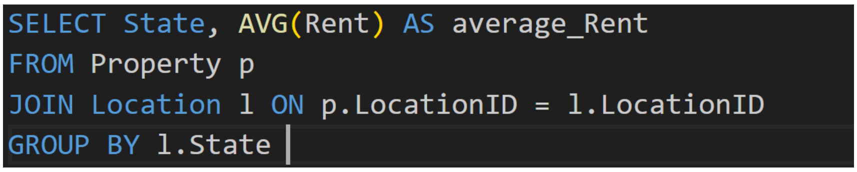
```
SELECT * FROM PROPERTY
INSERT INTO Property (ID, URL, Description, Rent, Sqfeet, Bedrooms, Bathrooms,
cats_allowed, dogs_allowed, smoking_allowed, wheelchair_access, electric_vehicle_charge,
comes_furnished, typeID, LocationID, LaundryID, ParkingID)
VALUES (1, 'https://cincinnati.craigslist.org/1.html', 'One bedroom apartment just 100 yards away
from Mt. Lookout Square', 425, 600, 1, 1, TRUE, TRUE, FALSE, FALSE, FALSE, TRUE, 1, 1, 1, 1);
```

Extract the 'description' for all properties with a rent equal to or less than 1000*, allows both cats and dogs, and is in the state represented by 'ca'.



```
SELECT p.Description
FROM Property p
JOIN Location l ON p.LocationID = l.LocationID
WHERE p.Rent <= 1000
AND p.cats_allowed = TRUE
AND p.dogs_allowed = TRUE
AND l.State = 'CA';
```

Extract the average rental value for each state so they can be compared.



```
SELECT State, AVG(Rent) AS average_Rent
FROM Property p
JOIN Location l ON p.LocationID = l.LocationID
GROUP BY l.State
```

References

[1] A. &. G. M. Mayorga, “Splatterplots: Overcoming Overdraw in Scatter Plots,” *Transactions on Visualization and Computer Graphics,* vol. 19, pp. 1526-1538, 2013.

[2] B. P. &. B. M. S. H. P. Vinutha, “Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset,” in *Advances in Intelligent Systems and Computing*, Springer, 2018, pp. 511-518.

[3] P. Norvig, “Imbalanced Data, Google Machine Learning Data Preparation and Feature Engineering in ML,” Google Machine Learning - Foundational Course, [Online]. Available: https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data. [Accessed 15 March 2024].

[4] H. F. Tayeb, M. Karabatak and C. Varol, “Time Series Database Preprocessing for Data Mining Using Python,” in *8th International Symposium on Digital Forensics and Security*, Beirut, 2020.

[5] W. W. B. d. l. I. Aliya Aleryani, “Multiple Imputation Ensembles (MIE) for Dealing with Missing Data,” *SN Computer Science,* vol. 1, no. 134, 2020.

[6] H. H. F. T. C. &. D. M. Liu, “Discretization: An Enabling Technique,” *Data Mining and Knowledge Discovery,* vol. 6, pp. 393-423, 2002.

[7] B. S. Dalwinder Singh, “Investigating the impact of data normalization on classification performance,” *Applied Soft Computing,* vol. 97, 2020.

[8] R. K. M. S. James Dougherty, “Supervised and Unsupervised Discretization of Continuous Features,” in *Machine Learning Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe, 1995.

[9] M. Smithson, “Models for fuzzy nominal data,” *Theory and Decision,* vol. 14, pp. 51-74, 1982.

[10] M. M. F. A. E. S. S. &. M. A. Amrehn, “The Random Forest Classifier in WEKA: Discussion and New Developments for Imbalanced Data,” *Computer Vision and Pattern Recognition,* 2018.

[11] Ç. İ. A. Gizen Mutlu, “SVM-SMO-SGD: A hybrid-parallel support vector machine algorithm using sequential minimal optimization with stochastic gradient descent,” *Parallel Computing,* vol. 113, 2022.

[12] G. S. R. S. Jaspreet Singh, “Optimization of sentiment analysis using machine learning classifiers,” *Human-centric Computing and Information Sciences,* vol. 7, pp. 1-12, 2017.

[13] L. T. C. H. H. H. F. &. L.-B. K. Kotthoff, “Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA,” in *The Springer Series on Challenges in Machine Learning*, 2017, pp. 81-95.

[14] M. Walesiak, “Przegląd wzorów na normalizację wartości zmiennych i ich wartości w wielowymiarowej statystycznej analizie wielowymiarowej,” *Przegląd statystyczny,* vol. 4, 2014.

[15] R. Maciejewski, R. Hafen, S. Rudolph, S. G. Larew and M. A. Mitchell, “Forecasting Hotspots—A Predictive Analytics Approach,” *Transactions on Visualization and Computer Graphics,* vol. 17, no. 4, pp. 440-453, 2011.

[16] T. M. K. J. L. L. &. R. A. B. Tawfiq Hasanin, “Severely imbalanced Big Data challenges: investigating data sampling approaches,” *Journal of Big Data,* vol. 6, no. 107, 2019.

[17] R. &. L. L. Blagus, “SMOTE for high-dimensional class-imbalanced data.,” *BMC Bioinformatics,* vol. 14, p. 106, 2013.

[18] E. A. S. D. B. A. Peng Li, “Multiple Imputation: A Flexible Tool for Handling Missing Data,” *JAMA,* 2015.

[19] D. G. &. M. G. Nir Friedman, “Bayesian Network Classifiers. Machine Learning,” *Machine Learning,* vol. 29, p. 131–163, 1997.

[20] H. Guy, “Google, Big Data, and Hadoop,” in *Next Generation Databases: NoSQLand Big Data*, Apress, 2015, pp. 21-37.

[21] T. M. &. D. S. Spencer Wheatley, “The extreme risk of personal data breaches and the erosion of privacy,” *The European Physical Journal B ,* vol. 89, no. 7, 2016.

[22] W. Roberds and S. Schreft, “Data Breaches and Identity Theft,” *Federal Reserve Bank of Atlanta Working Paper,* vol. 22, p. 54, 2008.

[23] L. Tosoni, “Article 4(12). Personal Data Breach,” in *The EU General Data Protection Regulation (GDPR) - A cimmentary* , New York, Oxford Academic, 2020, pp. 188-195.

[24] C. R. L. &. M. A. Knowen, “Prioritizing Personal Data Protection in Insurance Organizations.,” *Journal of Information Security and Cybercrimes Research.,* 2023.

[25] D. K. Ullah, “Comparison of Various Encryption Algorithms for Securing Data,” Islamabad, 2019.

[26] V. R. Joan Daemen, The Design of Rijndael: AES - The Advanced Encryption Standard, Brussels, Leuven: Springer Science and Data Media, 2013, pp. 31-35.

[27] H. H. Ali and S. H. Shaker, “Modified Advanced Encryption Standard algorithm for fast transmitted data protection,” *IOP Conference Series: Materials Science and Engineering,* vol. 928, 2020.

[28] M. S. a. S. Matwin, “HALT: Hybrid anonymization of longitudinal transactions,” *Eleventh Annual Conference on Privacy, Security and Trust,* pp. 128-134, 2013.

[29] E. O. o. t. President, “BIG DATA:,” The White House, Washington, 2014.

[30] A. W. D. R. O. U. G. Micah Altman, “Practical approaches to big data privacy over time,” *International Data Privacy Law,* vol. 8, no. 1, pp. 29-51, 2018.

[31] S. D. C. d. V. &. P. S. C. A. Ardagna, “Enhancing User Privacy Through Data Handling Policies,” *Data and Applications Security XX,* vol. 4127, pp. 224-236, 2006.