Predicting Property Prices

with neural networks

Yenul Weerabahu July 2024

Objective:

To predict Sydney property prices based on 15 features

- All properties sales and most feature variables were sourced from:
  - Kaggle by Alex Lau (2022) and Mihir Halai (2020)
- `cash_rate` was sourced from RBA and added to each property depending on which month the sale occurred
- `property_inflation_index` was sourced from ABS and added to each property depending on which quarter the sale occurred

Photo by Andrei J Castanha on Unsplash

	suburb	postalCode	bed	bath	car	propType	...	km_from_cbd	sellPrice
1	Prestons	2170	4.0	2.0	2.0	House	…	32.26	1087500
2	Kellyville	2155	4.0	3.0	2.0	House	…	30.08	1900000
3	Seven Hills	2147	7.0	3.0	2.0	House	…	26.58	1300000
4	Sydney	2000	2.0	2.0	1.0	Apartment	…	0.31	1025000
…	…	…	…	…	…	…	…	…	…

Table 1: Overview of Data

Data cleaning

Significant cleaning was needed to ensure no missing values

- 1461 sales which did not have postalCode feature were not geographically part of Sydney
- Missing values were set to median for:
  - 14,479 properties with no suburb-specific features
  - 18,151 properties with no car values
  - 151 with no bed values
- 65 properties with sellPrice below $100,00 were removed
- 23 categories for propType were merged into 10 categories
- Date removed as it will not be used in any models

Feature	Number of Missing Values
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 2: Missing Values

EDA

Property prices compared to distance from CBD

- Prices decrease further away from CBD
- Most ultra-expensive (above $10 million) properties are located close to the CBD
- There are some ultra-expensive properties further away from the CBD
    - Could possible by estates which large property area and/or development potential

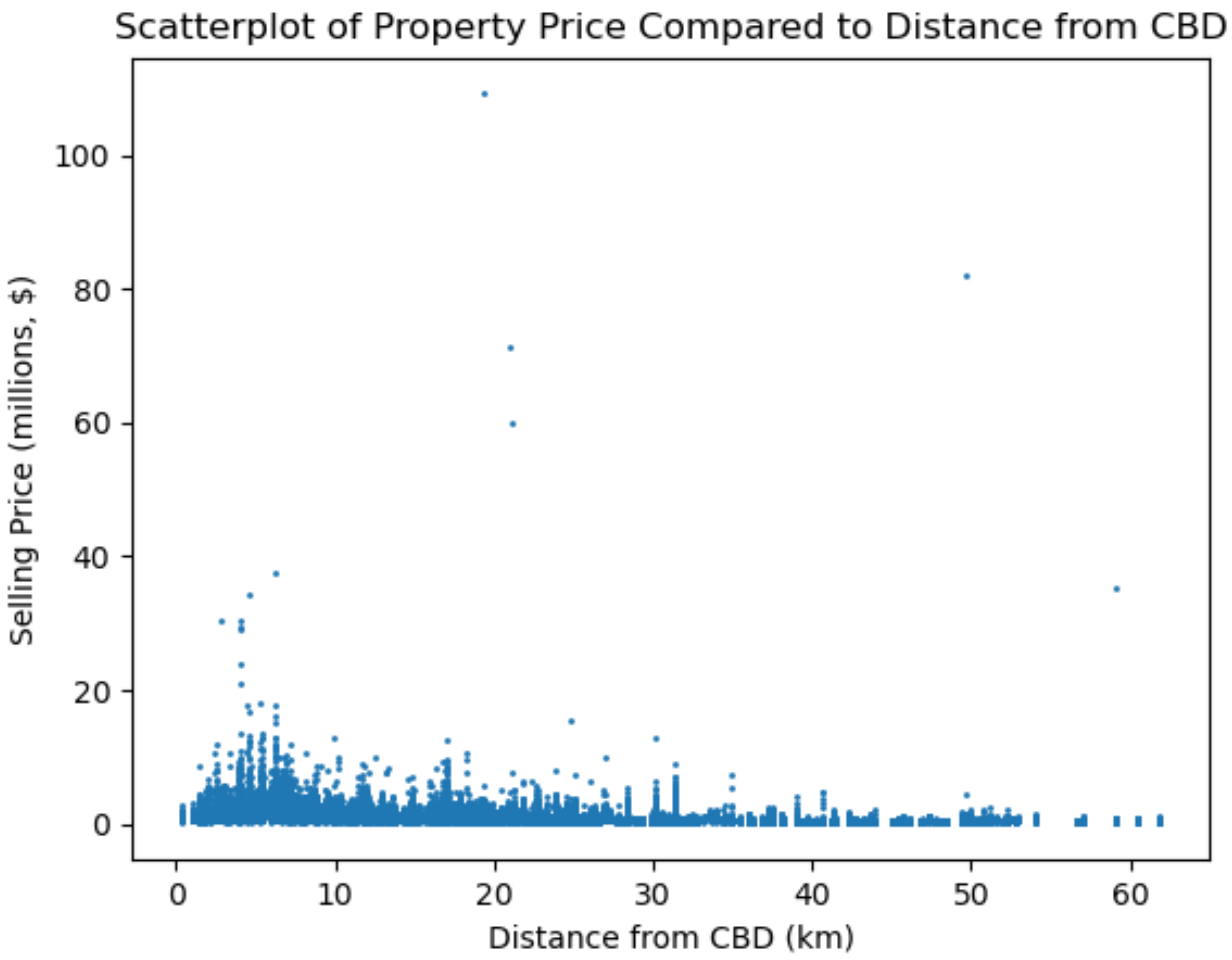

Figure 1: Property prices vs. distance to CBD (Adjusted to 2011-12 $)

Property prices by type

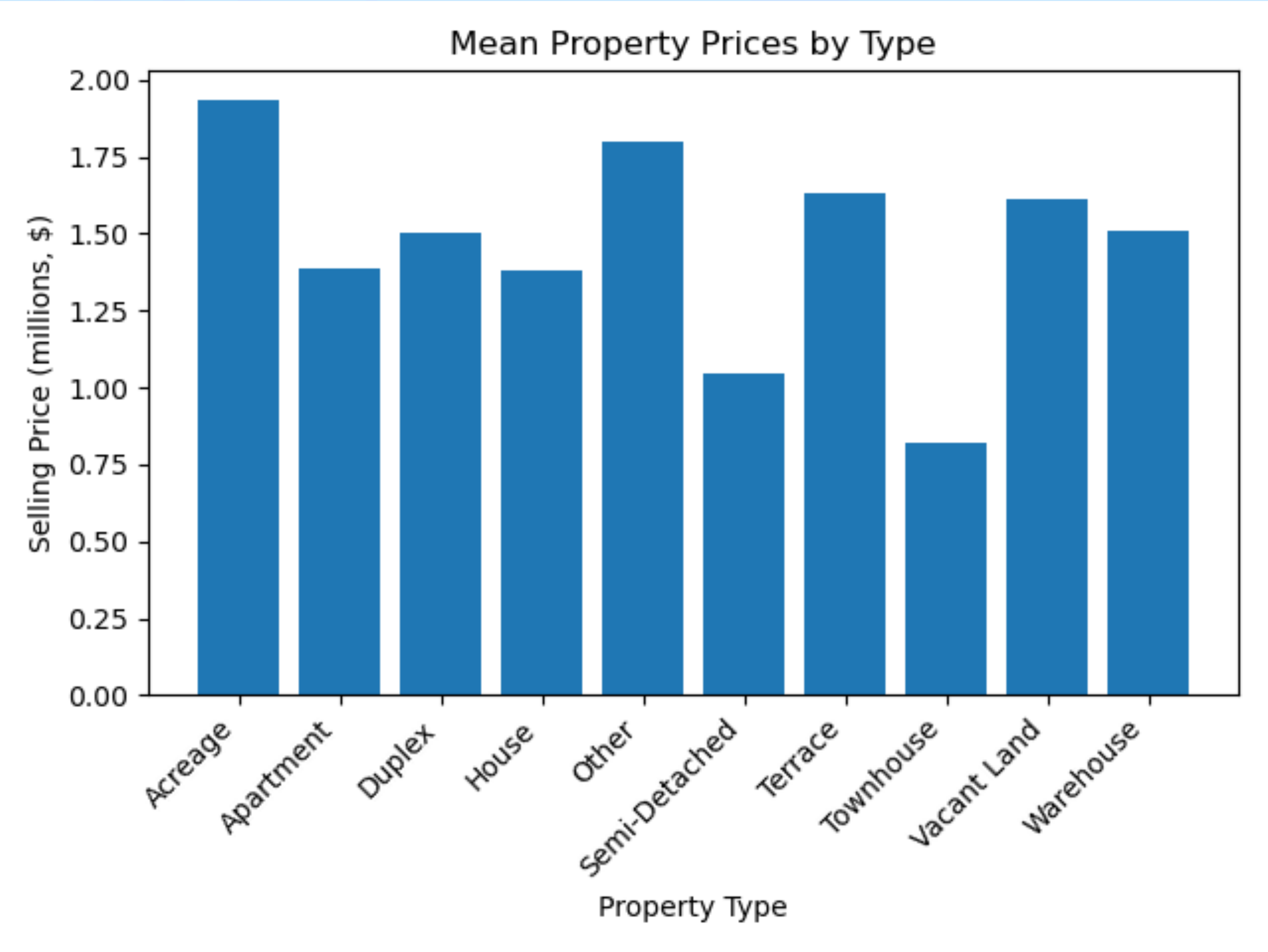

Figure 2: Property prices by type

- Acreage and vacant land are particularly more expensive
- Data included sales from 2010s during Sydney's rapid housing and apartment construction boom
- Townhouses have lowest mean price

Baseline model

Random forest

- A random forest was used with 100 trees and no maximum depth
- One-hot encoding used for features `suburb`, `propType` and `postalCode`

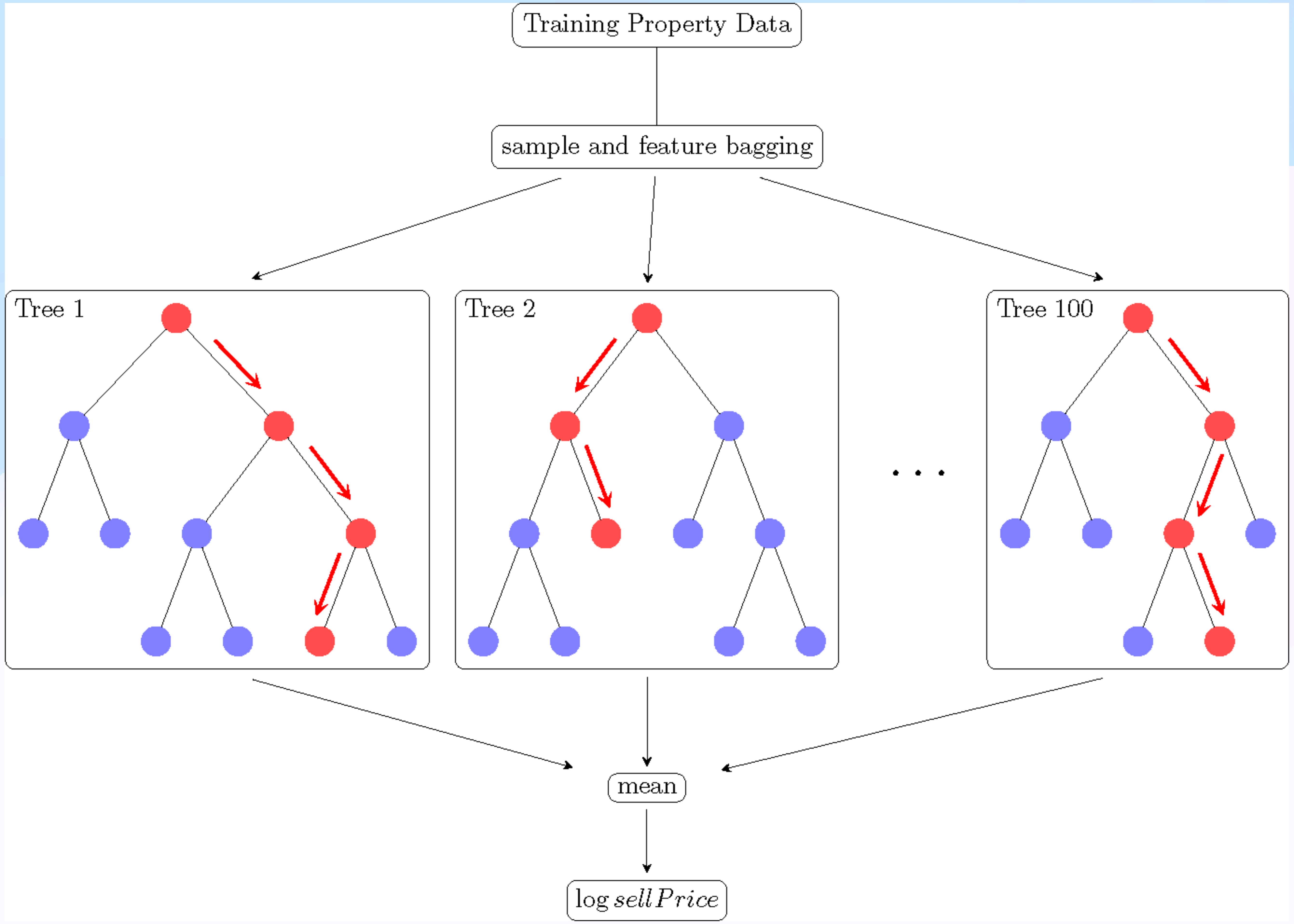

Figure 3: Random forest diagram

Metrics

- Negative log likelihood (NLL) used as metric instead of RMSE to account for both mean and variance

$$\text{NLL} = -\sum_{i=1}^{n} \log p(x_i \mid \theta)$$

- Continuous Ranked Probability Score (CRPS) is another possible metric as it compares predicted CDFs with observed values

$$\text{crps}(F, y) = \int_{-\infty}^{\infty} (F(t) - 1_{t \geq y})^2 dt$$

- *Assuming normally distributed errors*

Random forest results

- Total CPU time: 3 min 54 s

	NLL	CRPS
Training	14.7224	227263.1760
Test	15.1561	275414.0727

Table 3: Random Forest Results

Deep learning architectures

Neural network 1

- Basic fully-connected feedforward network
- One-hot encoding used for features `suburb`, `propType` and `postalCode`
- Standardisation used for all other features
- 61,441 trainable parameters
- “leaky_relu” used as activation function for all layers except output layer which used “softplus”
- Dropout layers used to reduce overfitting and improve training speed through faster convergence
- Early stopping with patience of 15 used although it ran for 57 epochs

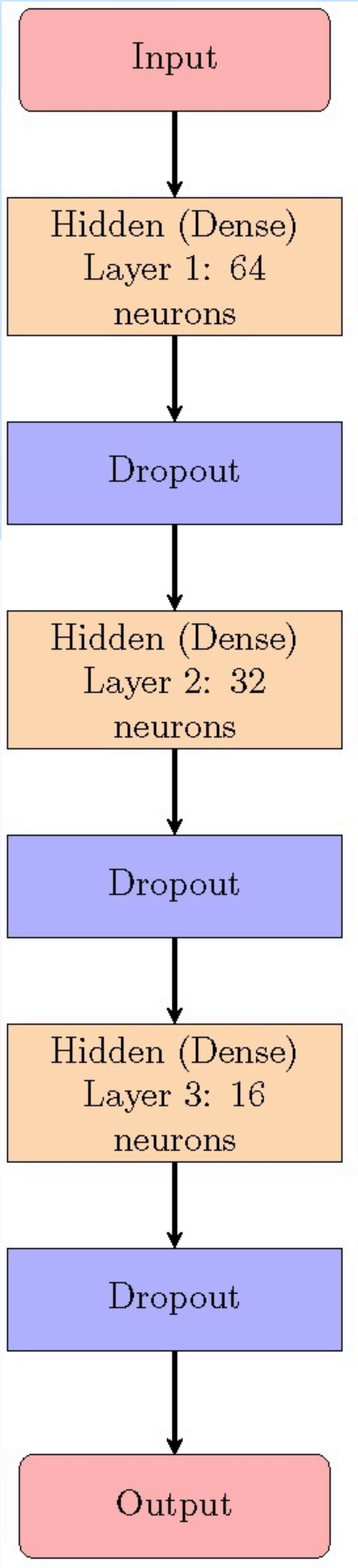
Input
Hidden (Dense) Layer 1: 64 neurons
Dropout
Hidden (Dense) Layer 2: 32 neurons
Dropout
Hidden (Dense) Layer 3: 16 neurons
Dropout
Output

Figure 4: Structure of the basic neural network

Neural network 2

- Wide and deep network with *skip connection* from input to output layers
- Wide (shallow) component allows for “memorisation” of frequent co-occurrences of features. Can capture common patterns.
- Deep component allows for “generalisation” to unseen data through multiple layers of non-linear transformations
- 62,359 trainable parameters
- “leaky_relu” used as activation function for all layers except output layer which used “softplus”
- Early stopping with patience of 15 used although it ran for 78 epochs

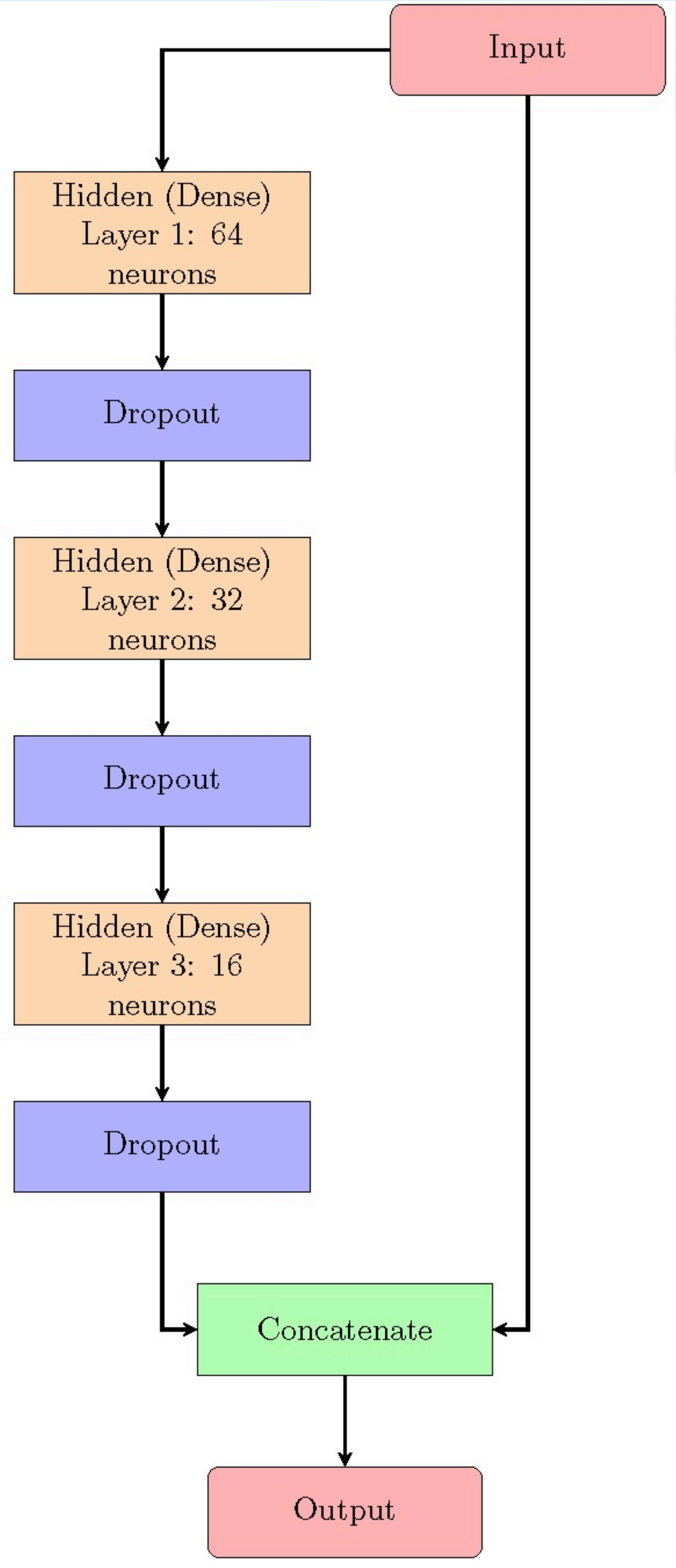
Input
Hidden (Dense) Layer 1: 64 neurons
Dropout
Hidden (Dense) Layer 2: 32 neurons
Dropout
Hidden (Dense) Layer 3: 16 neurons
Dropout
Concatenate
Output

Figure 5: Structure of the “wide and deep” neural network

Preliminary results

	Baseline	Neural Network 1	Neural Network 2
Training NLL	14.72	14.84	14.83
Training CRPS	227263.18	256927.22	259845.05
Validation NLL		15.26	15.08
Validation CRPS		315526.23	287644.64
Test NLL	15.16	15.15	15.14
Test CRPS	275414.07	258718.34	270710.36

Table 4: Comparison of Results