

# ShinyPET: A Predictive, Exploratory and Text RShiny Application using Airbnb data

Ang Su Yiin  
Singapore Management University  
suyiin.ang.2020@mitb.smu.edu.sg

Joey Chua  
Singapore Management University  
joey.chua.2020@mitb.smu.edu.sg

Kevin Gunawan Albindo  
Singapore Management University  
kgalbindo.2019@mitb.smu.edu.sg

## ABSTRACT

The increasing availability of data has resulted in the increased demand for data driven decisions. Although there is an extensive range of commercial statistical tools, they are often subscription-based and demand good technical knowledge to mine and draw insights from. Therefore, it may not appeal to the average user.

As such, our project aims to develop a user-friendly application that will enable users to make data-driven decisions without the need to understand programming languages or have extensive statistical knowledge. We will use Airbnb data as our baseline for this project - data generated is rich in information, which consists of structured, unstructured (textual), and location data.

With this application, users will be able to perform text analysis on review and listing data to generate more quantitative insights. The exploratory module allows users to identify interesting patterns based on selected variables. Findings from the exploratory module will be further augmented in the confirmatory module where selection of statistical methods will be guided based on user's chosen variables. Finally, the predictive module enables users to prepare and build a variety of prediction models without needing to have in-depth understanding of the predictive models and its algorithms.

## 1. MOTIVATION OF THE APPLICATION

xx

## 2. LITERATURE REVIEW

Review and critic on past works

Lu, Y., Garcia, R., Hansen, B. et al. (2017) [2] provides a comprehensive summary of research on Predictive Visual Analytics. The paper discusses how visual analytics systems are implemented to support predictive analytics pro-

cess such as feature selection, incremental learning, model comparison and result exploration. The overall goal of visual analytics is to support explanation in each step of predictive analytics exercise which is also our motivation in developing this application.

Radiant package, an open-source platform-independent browser-based interface for business analytics in R, illustrate the robustness of Rshiny for web-based application. Developed by Vincent Nijs [3] to promote quick and reproducible data analytics, the package provides interactivity and flexibility in performing predictive analysis. Most of the plots produced are of static nature and can be enhanced by wrapping plotly around them. In addition, more recent package such as visNetwork allows interactive tree visualisation which will improve the assessment of decision tree model.

In R community, Tidymodels [1] has gained interest by providing a framework for predictive modeling and machine learning. It is aligned with the tidyverse principles which leads to a tidier and consistent grammar in the predictive analytics process. Different models offered in Radiant package are also available for implementation in Tidymodels framework, which is why our application leverages Tidymodel as the main framework to conduct predictive analytics on Airbnb data.

## 3. DESIGN FRAMEWORK

A detail description of the design principles used and data visualisation elements built

### 3.1 Exploratory module

### 3.2 Text module

### 3.3 Predictive module

Our predictive module design framework follows Tidymodels framework for data pre-processing, model training, tuning, and validation. On top of that, feature selection are supported by other R packages such as ggcorplot (for correlation matrix), ranger and Boruta (for feature importance). The visualisation and interactivity are embedded in each step of predictive analytics as explained below.

Data sampling - Selection of training-test split proportion provides flexibility in deciding how to spend data budget on the model development process. The distribution plot

The screenshot displays a data science dashboard with a sidebar on the left and a main plot area on the right.

**Sidebar:**

- Choose response variable:** A dropdown menu with 'prior' selected.
- Training set (%)**: A slider set to 100%.
- Split data**: A button.

**Main Plot Area:**

- Numerical variables:** A row of three density plots for 'prior', 'posterior', and 'posterior\_predict'. The 'prior' plot shows a sharp peak at 0. The 'posterior' and 'posterior\_predict' plots show broader distributions centered around 10.
- Categorical variables:** A grid of plots for various categorical features:
  - Location\_type:** A bar chart showing counts for 'suburb' and 'city'.
  - Neighborhood:** A bar chart showing counts for 'suburb' and 'city'.
  - Neighborhood\_group:** A bar chart showing counts for 'suburb' and 'city'.
  - Property\_type:** A bar chart showing counts for 'suburb' and 'city'.
  - Room\_type:** A bar chart showing counts for 'suburb' and 'city'.
  - Host\_is\_superhost:** A bar chart showing counts for 'suburb' and 'city'.
  - Neighborhood\_group\_created:** A bar chart showing counts for 'suburb' and 'city'.
  - Host\_verification\_status:** A bar chart showing counts for 'suburb' and 'city'.
  - Latitude:** A bar chart showing counts for 'suburb' and 'city'.

Figure 1: Data sampling and distribution plot

Figure 2: Correlation matrix and variable importance

**Recipe for linear regression and lasso test tree**

**Data Recipe**

**Inputs:**

- role `ivar` variables
- id variable 1
- outcome 1
- predictor 34

**Operations:**

```

Correlation filter on all_numeric(), -all_outcomes()
Centering and scaling for all_nominal(), -id, -all_outcomes()
Collapsing factor levels for all_predictors(), -all_numeric()
Dummy variables from all_nominal(), -all_outcomes()
          
```

**Check transformed variables:**

original\_value processed

Figure 3: Data transformation steps

Figure 1 displays the Orange3 software interface, showing the workflow for building a decision tree model. The workflow includes data loading, preprocessing, feature selection, and model training. The 'Variable importance' plot shows the relative importance of each feature. The 'Decision Tree visualisation' plot shows the structure of the trained decision tree.

Figure 4: Training result evaluation

The figure consists of two parts. On the left is a scatter plot titled 'LM Prediction Result'. The x-axis is labeled 'Actual price' and ranges from 0 to 1000. The y-axis is labeled 'Predicted price' and ranges from 0 to 1000. A solid black diagonal line represents the line of perfect prediction. Data points are red dots, mostly clustered between 0 and 500 on both axes. One specific point is highlighted with a red dashed box and labeled: 'Actual: 792' and 'Predicted: 472.07'. On the right is a table titled 'Model validation result'.

.metric	.estimator	.estimate
mae	standard	31.08
mape	standard	40.63
rmse	standard	47.18
rsq	standard	0.80

Figure 5: Validation result evaluation

The screenshot displays a Shiny web application interface. At the top, there are two selection controls: "Select top N prediction error:" with a slider set to 10, and "Select top N predictor:" with a slider set to 3. To the right, the "p-value:" is set to 0.05. Below these controls, there are four diagnostic plots arranged in a 2x2 grid: "host\_acceptance\_rate" vs "price", "longitude" vs "id", "price" vs "value", and "property\_type\_hostel" vs "value". A vertical legend on the right side of the plots lists IDs 203, 316, 358, 431, 432, and 481, each with a red dot. At the bottom, a table displays the data for the selected top 10 prediction errors, with columns for id, pred\_error, price, host\_acceptance\_rate, longitude, and property.

id	pred_error	price	host_acceptance_rate	longitude	property
203	196.039529821819	421	0.609018923131641	-0.00391705108785079	1
316	394.351822870867	443	0.0910008044101326	-2.70138276406442	0
358	222.695232386823	415	-3.319815413898	0.22252806532509	0
431	189.05379105113	634	0.609018923131641	-0.349367943777026	1

Figure 6: Prediction error assessment

The figure displays a grid of 16 line plots, organized by `min_n` (columns) and `tree_depth` (rows). Each plot shows the mean performance metric against `cost_complexity` on a logarithmic scale. The legend indicates that the lines represent different `tree_depth` values: 3 (red), 4 (green), 5 (blue), and 7 (purple). A text box on the right specifies the parameters for the highlighted plot: `cost_complexity: 1e-04`, `mean: 64.555809`, and `tree_depth: 7`.

Figure 7: Hyper-parameter tuning result

### Best model performance comparison

The figure consists of four bar charts arranged in a 2x2 grid, comparing the performance of four models: LM, GLM, XGBoost, and DTree. The y-axis for all charts is labeled 'Estimate'.

- MAE (Mean Absolute Error):** The y-axis ranges from 0 to 30. LM and GLM have the highest MAE (around 35), while XGBoost and DTree have lower MAE (around 22).
- MAPE (Mean Absolute Percentage Error):** The y-axis ranges from 0 to 40. LM and GLM have the highest MAPE (around 45), while XGBoost and DTree have lower MAPE (around 18).
- RMSE (Root Mean Square Error):** The y-axis ranges from 0 to 60. LM and GLM have the highest RMSE (around 58), while XGBoost and DTree have lower RMSE (around 48).
- RSQ (R-squared):** The y-axis ranges from 0.0 to 0.8. DTree has the highest RSQ (around 0.8), followed by XGBoost (around 0.78), GLM (around 0.7), and LM (around 0.7).

Model	MAE	MAPE	RMSE	RSQ
LM	~35	~45	~58	~0.7
GLM	~35	~45	~58	~0.7
XGBoost	~22	~18	~48	~0.78
DTree	~22	~18	~48	~0.8

Figure 8: Models performance comparison

- use case

Data sets like Airbnb are rich with large numbers of variable. However, multicollinearity among variables are known to affect predictive model performance. Correlation matrix helps us to avoid such case by highlighting variables with high correlation value. In our example below, we observe correlations within rating score components, listing availability period, and review components. With this information, we

can then select our variables more wisely.

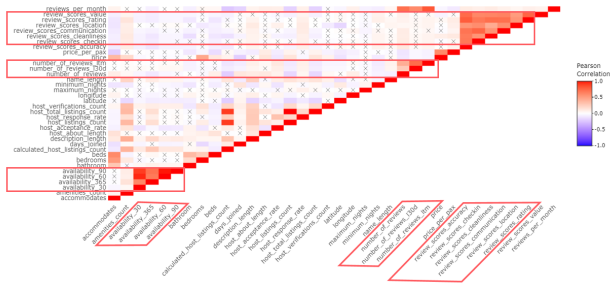


Figure 9: Correlation among variables

## 4.2 Model explanation

In predicting listing price using linear model, the plot of coefficient estimate helps to explain the trained model. In the example below, our interface allows sorting of variables based on p-value score where variables with lowest p-value is located on top. Property type which falls under “Others” category (those with counts of less than 5% in the data set) has the lowest p-value score and positive estimate, which may represent unique property type (e.g. boat, campsite, chalet, villa) where the listing price is above the average price of common property type like apartment and condominium (as shown in the boxplot from exploratory module). Amenities and beds are also in the top 5 predictor where it correlates positively with listing price. However, the error bar is wider for property type “Others” as compared to the amenities and beds, representing more uncertainty in the estimate value.

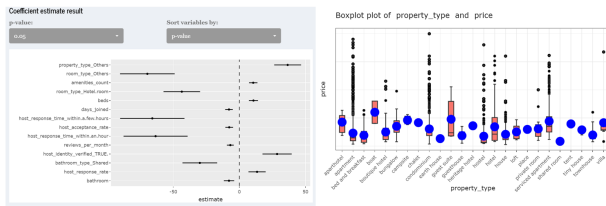


Figure 10: Coefficient estimate and boxplot from exploratory module

## 5. DISCUSSION

What has the audience learned from your work? What new insights or practices has your system enabled? A full blown user study is not expected, but informal observations of use that help evaluate your system are encouraged.

## 6. FUTURE WORK

The current predictive module is limited to 5 types of predictive model. In future, more predictive models can be added to the list, such as neural network to provide user with wider model selection. In terms of hyper-parameter tuning, parameters can be made available for user input to provide more flexibility in developing predictive model. In-depth statistical analysis in model training such as residual analysis are currently not available and this would be a good additional tool to improve our application.

## References

[1] Kuhn, M. and Wickham, H. 2020. Tidymodels: A collection of packages for modeling and machine learning using tidyverse principles.

[2] Lu, Y. et al. 2017. The state-of-the-art in predictive visual analytics. *Computer Graphics Forum*. 36, 3 (2017), 539–562.

[3] Nijs, V. 2019. Radiant – business analytics using r and shiny.