Introduction

This report aims to demonstrate the application of key machine learning principles by developing, training and evaluating three distinct machine learning models on a publicly available dataset. The three different models I have chosen for this report are; Decision Trees, Support Vector Machines (SVM), and Neural networks (Multi-layer Perceptron). I will be applying this to a supervised learning task, with the goal of comparing their performance, robustness, and sustainability for the task at hand. Through this project, various aspects of model development, such as data preprocessing, hyperparameter tuning, and performance evaluation, will be explored in depth.

Aim and Objective

The main aim of this project is to implement, tune, and evaluate different machine learning algorithms to compare their performance and interpret the results. The objectives of this assignment are as follows:

1. Data Preprocessing: clean and preprocess the datasets, addressing issues such as missing values, outliers, and categorical variables. Feature scaling will be applied as needed to ensure models like SVM and Neural Networks perform optimally.
2. Model development: implement three different machine learning algorithms and to train them on the same dataset, the three models include Decision Trees, SVM, and Neural Networks (MLP). Hyperparameter tuning will be applied to each model to ensure optimal performance.
3. Model evaluation: evaluate the models performance using appropriate metrics, such as accuracy, precision, recall, and F1 score for classification tasks. Cross validation will be used to assess the model robustness.
4. Comparison and analysis: analyse the strengths and weaknesses, and overall performance of each model, comparing their effectiveness in solving the same task. This will involve a detailed discussion of overfitting, underfitting, model complexity, and computational cost.
5. Report Writing: document the entire process in detailed and clear report, presenting key insights, results, and reflections on the methods used and challenges faced during the project.

Justification for the Selection of Methods

The decision to use Decision trees, SVM, and Neural Networks (MLP) is drien by the need to explore a wide range of machine learning techniques that offer varying levels of interpretability, flexibility, and complexity. Decision trees were chosen for their simplicity and their interpretability which makes them ideal for understanding feature importance and decision-making processes. SVM was selected for its ability to handle both linear and nonlinear data efficiently, especially with the use of kernel function to transform the feature space. Lastly neural networks (MLP) were chosen to explore more complex, flexible models that can capture intricate patterns in data. Together, these three methods offer a comprehensive comparison across different sections within machine learning, allowing for meaningful analysis and interpretation of results.

The Dataset

Within this section I will share the dataset I have chosen for this report, along with the reasoning behind choosing it.

The dataset chosen is the California Housing Dataset found from Kaggle*, which originates from the 1990 US Census. This dataset contains information about various attributes of California neighbourhoods, including features related to housing and geographical location. The primary goal is to predict the median house prices in different neighbourhoods based on these features. The dataset is large and diverse, providing a good basis for both regression and classification tasks, which will be looked at later in this report.

Key Features of the Dataset

The California Housing dataset contains 20640 rows and 10 columns, where each row represents a different neighbourhood or block in California. The dataset provides both categorical and continuous features, which is essential for training diverse machine learning models and the reason as to why I chose this dataset. The key features include:

1) longitude: A continuous variable representing the longitudinal coordinate of the neighbourhood.
2) latitude: A continuous variable representing the latitudinal coordinate of the neighbourhood.
3) housing_median_age: A continuous variable representing the median age of the houses in the neighbourhood.

4) total_rooms: A continuous variable representing the total number of rooms in all the houses within the neighbourhood.
5) total_bedrooms: A continuous variable representing the total number of bedrooms in all the houses within the neighbourhood.
6) population: A continuous variable representing the total population residing in the neighbourhood.
7) households: A continuous variable representing the total number of households in the neighbourhood.
8) median_income: A continuous variable representing the median income of households in the neighbourhood (in tens of thousands of dollars).
9) median_house_value: A continuous variable representing the median house price in the neighbourhood (in hundreds of thousands of dollars). This is the target variable for the regression task.
10) Ocean_proximity: A descriptive string representing the location of the neighbourhood with respect to the ocean.

Target variables for This Project

For the purposes of this project, two distinct tasks will be addressed:

1. Regression task: The target variable for regression is median_house_value, which represents the median price of hoses in the neighbourhood. The objective is to predict the continuous value based on other neighbourhood attributes.
2. Classification task: For the classification task, a bnary target variable will be created by classifying neighbourhoods as “expensive” or “affordable.” For example, neighbourhoods with a median house price greater $300,000 will be labelled as “expensive” (1) and those with prices less than or equal to $300,000 will be labelled as “affordable” (0).

Justification for Dataset Selection

The California Housing Prices Dataset was selected for two main reasons its size and diversity, which make it ideal for both regression and classification tasks. With over 20,000 rows, it provides enough data to train and evaluate complex models like Neural Networks and SVMs while remaining manageable for the purposes of this project. The datasets geographic and demographic features provide a rich set of variables that are essential for feature engineering and model evaluation. Moreover, it's a widely used dataset n machine learning, ensuring that the results obtained from the models can be compared to benchmarks.

Preprocessing Steps

The steps for preprocessing will be found in the attached Jupyter Notebook, within this Notebook there are explanations and comments made for this process.

Model Evaluation

Evaluation Metrics

We used several metrics to evaluate the performance of the three models on the test set, providing a thorough comparison:

- Accuracy: Measures the overall percentage of correct classifications.
- Precision: Indicates the proportion of true positives out of all positive predictions, reflecting how often the model is correct when predicting the "expensive" class.
- Recall: Shows the proportion of actual positives that were correctly identified by the model, measuring how well the model captures true positives.
- F1-score: The harmonic mean of precision and recall, giving a balanced view of the model's performance when both metrics are important. This is particularly useful when there's an imbalance between classes.

These metrics ensure that the models' ability to correctly classify "expensive" and "affordable" neighbourhoods is assessed across multiple dimensions, not just based on accuracy.

Cross-Validation for Robustness

Cross-validation is a critical step to ensure that the models generalize well to unseen data. We employed 5-fold cross-validation during hyperparameter tuning. This method reduces the risk of overfitting by training and testing the models on different subsets of the training data, ensuring a more reliable estimate of their real-world performance. The cross-validated scores were consistent with the test results, confirming that the models were robust and not overfitted to the specific data split.

Model Performance

The performance metrics for each model on the test set are shown in the table below:

Model	Accuracy	Precision	Recall	F1-score
Decision Tree	90.33%	0.89	0.87	0.88
SVM	91.64%	0.91	0.90	0.91
Neural Network	90.28%	0.88	0.89	0.88

Decision Tree

Strengths: The Decision Tree model offered interpretability with an accuracy of 90.33%. The model's structure makes it easy to understand which features influence the classification of neighbourhoods.

Weaknesses: Despite good performance, Decision Trees are prone to overfitting, even with tuned parameters like max_depth. Its F1-score reflects a slight imbalance between precision and recall, showing that the tree was not as effective in capturing true positives as SVM.

Support Vector Machine (SVM)

Strengths: SVM achieved the highest performance across all metrics. Its accuracy of 91.64% and balanced F1-score of 0.91 demonstrate its ability to generalise well across both "expensive" and "affordable" classes. The RBF kernel enabled SVM to model non-linear relationships in the dataset, giving it an edge over simpler models like Decision Trees.

Weaknesses: SVM can be computationally intensive, especially during hyperparameter tuning and training with non-linear kernels. Additionally, SVM's requirement for feature scaling adds complexity to the preprocessing pipeline.

Neural Network (MLP)

Strengths: The Neural Network (MLP) performed similarly to the Decision Tree, with an accuracy of 90.28% and a balanced F1-score of 0.88. Neural Networks are powerful for capturing non-linear patterns and complex relationships between features, making MLP a suitable model for this task.

Weaknesses: Neural Networks require more computational resources and training time, especially with larger datasets and deeper architectures. Despite its complexity, the model did not outperform SVM, possibly due to the relatively small size of the dataset.

Model Comparison

The results indicate that the Support Vector Machine (SVM) model outperformed the other models in all key metrics, making it the most effective model for this classification task. Its use of the RBF kernel allowed it to capture complex relationships between features and predict housing classifications more accurately.

SVM had the highest accuracy (91.64%) and balanced precision, recall, and F1-score, indicating strong performance on both "expensive" and "affordable" classes.

Decision Tree showed competitive performance, with an accuracy of 90.33%, but its reliance on splitting rules made it less flexible in capturing complex patterns.

Neural Networks (MLP) performed well but required more resources, and its slight underperformance compared to SVM suggests that more data or tuning might be necessary to fully leverage its potential.

Conclusion

Based on the evaluation, the Support Vector Machine (SVM) emerged as the best-performing model for this classification task. Its ability to generalize across non-linear boundaries and handle high-dimensional data made it particularly effective in classifying neighbourhoods as "expensive" or "affordable." Despite its computational demands, SVM provided the highest accuracy and F1-score, reflecting its balanced performance on both classes.

SVM's success can be attributed to the effective tuning of its hyperparameters (C and gamma), which control the margin and decision boundaries. The RBF kernel enabled SVM to capture complex relationships in the data that simpler models like Decision Trees could not.

Decision Trees, while interpretable, were slightly less effective at modelling complex interactions between features. However, their simplicity and transparency make them ideal for scenarios where interpretability is prioritised over raw performance. The model performed well with an accuracy of 90.33%, making it a viable option for less complex tasks.

Neural Networks (MLP), while powerful, require larger datasets to perform optimally. In this project, the dataset size may have limited its potential, but it still provided competitive results with an accuracy of 90.28%. Neural Networks offer flexibility in capturing complex patterns, but they require more careful tuning and computational power.

References:

*Nugent, C. (2017). California Housing Prices. Kaggle. https://www.kaggle.com/camnugent/california-housing-prices