Predicting Sydney Property Sales

Report Part II

Yenul Weerabahu

July 2024

Contents

1 Introduction

The goal of this regression task is to predict the price of a property (`sellPrice`) using 15 feature variables. The task also involves predicting the uncertainty, prompting the use of distributional regression techniques. This dataset provides information on 207,267 property sales in the Sydney region from 1st January 2009 to 1st January 2022. Consequently the dataset contains 207,268 rows and 17 columns (including headers). However, the variable `Date` will only be used for data cleaning and indexing purposes, and not in the model. A data dictionary is given in Table 3.

2 Data Collection and Cleaning

Most of the property sales data was obtained from two datasets available on Kaggle by Alex Lau [2] and Mihir Halai [1]. The data were scraped off domain.com.au and realestate.com.au respectively. Additionally, as macroeconomic data could also influence property prices, the two features `cash_rate` and `property_inflation_index` were added for each property. The `cash_rate` is reported monthly by the Reserve Bank of Australia [3] and was added to each property depending on the date of the sale. The `property_inflation_index` is reported quarterly by the Australian Bureau of Statistics [4] and was added to each property based on which quarter the sale occurred in.

Data cleaning was necessary as there were many missing values and incorrect data types being used. The `Date` feature was formatted as a yyyy/mm/dd date. The values for `postalCode`, `bed`, `bath` and `car` features were interpreted as floats and needed to be converted to integers. Table 4 shows the missing data for each feature.

As the `property_inflation_index` data from the ABS is only available from the Quarter 4 of 2011, all property sales prior to this date had to be removed.

There were 1461 sales which did not have a `postalCode` feature. It was questionable whether most of these were actually geographically part of Sydney, or whether they were part of the 'Greater Sydney' region. For example, a property sale in Ourimbah was included despite the suburb being 61.95 km away from the CBD. Additionally, as postcodes are numbered in a hierarchical rather than an ordinal system, the median or mean could not be used. Hence, property sales with missing postcodes were removed from the dataset.

Following this, there remained 12,264 properties which did not have the suburb-specific features `suburb_population`, `suburb_median_income`, `suburb_sqkm`, `suburb_lat`, `suburb_lng`, `suburb_elevation` and `km_from_cbd`. Also, 18,151 properties did not have values for the feature `car` and 154 did not have the feature `bed`. As such, all the missing values for these 9 features were set to their respective median values using the `SimpleImputer()` transformer in Scikit-Learn.

Following this, there were 775 property sales with no `cash_rate`. Upon inspection, these were all from January 2019 and were replaced by the cash rate target of 1.50% set in December 2018.

There were also some questionable values for `sellPrice`, with 65 properties being sold at a price less than $100,000. These outliers were also excluded from the dataset.

The categorical variable `propType` had 23 possible descriptive values, with many overlapping. These were merged into the 10 categories defined below.

- House
- Vacant land
- Terrace
- Warehouse
- Apartment
- Duplex
- Acreage
- Other
- Semi-Detached
- Townhouse

Finally, as `Date` is not used in the model, it was dropped from the dataset.

3 Exploratory Data Analysis

Figure 1 illustrates the distribution of selling prices of Sydney properties since 2011 Q4. The prices were adjusted by the Residential Property Price Index (ABS, 2022) which indexed prices to 2011-12 levels. From the modal class, it is clear that most properties were sold at under $1,000,000 with the median adjusted

property price at $745,116. The horizontal axis of Figure 1 had to be limited to $6 million (adjusted) as there were a handful of properties with extremely high values, with the highest adjusted selling price at $109,243,698 in Point Piper. The right-tailed nature of the distribution of prices is indicative of the large inequalities in Sydney based on a property's features and location.

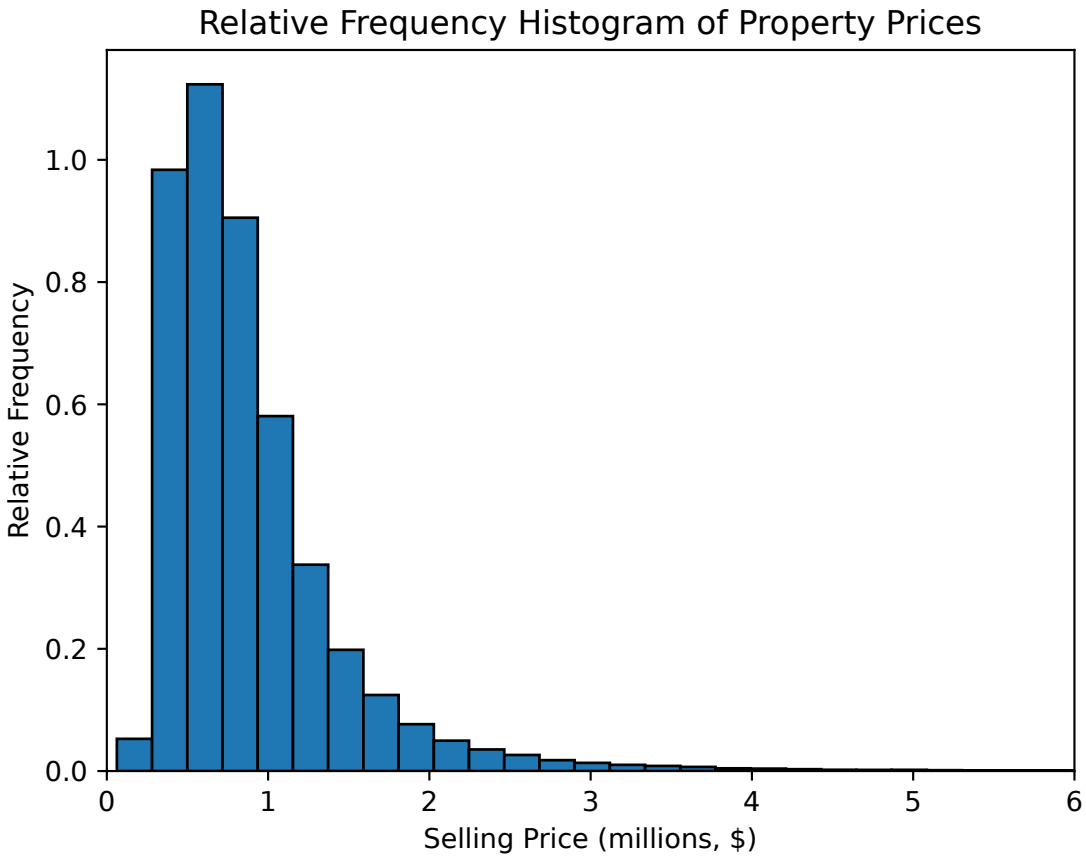


Figure 1: Distribution of Property Prices (Adjusted to 2011-12 $)

A correlation heatmap can be used to understand the correlation between feature variables and the target variable. Figure 2 shows how almost every feature is correlated to the target variable `sellPrice` (bottom row/right column). For a regression model `suburb_population`, `suburb_lat` and `suburb_elevation` could be excluded as the lack significant correlation with `sellPrice`. There is strong negative correlation between some feature variables such as `property_inflation_index` and `cash_rate` which is expected as lower cash rate targets typically do result in higher property prices. Other expected relationships include positive correlation between the number of bedrooms and the number of bathrooms as both generally are dependent on the floor area of the property. Interestingly, `sellPrice` is more correlated with `suburb_lng` than `suburb_lat`. This indicates that property prices are more determined by the horizontal distance to the CBD than vertical distance. Properties to the west with lower `suburb_lng` may have less transport connectivity and lower proximity to coastlines, both of which are usually associated with higher prices. The presence of varying degrees of correlation would cause significant issues of multicollinearity if a linear regression model was used, and highlights the advantage of using neural networks.

4 Model Performance Metrics

As this is a distributional regression task rather than a simple regression, the Negative Log-Likelihood (NLL) is used as it accounts for accuracy of both the mean and variance. This metric is superior is this context to the Mean Squared Error (MSE) which only accounts for the accuracy of point forecasts of the mean.

4.1 Negative Log-Likelihood (NLL)

The NLL is used to measure the goodness of fit of a model by comparing the observed data to the predicted probability distribution function. A log-normal distribution assumption for property prices will be used for the neural networks as `sellPrice` values are positive with some skewness and a heavy tail, evident in Figure 1. The probability density function for the log-normal distribution is,

$$f(y;\mu,\sigma) = \frac{1}{y\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log y - \mu)^2}{2\sigma^2}\right),$$

where:

- y is the observed value (`sellPrice`),
- μ is the predicted log-transformed values,
- σ^2 is the variance (estimated through the residuals for the Random Forest and as an output for the Neural Networks).

The NLL[1] is calculated through,

$$\begin{aligned}
\text{NLL} &= -\sum_{i=1}^{n} \log\left(f(y_i;\mu_i,\sigma_i)\right) \\
&= -\sum_{i=1}^{n} \log\left(\frac{1}{y_i\sigma_i\sqrt{2\pi}} \exp\left(-\frac{(\log y_i - \mu_i)^2}{2\sigma_i^2}\right)\right) \\
&= \sum_{i=1}^{n} \left(\log y_i + \frac{(\log y_i - \mu_i)^2}{2\sigma_i^2} + \log\left(\sigma_i\sqrt{2\pi}\right)\right).
\end{aligned}$$

This metric will be used as a loss function and/or for evaluation of the models in this task. Although the NLL is superior to MSE or MAPE in terms of quantifying point accuracy as well as epistemic uncertainty, it is much less interpretable. The NLL is best used when comparing models, rather than as an absolute measure for one model.

5 Benchmark Random Forest Model

For this regression task, a random forest was used as the baseline model. Random forests create multiple decision trees with the nodes being split based on a randomised selection of features. Random forests are complex models which are generally difficult to interpret, giving it no distinct advantage over neural networks in this regard. Additionally, there is high memory usage and is time-consuming, with this model taking 366 seconds of CPU time. A visualisation of the model can be seen in Figure 5.

This model contained every feature variable and log (`sellPrice`) as the target. The data was split into training and test sets, with the same test set being kept for later assessment of the neural network. Dummy variables were made for the categorical features (`suburb`, `postalCode` and `propType`). A random forest with 100 trees was fitted on the test set and the following results were obtained.

	Benchmark Random Forest
Training	14.7236
Test	15.1564

Table 1: NLL Values for Random Forest Model

A lower NLL is equivalent to a higher log-likelihood which is indicative of the probability distribution of the observed data being closer to the probability distribution of the true data — log-normal for this task. As expected, the NLL for the training set is better (lower values) than the test set, indicating some over-fitting. It must be noted that the numeric data were not standardised and irrelevant features were not removed, which could have potentially improved the model. Additionally, bagging and pruning could be used to diversify trees and control tree depth respectively. Alternatively, the out-of-bag error could have been used to estimate the standard deviation, rather than calculating the deviation of the residuals.

[1] As the benchmark random forest model does not output the standard deviation (σ), the standard deviation of the residuals was used as an estimate instead. Consequently, σ is constant for all outputs, and σ_i should be replaced with σ when calculating the NLL for the random forest. The neural networks were constructed to output both values — μ_i and σ_i.

6 Simple Feedforward Neural Network

The first neural network was a basic fully-connected feed forward network. Unlike the random forest, a pre-processing transformation pipeline was applied.

6.1 Pre-Processing

To pre-processs the inputs, one-hot encoding was applied to the three categorical features `suburb`, `postalCode` and `propType`. Entity embedding was considered, however other than reducing model fitting time, there was no improvement in the model's performance. Standardisation was applied to the remaining 12 numeric features using the `StandardScaler()` transformer in Scikit-Learn. This ensured that most values had a magnitude of less than 1, which is often expected by neural networks.

6.2 Model Architecture

This model has a simple architecture with three hidden layers, as evident in Figure 6. 112, 32 and 16 neurons were used for the first, second and third hidden layers respectively. As this was a distributional regression, this model was designed to have two outputs — mean and standard deviation. To do this, two separate values were obtained from the third hidden layer. These were concatenated using Keras' functional API so that the output would be an array with two columns. `leaky_relu` was used as the activation function for all three hidden layers, while `selu` and `softplus` were used for the mean and standard deviation respectively.

To optimise the values for the mean and standard deviation, a custom NLL loss function was made. This function used the two outputs from the neural network (in the log-scale) and the observed `sellPrice` to compute the log-likelihood. To minimise the negative mean of this value, μ and σ would be optimised. Hyperparameter tuning over 10 trials was used to select the number of neurons for the first hidden layers, the activation function for the layer computing the mean, and the learning rate. Bayesian optimisation was used for the search as it was more efficient than random or grid searches, and it balances exploration (where uncertainty is high) with exploitation (prior knowledge of high performance). Further details of the hyperparameter tuning are in Table 5 and 6.

After hyperparameter tuning, the best model was fitted on the training set with early stopping enabled. The performance was quite stable over the 39 epochs, with the loss curve shown in Figure 7. The NLL loss value stabilised after 10 to 20 epochs.

7 Wide and Deep Neural Network

The second neural network was a wide and deep network with a skip connection from the input layer to the fourth hidden layer. The same pre-processing which was used for the first neural network in 6.1 was applied. The wide (shallow) component allows for "memorisation" of frequent co-occurences of features. This allows common patterns between the features and the selling price to be captured. The deep component allows for "generalisation" to unseen data through multiple layers of non-linear transformations.

7.1 Model Architecture

This model's architecture was based off the first neural network, with the deep path being a replication as 112, 32 and 16 neurons were used for each of the three hidden layers respectively. The only exception was the presence of three dropout layers after each hidden layer. The wide path included a skip connection which was concatenated with the deep path after the third dropout layer. Together, both layers split into mean and standard deviation before being concatenated again to form the output layer. Similar to the simple feedforward network, the same NLL loss function was used. More details of the model can be seen in Figure 8.

Hyperparameter was again used through Bayesian search. In this case, the parameters tuned included the learning rate, the number of neurons in the third dense layer and the rate of dropout in the third dropout layer. Further details of the hyperparameter tuning are in Table 5 and 6.

Unlike the first network, the loss curve in Figure 9 does not decrease immediately. However, after 15 epochs, the curve is practically constant with no visible improvements in the loss.

8 Results

A summary of the results from all three models are given below in table 2.

	Benchmark Random Forest	Simple Feedforward Network	Wide and Deep Network
Training	14.7236	14.6975	14.6964
Validation	-	14.6912	14.6895
Test	15.1564	14.6949	14.6947

Table 2: NLL Values from all Models

Both neural networks performed better than the random forest on both the training and test sets, especially the later. After hyperparameter tuning the results for both the simple feedforward, and wide and deep networks have improved marginally. The wide and deep network had a lower NLL on all three sets, with its best performance in the validation set. Surprisingly, the neural networks had a lower NLL for the test set than the training set, with the simple feedforward network having the greatest improvement. This was less surprising for the wide and deep network as the wide path would have improved the model's flexibility in adapting to the unseen data. Additionally, this second network had more regularisation through dropout layers which randomly removes neurons so that the model is not reliant on a specific pathway. The use of early stopping would also help in regularisation for both networks as it stops training before the model begins to capture noise. It must also be noted that bootstrapping was disabled for the random forest causing it to be more prone to over fitting.

Overall, justifying the use of a particular model depends on performance as well as computational expense. The random forest took 366 seconds to fit, whilst the neural networks took 300 to 420 seconds to fit onto the training data once the appropriate model was chosen. However, the hyperparameter tuning took 1h 44m and 2h 14m for the first and second network respectively. Further improvements to the model could have been made such as tuning of more hyperparameters or the use of ensembles. This would lengthen model selection and fitting time despite the improvements possibly being marginal.

8.1 Potential Ethical Concerns

Although the data for this task was obtained legally and is available in the public domain, there remains ethical concerns about the misuse or misunderstanding of AI models. Using predictive models could enable real estate professionals or investors to gain unfair insights into property valuations, which less informed buyers or market participants do not have. Additionally, the models inherently consist of aleatoric uncertainty from the data and epistemic uncertainty from wrong distributional assumptions or other model errors. Alternatively, the data bias could perpetuate discriminatory outcomes where negative suburb-specific features reflect demographic inequalities. These factors, among others, must be considered when using such models for academic or industry purposes.

9 Appendix

9.1 Data Collection and Cleaning

Variable Name	Description	Datatype
`Date`	Date of property sale	ordinal categorical
`suburb`	Suburb of the property	nominal categorical
`postalCode`	4 digit postcode of the property	nominal categorical
`bed`	Number of bedrooms in the property	ordinal categorical
`bath`	Number of bathrooms in the property	ordinal categorical
`car`	Number of enclosed car spaces/garages in the property	ordinal categorical
`propType`	The property type i.e., house, apartment etc.	nominal categorical
`suburb_population`	Population of the suburb of the property	numeric
`suburb_median_income`	Median individual income of the suburb of the property	numeric
`suburb_sqkm`	Size of the suburb of the property measured in m^2	numeric
`suburb_lat`	Latitude of the suburb of the property	numeric
`suburb_lng`	Longitude of the suburb of the property	numeric
`suburb_elevation`	Elevation of the suburb of the property measured as metres above sea level	numeric
`cash_rate`	Official cash rate target (%) set by the RBA on the date the property was sold	numeric
`property_inflation_index`	The residential property inflation index in Sydney calculated by the ABS in the quarter the property was sold in	numeric
`km_from_cbd`	Linear distance measured in km from the suburb in which the property was sold in to the centre of the CBD	numeric
`sellPrice`	*The nominal price in $A at which property was sold for*	*numeric*

Table 3: Data Dictionary

Feature	Number of Missing Values
Date	0
suburb	0
postalCode	1461
bed	154
bath	0
car	18151
propType	0
suburb_population	14479
suburb_median_income	14479
suburb_sqkm	14479
suburb_lat	14479
suburb_lng	14479
suburb_elevation	14479
cash_rate	867
property_inflation_index	32499
km_from_cbd	14479
sellPrice	0

Table 4: Missing Values

9.2 Further Exploratory Data Analysis

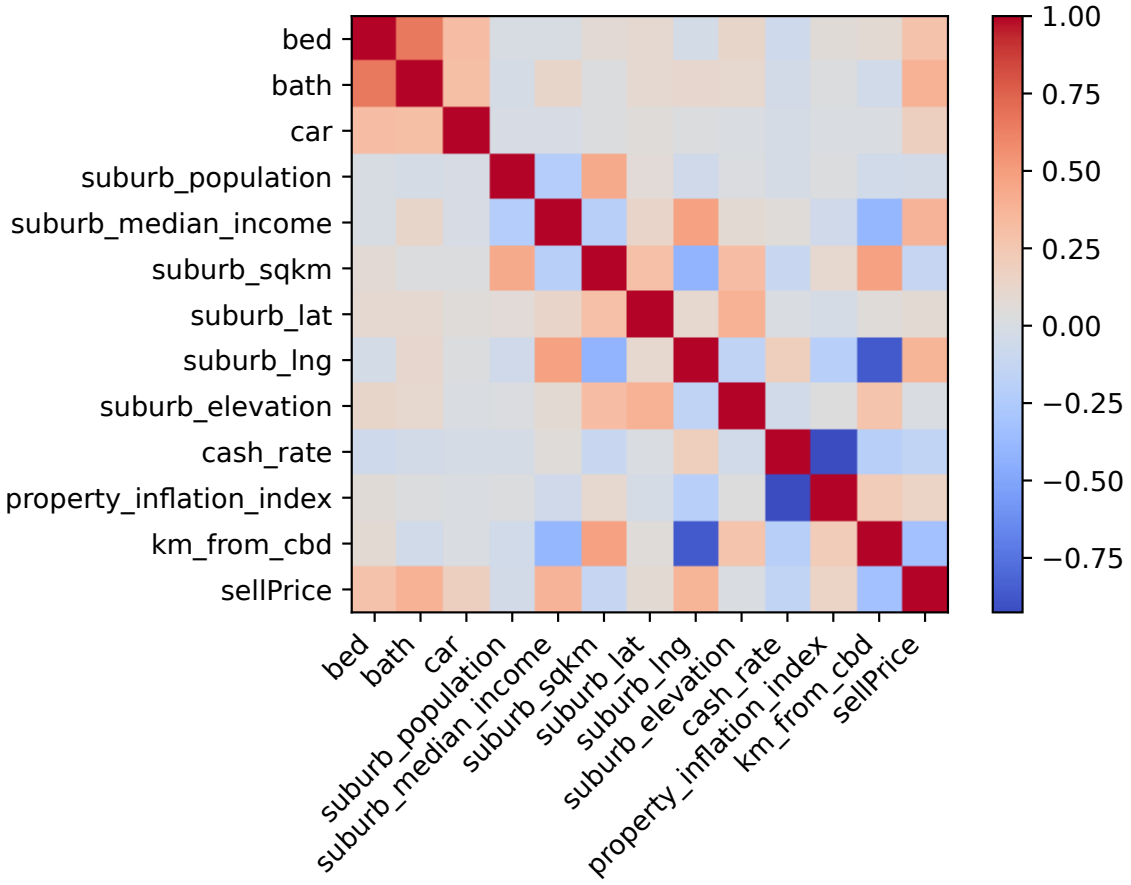


Figure 2: Correlation Heatmap Between Property Price Features

The scatterplot in Figure 3 emphasises how property prices decrease as distance from the CBD increases. As expected, there are many more properties in close proximity to the CBD due to high-rise apartments. Additionally, most of the ultra-expensive, luxury properties (above $10 million) are located closer to the CBD. Additionally, there are some very expensive properties further away from the CBD which are large estates that may have large property sizes and/or development potential.

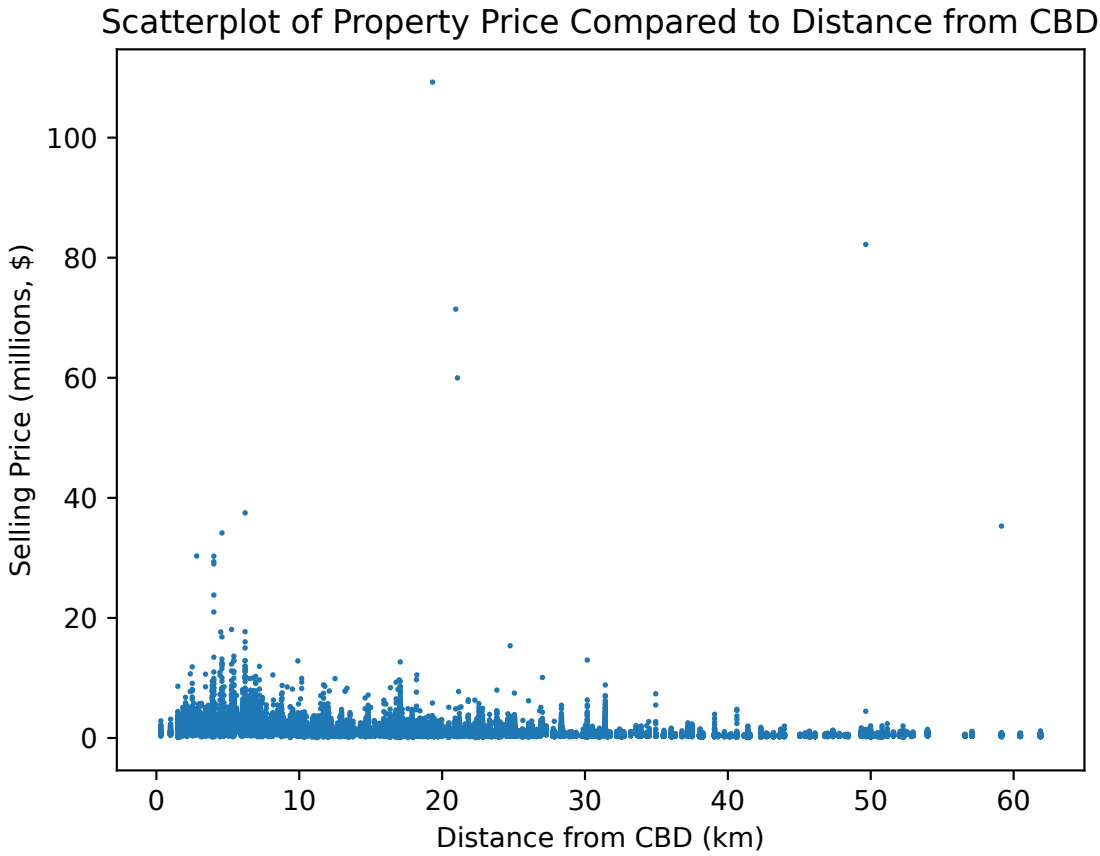


Figure 3: Property Prices (Adjusted to 2011-12 $) vs. Distance to CBD (km)

These property types can be observed in Figure 4 which shows that acreage properties, which are typically further away from the CBD, have the highest mean selling price. Vacant land also have high selling prices due to their development potential since Sydney is undergoing rapid housing and apartment development. Townhouses have the lowest mean price as some may consider them less desirable compared to apartments or duplexes (realestate.com.au, 2022).

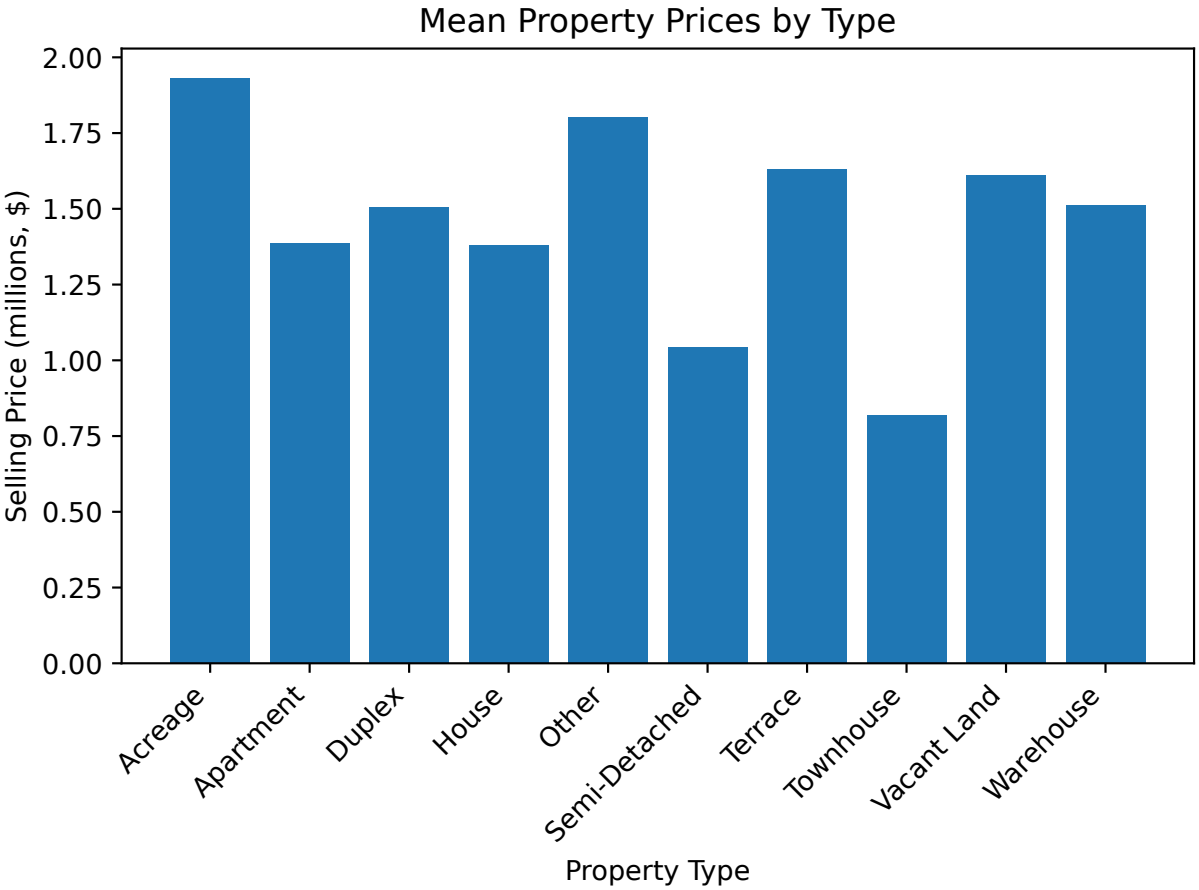


Figure 4: Column Graph of Property Prices by Type

9.3 Benchmark Random Forest Model

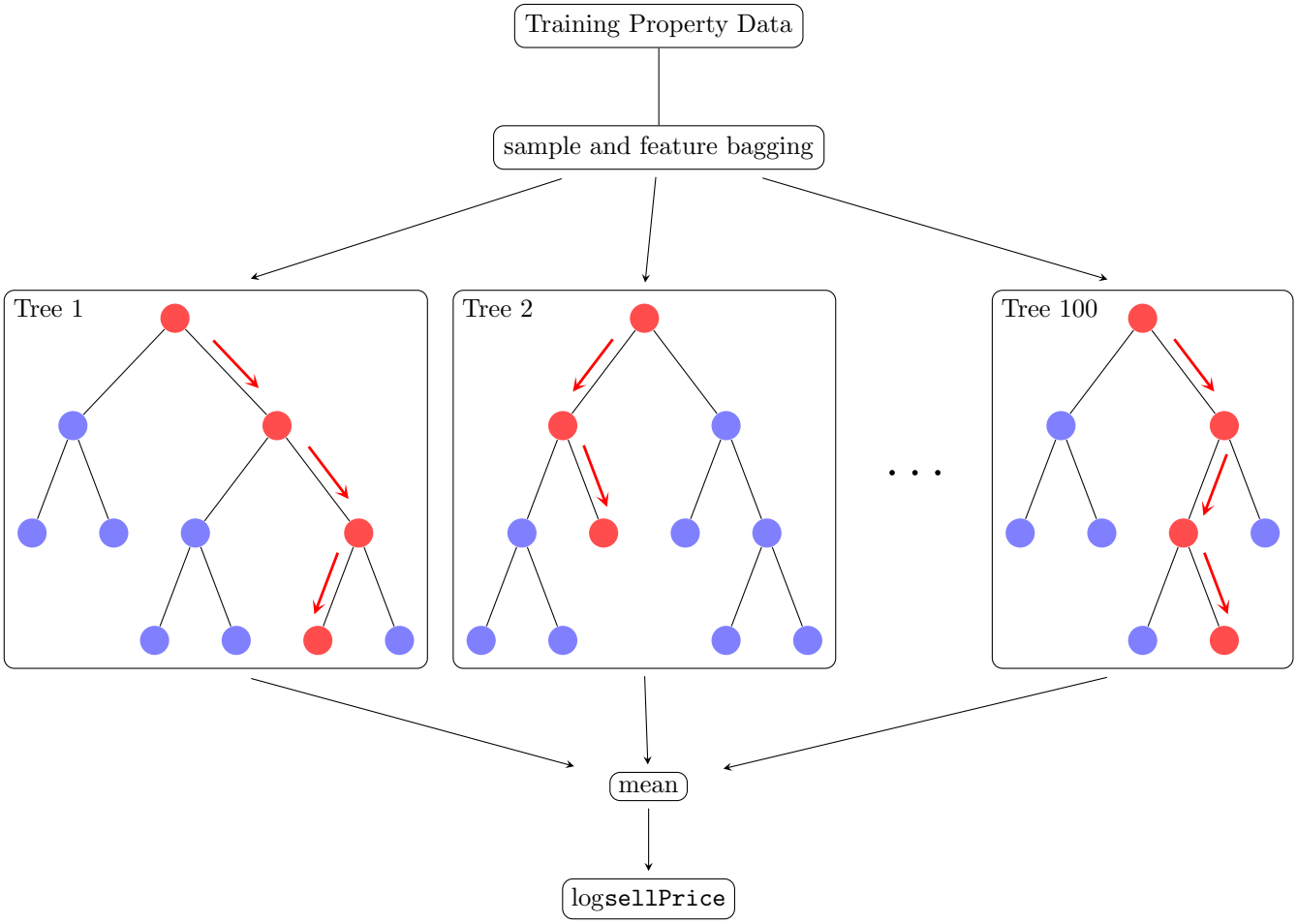
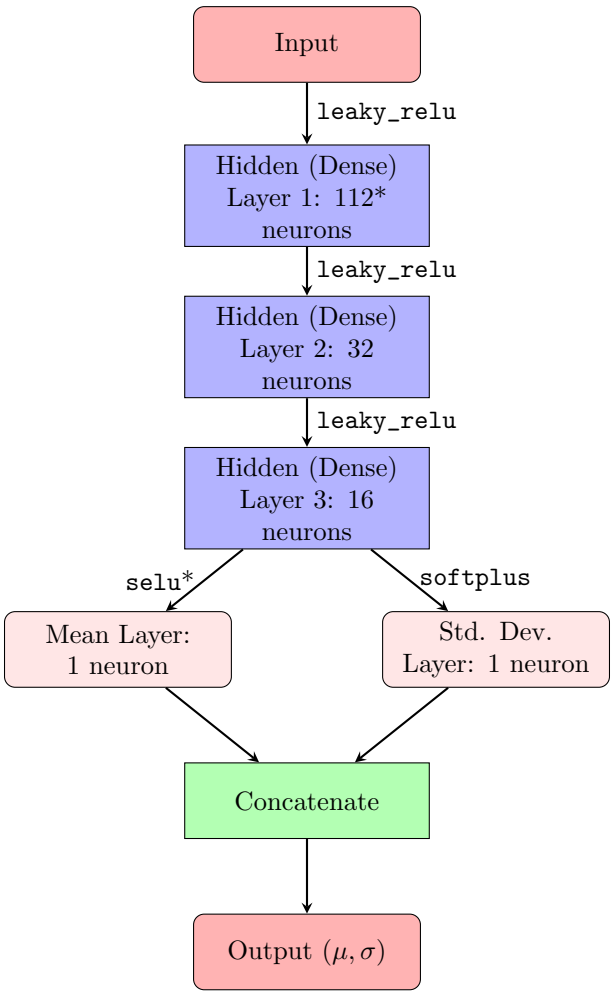


Figure 5: Diagram of the Random Forest (Adapted from Riebesell [5])

9.4 Simple Feedforward Neural Network



* Determined through tuning detailed in Section 9.6.

Figure 6: Structure of the Simple Feedforward Network

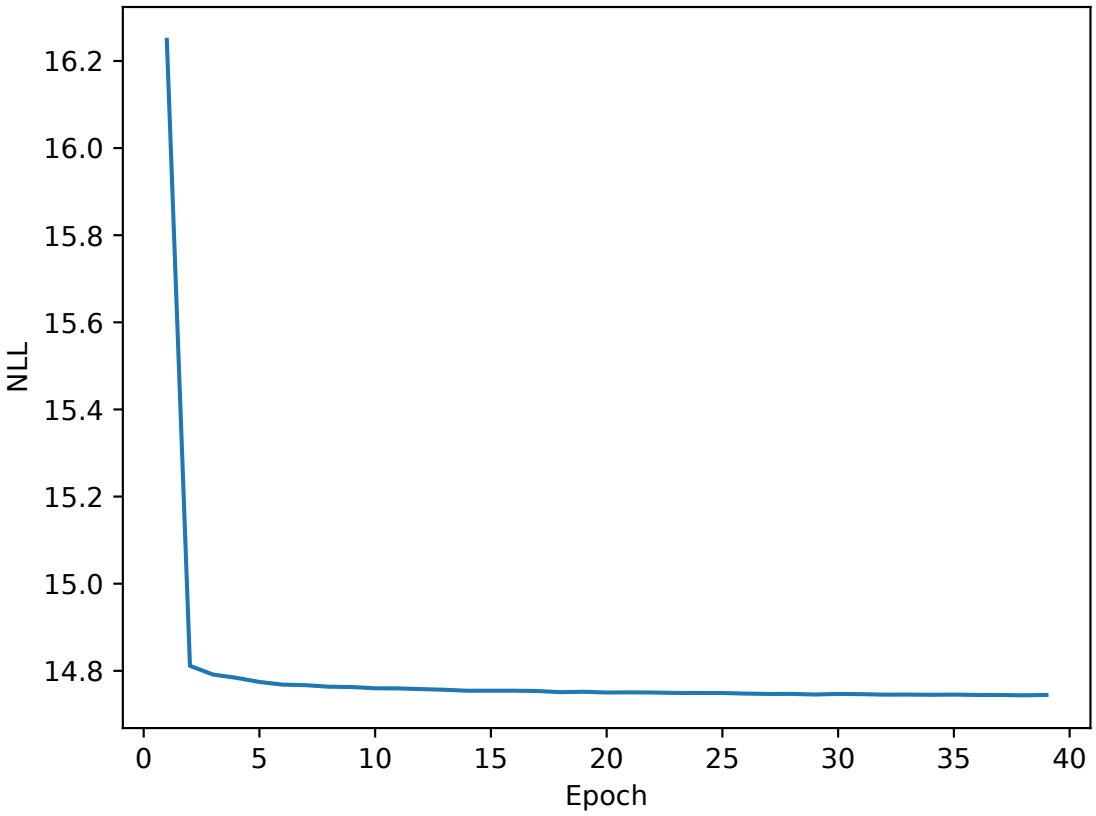
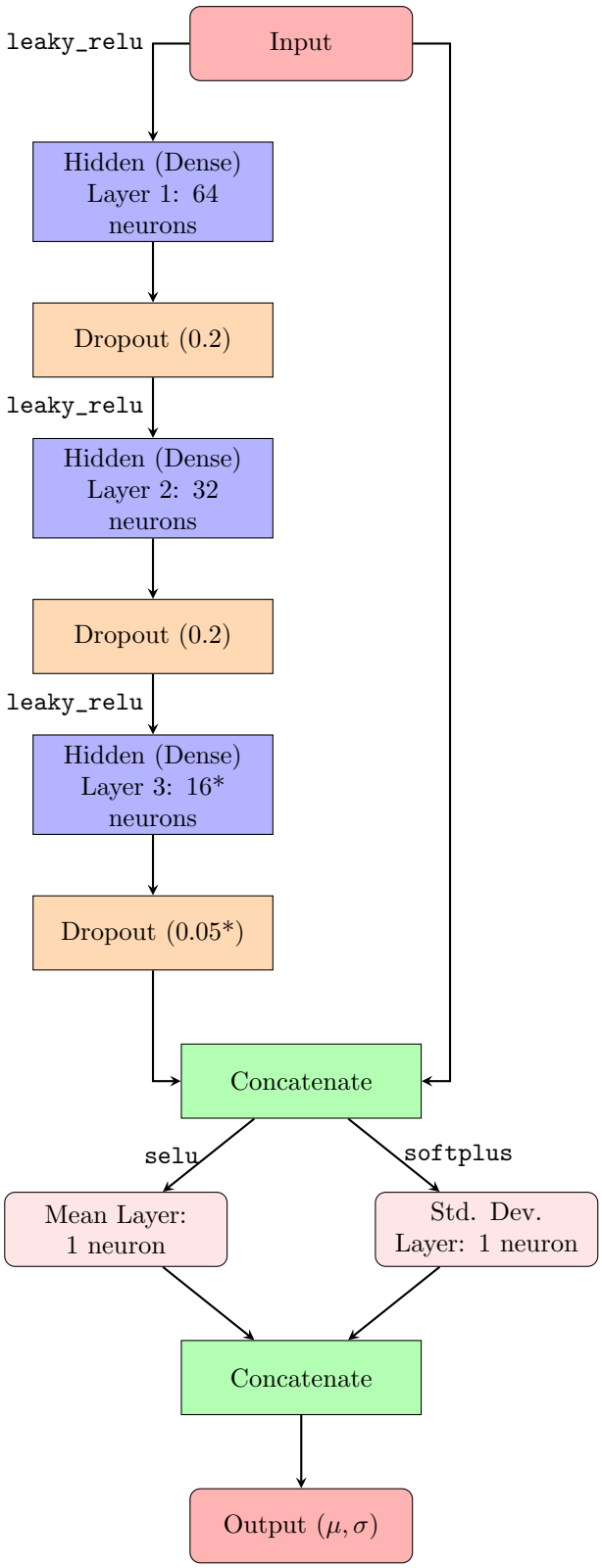


Figure 7: Loss Curve of Simple Feedforward Network

9.5 Wide and Deep Neural Network



* Determined through tuning detailed in Section 9.6.

Figure 8: Structure of the Wide and Deep Network

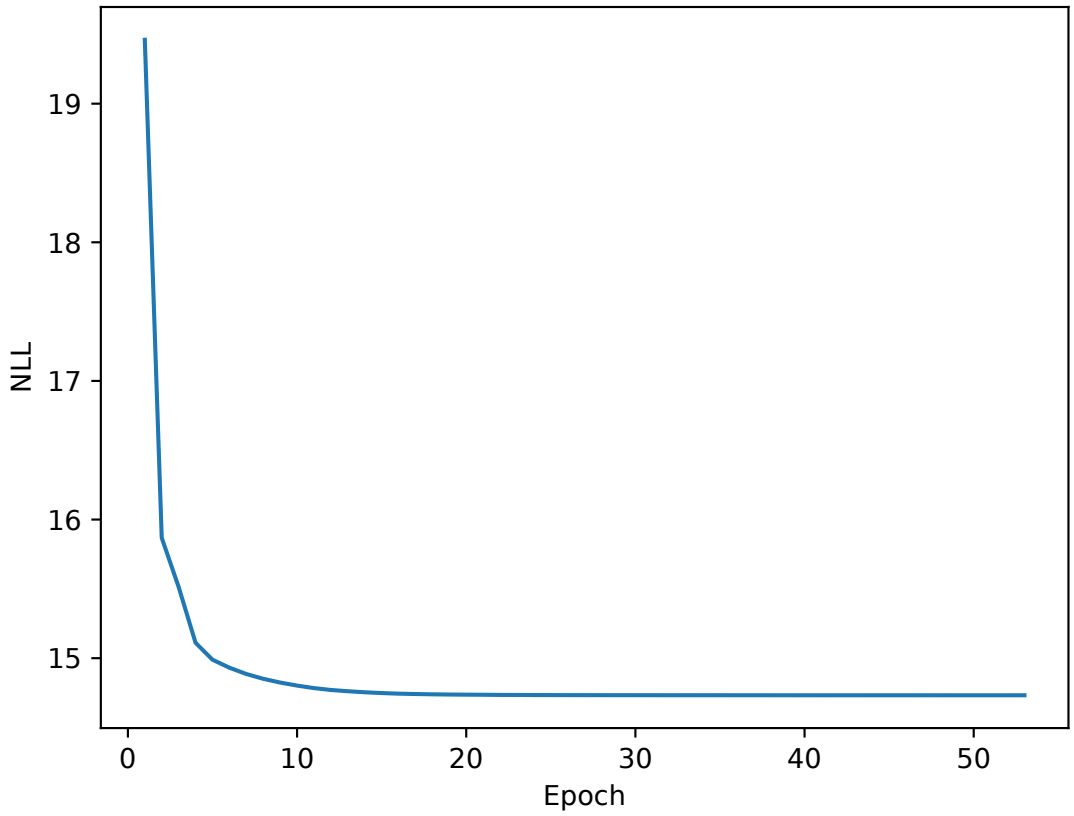


Figure 9: Loss Curve of Wide and Deep Network

9.6 Hyperparameter Tuning

Hyperparameter	Simple Feedforward	Wide and Deep
Learning Rate	$log[0.0001, 0.01]$	$log[0.0001, 0.01]$
Neurons in Hidden Layer 1	$[112, 128, 144, 160, 176, 192, 208, 224, 240, 256]$	-
Neurons in Hidden Layer 3	-	$[4, 8, 12, 16, 20, 24, 28, 32]$
Dropout Rate in Droput Layer 3	-	$(step\ 0.05)[0.0, 0.5]$
Mean Activation	[`softplus`, `linear`, `selu`, `exponential`]	-

$log[a, b]$ means the hyperparameter space is continuous from a to b (inclusive), sampled logarithmically; $[a_1, ..., a_n]$ means the hyperparameter space contains n discrete values from a_1 to a_n (inclusive); $(step\ x)[a, b]$ means the hyperparameter space contains values from a to b (inclusive), sampled with increments of x.

Table 5: Hyperparameter Ranges Across Models

Hyperparameter	Simple Feedforward	Wide and Deep
Learning Rate	0.0006045	0.0001789
Neurons in Hidden Layer 1	112	-
Neurons in Hidden Layer 3	-	16
Dropout Rate in Droput Layer 3	-	0.05
Mean Activation	`selu`	-

Table 6: Hyperparameters for the Two Neural Networks

9.6.1 Activation Function for the Mean Layer

Four possible activation functions were provided — `softplus`, `linear`, `selu` and `exponential`. The activation function needed to include $(0, \infty)$ as a subset of its range since the output was log(`sellPrice`).

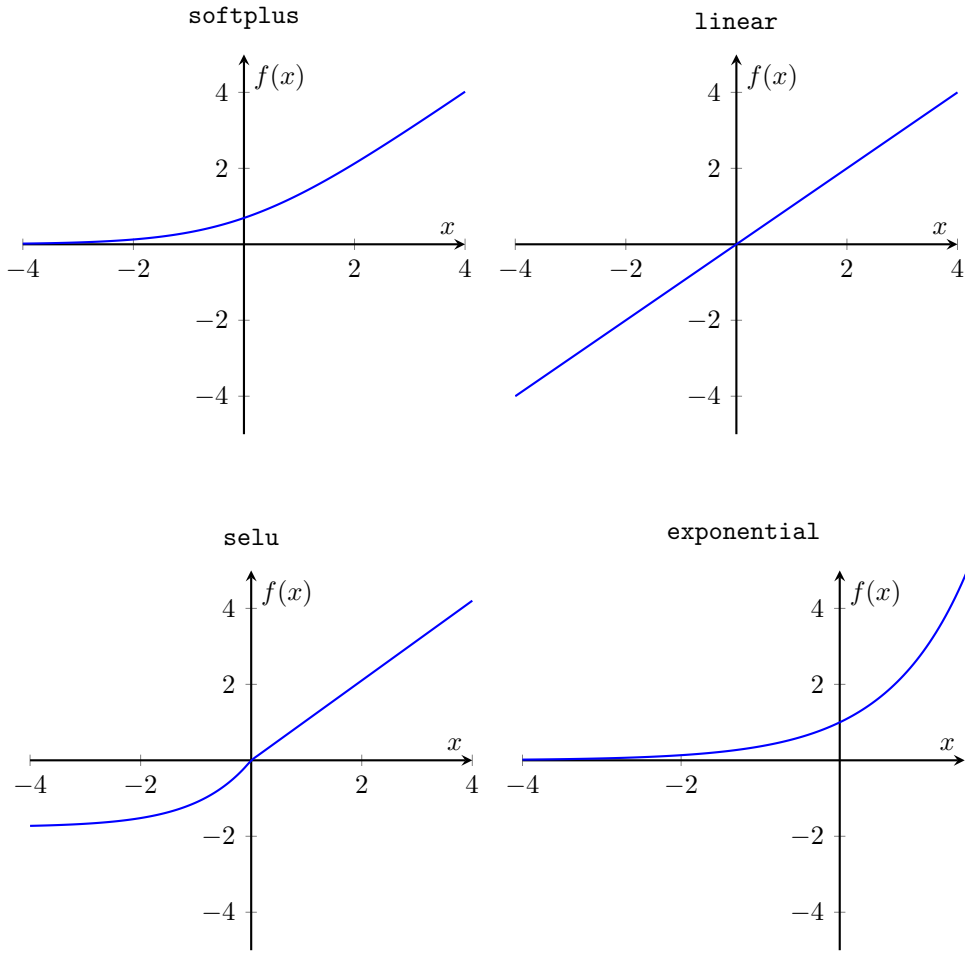


Figure 10: Activation Functions: `softplus`, `linear`, `selu`, and `exponential`

9.7 Generative AI Usage

Generative AI was used to create some of the code in the Jupyter notebook and some stylistic elements of the report. Examples of generative AI use include:

- downloading 'propertydata.csv' file into the notebook from the cloud,
- creation of the correlation heatmap in Figure 2 including specific design elements such as slanting horizontal axis labels for enhanced readability,
- creation of the column graph in Figure 4,
- calculation of the NLL such that it is compatible with TensorFlow (as detailed in the Jupyter notebook),
- creation of the four diagrams in Figure 10 in LaTeX.

References

[1] Mihir Halai. Sydney house prices, 2020. https://www.kaggle.com/datasets/mihirhalai/sydney-house-prices.

[2] Alex Lau. Sydney house prices, 2022. https://www.kaggle.com/datasets/alexlau203/sydney-house-prices.

[3] Reserve Bank of Australia. Cash rate target, 2024. Last accessed 18 July 2024.

[4] Australian Bureau of Statistics. Residential property price indexes: Eight capital cities, 2022. Last accessed 18 July 2024.

[5] Janosh Riebesell and Stefan Bringuier. Collection of standalone tikz images, 2020. 10.5281/zenodo.7486911 - https://github.com/janosh/tikz.