ABSTRACT

This work consists of the implementation of supervised and unsupervised learning, Image recognition, and research of ethics in AI. Starting with Supervised and unsupervised learning, an imaginary dataset having sales of second-hand car in the UK has been provided in order to do price prediction of second-hand cars. Different regression models were used and compared by considering various combinations of features. An Artificial Neural Network was also developed to evaluate its performance with other supervising learning models, further hyperparameter tuning was also performed to achieve the best performance. Unsupervised learning implementation leads to the identification of clusters in the dataset. An appropriate evaluation metric was also used to justify the results. For image recognition, a Convolutional Neural Network model was made to recognize species of flowers from images. To train the model, a dataset from Tensor Flow has been used. Optimization of the model was also performed by hyperparameter tuning. The project was concluded with reviews of journals related to the Ethical implications of AI.

1. Employing Supervised & Unsupervised Learning

1.1. Introduction

Mock cleaned dataset consisting of second-hand car sales in UK has been provided, with a shape of (50000,7). Dataset consist of following features:

	Feature Name	Description of Feature	Feature Type
Independent Feature	Price	Price of the car in British Pounds (GBP)	Numerical
Dependent Feature	Manufacturer	Name of the manufacturer that produced the car	Categorical
	Model	Name of the model of the car	Categorical
	Engine Size	Size of the engine, in liters	Numerical
	Fuel Type	Type of fuel that the engine uses.	Categorical
	Year of Manufacture	Year of the car's manufacture	Numerical
	Mileage	Total distance the car has traveled, measured in miles	Numerical

Goal of this study is to do comparative analysis of accuracy of different models by predicting the price.

1.2. Supervised Learning

For evaluation, mean absolute error (MAE) (Bernico, n.d.) and Coefficient of Determination (R^2 score) (Anon., n.d.) is checked. A model is said to be robust if its Mean absolute error is low and Coefficient of Determination is closer to 1.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \qquad R^2 = 1 - \frac{RSS}{TSS}$$

1.2.1. Comparing linear & Non-linear model using single Numerical input feature

Dataset has 3 dependent numerical features, namely Engine size, Year of manufacture & Mileage. Simple linear regression and polynomial regression were applied on each set (Price vs. Mileage, Price vs. Engine Size & Price vs. Year of Manufacture) to predict price.

Linear Regression		
Features	**MAE**	**R^2**
Price vs Mileage	7964.78467	0.40131391
Price vs Engine Size	10817.49156	0.150625625
Price vs Year of Manufacture	7031.039209	0.511086524

Polynomial Regression (Degree 2)		
Features	**MAE**	**R^2**
Price vs Mileage	6409.911605	0.52235759
Price vs Engine Size	10807.26235	0.151263076
Price vs Year of Manufacture	5387.109075	0.609419402

From the result of both Linear Regression and Polynomial Regression (Degree 2), Price vs Year of Manufacture feature's MAE is lowest & R^2 score is highest. Therefore, we can say that **Year of Manufacture** variable is the best predictor for a car's price. Also, R^2 score of each feature in polynomial regression is better than Linear Regression thus we can conclude that **Polynomial Regression** is a better fit for price prediction.

1.2.2. Considering Multiple Numerical Variables

Numerical features like Engine size, year of manufacture & Mileage are all together used to predict the price. Both linear and polynomial model were applied.

Linear Regression		
Features	**MAE**	**R^2**
Price vs Mileage, Engine Size & Year of Manufacture	6091.458142	0.671456306

Polynomial Regression (Degree 2)		
Features	**MAE**	**R^2**
Price vs Mileage, Engine Size & Year of Manufacture	3196.824934	0.891991018

Inclusion of multiple numerical features significantly improves the accuracy of model. Previously for linear model highest R^2 score of 0.51 was recorded for the case of Year of Manufacture, now inclusion of all numerical feature improves the R^2 score to 0.67.

1.2.3. Considering All Input Features

In this case, in addition to Numerical features, Categorical features after label encoding are also considered to predict the price.

Linear Regression		
Features	**MAE**	**R^2**
Price vs All other features	6076.3458	0.671990366

Polynomial Regression (Degree 2)		
Features	**MAE**	**R^2**
Price vs All other features	2989.44386	0.906675288

Decision Tree Regression		
Features	**MAE**	**R^2**
Price vs All other features	486.1968	0.995799205

Random Forest Regression		
Features	**MAE**	**R^2**
Price vs All other features	332.2704	0.998246961

Slight improvement in accuracy is observed when including categorical features. Previously for linear case when only numerical features were considered R^2 score was 0.6714 which increases to 0.6719 when all features are included. For polynomial R^2 score increase from 0.891 to 0.906. Apart from linear and polynomial models, Decision Tree and Random Forest were also evaluated. Based on the results of all models, **Random Forest gives the best accuracy** with R^2 score of 0.99824.

1.2.4. Implementation of Artificial Neural Network

An Artificial Neural Network is made which takes 6 inputs both numerical and categorical features and give 1 output which is the price.

1.2.4.1. ANN Architecture

Constructor Stage	Compilation Stage
➢ Input layer with 64 units and ReLU activation ➢ To prevent overfitting, a drop layer with a rate of 0.2 ➢ 1 Hidden layer with 64 units and ReLU activation ➢ Output layer with 1 unit and Linear activation	➢ Adam Optimizer with default learning rate ➢ Early stopping with patience of 20 ➢ Validation dataset split of 10% ➢ Epochs equal to 200

Artificial Neural Network		
Features	**MAE**	**R^2**
Price vs All other features	1460.894	0.972150981

With a simple Artificial Neural Network consisting of default settings, R^2 Score of 0.972 is obtained which is less than previously used Random Forest Regression model where obtained R^2 score was 0.99824.

1.2.4.2 Hyperparameter Tuning

To obtain the best performance, hyperparameters were tunned using Keras Turner Random Search. Parameters choices were given based on which Random Search gives the best possible combination of parameters that maximizes output.

Choices given to Random Search

- Choice of hidden layers between 1 to 3
- Vary neurons from 32 to 128 in each hidden layer with step size of 32
- Activation Function is set as ReLU
- Dropout choices: 10%, 20% or 30%
- Output layer with 1 unit and linear activation
- Learning rate choices: 0.01 or 0.001
- Epochs set as 50
- Validation split of 0.1
- Early stopping with patience of 20

Tunned parameters provided by Random Search

Constructor Stage	Compilation Stage
➢ Input layer with 128 units and ReLU activation ➢ To prevent overfitting, a drop layer with a rate of 0.1 ➢ 2 Hidden layers with 128 & 64 units respectively and ReLU activation ➢ Output layer with 1 unit and Linear activation	➢ Adam Optimizer with learning rate of 0.01 ➢ Early stopping with patience of 20 ➢ Validation dataset split of 10% ➢ Epochs equal to 200

Artificial Neural Network (Hyperparameter Tunning)		
Features	**MAE**	**R^2**
Price vs All other features	899.512	0.992128014

After hyperparameters tunning, R^2 Score of 0.9921 was obtained. As compared to the models used earlier, ANN performed well except it fall short of Random Forest Model, which delivered R^2 score of 0.9982.

1.2.5. Best Model

Figure 1 shows the R2 score of the tested models, based on the R2 scores, Random Forest performs the best predictions.

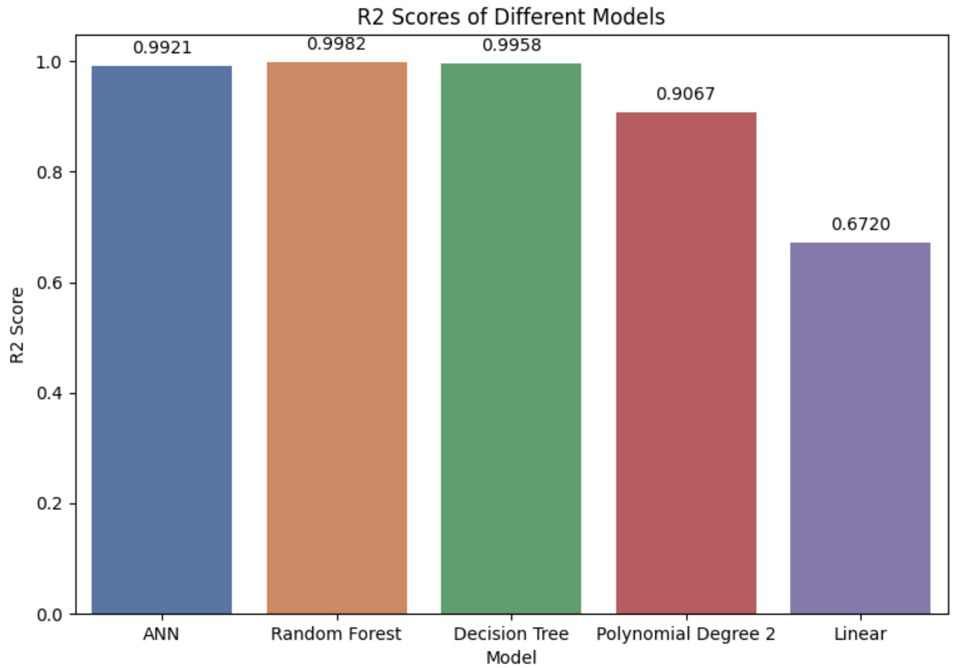


Figure 1, R2 score of different models

Figure 2 shows the graph of actual vs predicted price. Graph illustrates the wellness of model fit.

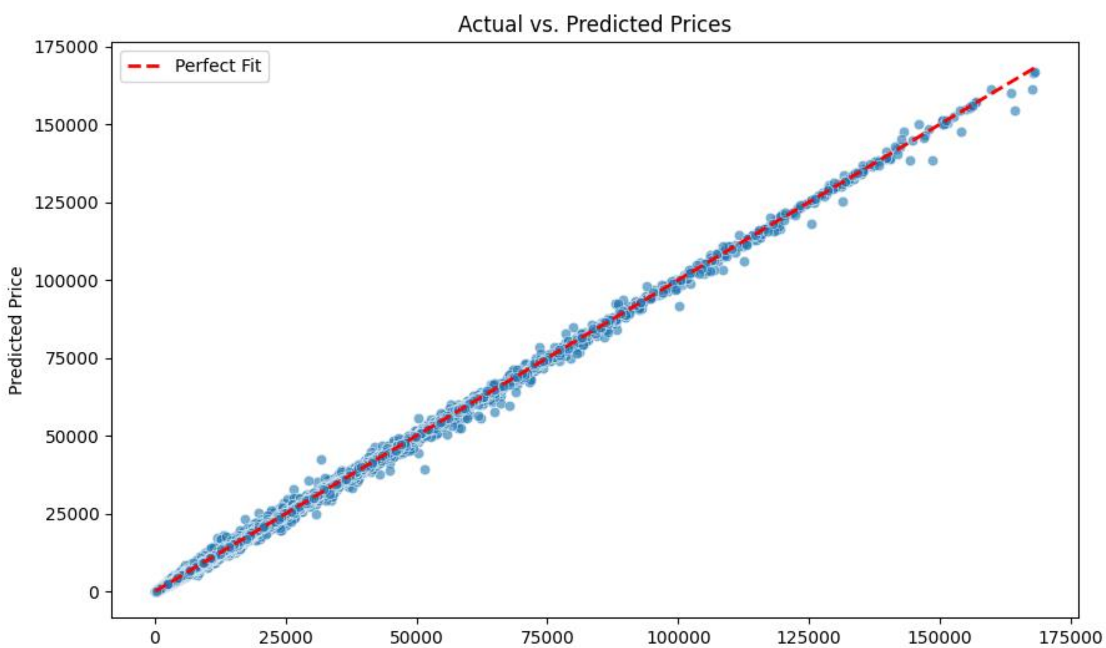


Figure 2, Random Forest, Actual Price vs. Predicted Price

Figure 3 shows the residuals plot, where it can be seen that residuals are randomly scattered around the horizontal axis without forming any pattern. This shows the model has captured all the underlying patterns in the data.

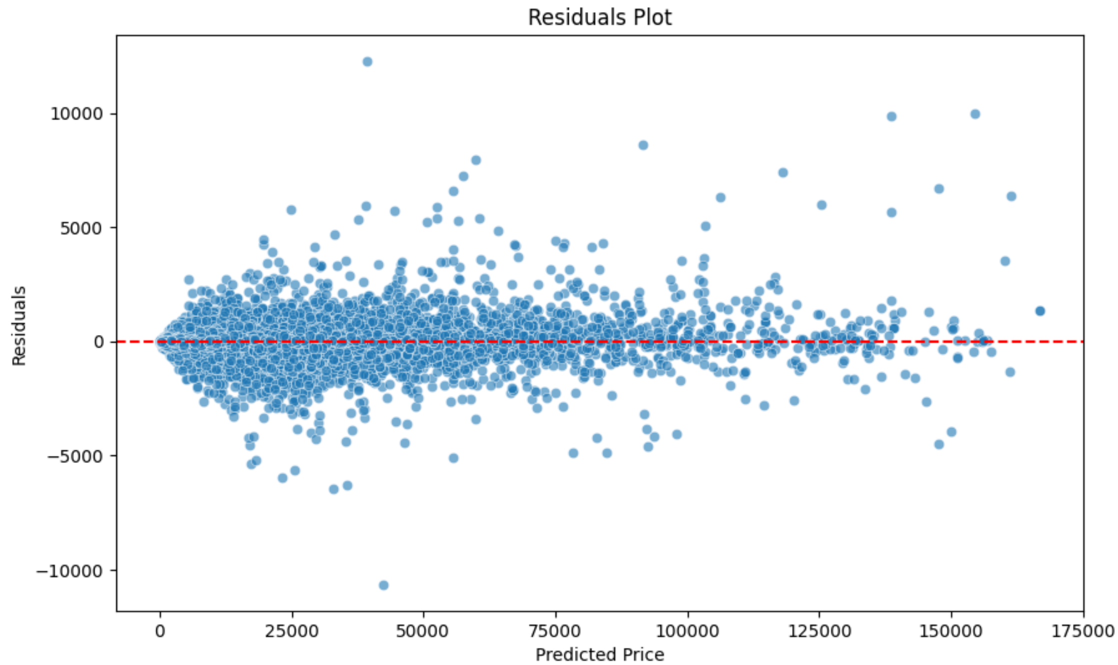


Figure 3, Random Forest, Residuals Plot

1.3. Unsupervised Learning

For accuracy evaluation of model, Davies-Bouldin (DB) Index and Silhouette Coefficient are used. A model is said to be robust if its Davies-Bouldin Index is small and Silhouette Coefficient (S) (Bonnin, n.d.) is close to +1.

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j} R_{ij}$$

$$s = \frac{b-a}{\max(a,b)}$$

1.3.1. K-Means Clustering

1.3.2. Optimal Number of Clusters (k)

Optimal number of clusters is determined using Elbow method & Silhouette coefficient. K is selected where minimum inertia and maximum Silhouette is obtained as shown in graphs below. This Point balances the trade-off between having too many clusters and too few clusters. Since the dataset is labelled so deliberately ignoring the price column to make dataset unlabeled.

Engine Size & Year of Manufacture (k = 3)

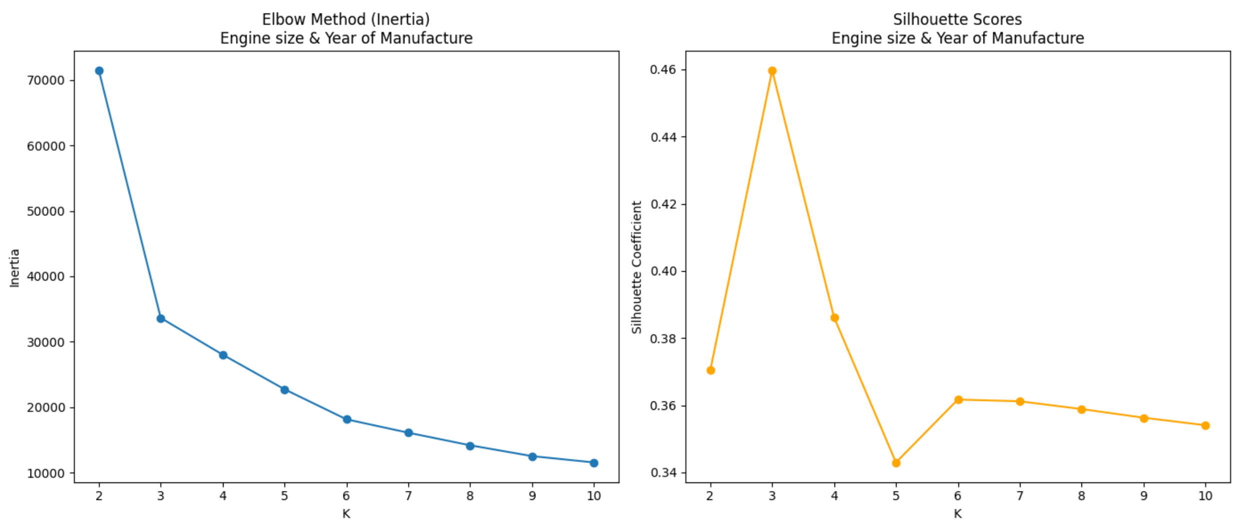


Figure 4: Elbow Method & Silhouette Score of Engine Size & Year of Manufacture

Year of Manufacturer & Mileage (k = 2)

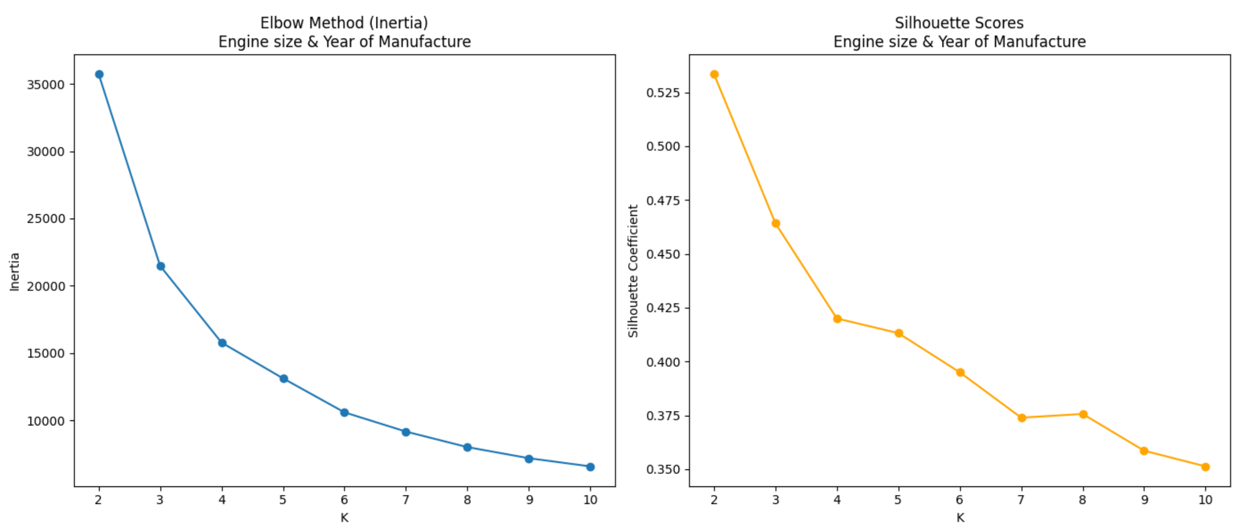


Figure 5: Elbow Method & Silhouette Score of Year of Manufacture & Mileage

Mileage & Engine Size (k = 3)

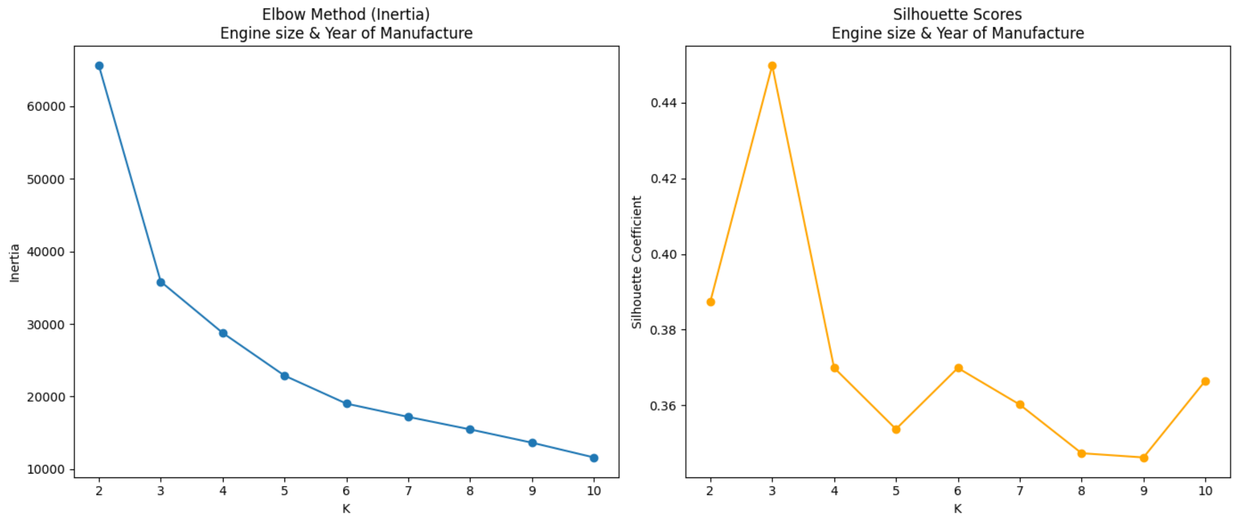


Figure 6: Elbow Method & Silhouette Score of Engine Size & Mileage

Engine Size, Year of Manufacture & Engine Size (k=3)

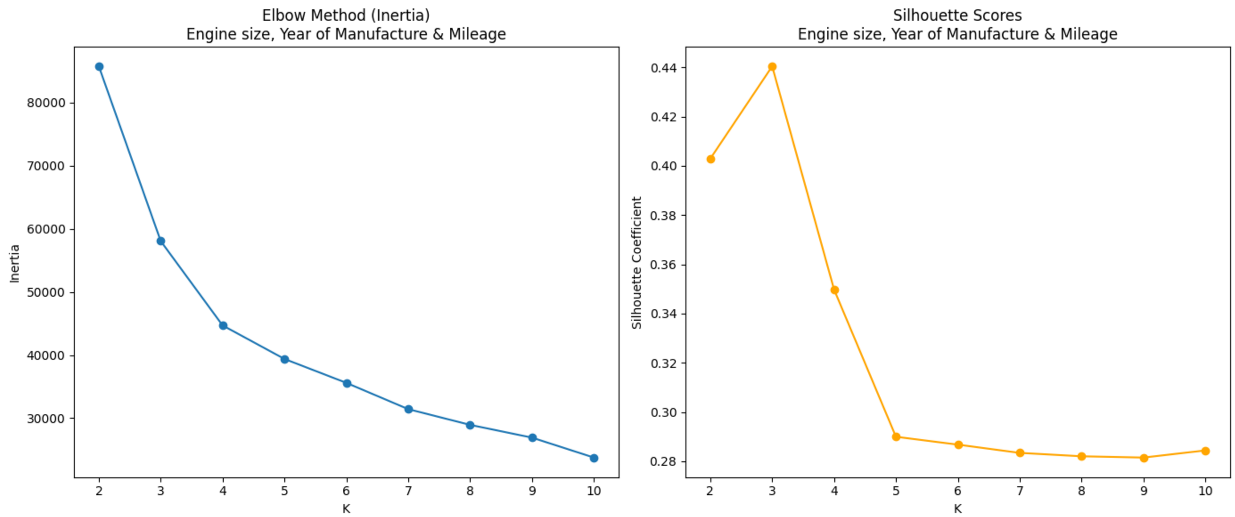


Figure 7: Elbow Method & Silhouette Score of Engine Size, Year of Manufacture, Mileage

Combination	Davies-Bouldin Index	Silhouette Coefficient
Engine size & Year of Manufacturer	0.752500164	0.459695553
Year of Manufacturer & Mileage	0.658826523	0.533497893
Mileage & Engine size	0.776741057	0.449839987
Engine Size, Mileage & Year of Manufacture	0.805472041	0.440399475

Combination of Year of Manufacturer & Mileage gives the best result as its DB score is lowest and Silhouette Coefficient is close to +1.

1.3.3. Gaussian Mixture Model

	K- Means		Gaussian Mixture	
Combination	DB Index	Silhouette Coefficient	DB Index	Silhouette Coefficient
Engine size & Year of Manufacturer	0.752500164	0.459695553	0.753737654	0.459851792
Year of Manufacturer & Mileage	0.658826523	0.533497893	0.708538567	0.474691579
Mileage & Engine size	0.776741057	0.449839987	0.784722482	0.446917018
Engine Size, Mileage & Year of Manufacture	0.805472041	0.440399475	0.877884627	0.390378180

For same number of clusters, DB index of K-Means clustering is lowest for all combination. Therefore, **K-Means produce best clustering**.

1.4. Conclusion

For labelled data set, supervised learning is usually applied for prediction purpose. Several Supervised Machine Learning algorithms were applied and appropriate evaluation metrics were used to evaluate the model performance. **Random Forest Regression model** turns out to be the best predictor with R^2 value of 0.9982. Clustering was also performed on the same dataset using different unsupervised algorithms by ignoring the price column and considering dataset as unlabeled. **K-Means** provides the best clustering in this case.

2. Flower Species Identification Using Image Recognition

2.1. Introduction

In this work, a Convolution Neural Network (CNN) has been developed to recognize the species of flowers from the images. CNN model has been trained using flowers dataset downloaded from the TensorFlow website. Dataset contain 3670 RGB images of the following flower species:

- Daisy
- Dandelion
- Roses
- Sunflowers
- Tulips

2.2. Convolutional Neural Network (CNN)

CNN is considered as branch of Artificial Neural Network (ANN) that's basically inspired by biological nervous system in which several neurons work in collaboration to crack a problem. A typical CNN system is built up of four layers consisting of a convolutional layer that extract features from input data using feature map, a pooling layer that decreases the size of feature map to eliminate noise, a flattening layer that transform 2D features to 1D vector and a fully connected layers that combine features extracted from previous layers and produces prediction (FatihahSahidan, 2019).

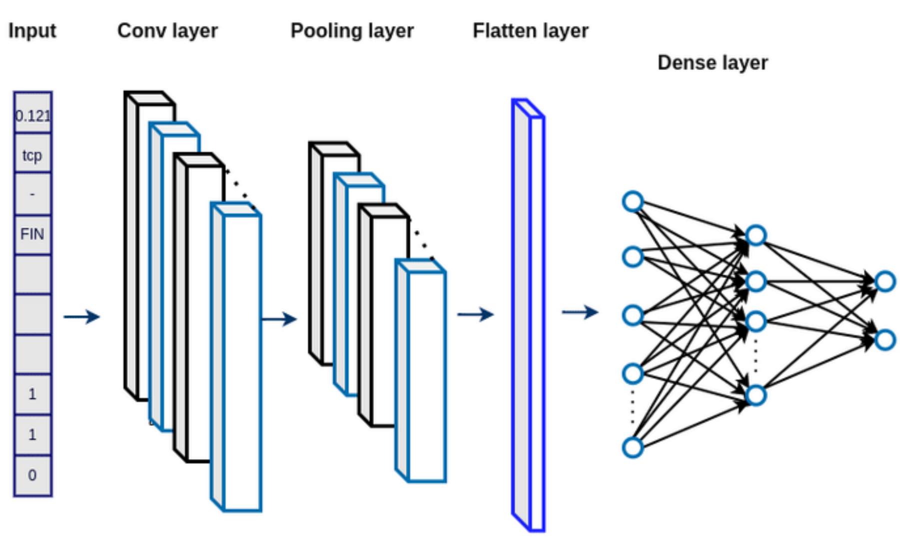


Figure 8: CNN Typical Layers (Meliboyev, n.d.)

2.3. Architecture of CNN Model

2.3.1. Convolutional Layer

Conv #	Filters	Kernel Size	Activation Function
Conv1	32	3 x 3	ReLU
Conv2	64	3 x 3	ReLU
Conv3	128	3 x 3	ReLU

2x2 Max Pooling Layer is used after each convolutional layer to reduce the spatial dimensions for computation ease.

2.3.2. Flatten Layer

To pass the extracted features into the fully connected dense layer, flatten layer is used that convert 3D feature maps to 1D vector.

2.3.3. Fully Connected Dense Layer

Layer	Neurons	Activation Function
First Dense Layer	64	ReLU
Output Layer	5	Softmax

Dropout rate of 20% is applied after first dense layer to prevent overfitting. Also 5 neurons in output layer is referred to the type of flower probability over five flower classes.

2.3.4. Compilation Stage

- Optimizer used was adam with default learning rate
- Batch size of 32 was used
- 20 epochs were used
- To guide the training process, a multi classification loss cross entropy was used

2.4. CNN Model Architecture Operation & Parameter Justification

Input shape of RGB pictures is (128, 128, 3) where 128 x 128 correspond to the pixel size and 3 correspond to three-color channels i-e Red, Green Blue.

CONV1: First convolutional layer applied 32 filters of 3 x 3 matrix to the input image. To perform convolutional operation each filter will slides over the input image capturing features such as edges or corners. ReLU introduces non-linearity allowing model to capture complex patterns and prevent vanishing gradient issue. This transforms the shape from (128, 128, 3) to (126, 126, 32).

POOL1: To reduce the dimensionality of feature map and lower computational load, maximum pooling is used. A 2 x 2 Matrix slides over transformed shaped and select maximum value from its 2 x 2 block. This layer reduces the parameters to (63, 63, 32).

CONV2: Second convolutional layer applied 64 filters of 3 x 3 matrix with ReLU activation function to detect further detailed features like textures. This layer reduces the shape to (61,61,64).

POOL2: Similar to POOL1, POOL2 reduces the feature map size by half i-e (30, 30, 64) by sliding 2 x 2 matrix and selecting maximum value in block.

CONV3: Third convolutional layer uses 128 filters of same matrix size and activation function. This layer operates on even more detailed features like color of flower. This layer transforms the shape to (28, 28, 128).

POOL3: This layer performs final reduction of feature map dimension bringing them to (14, 14, 128).

FLATTEN: Finally, 3D feature map of shape (14, 14, 128) is converted to 1D vector of size 25088 for feeding into the fully connected layer

FC1: Dense layer contains 64 neurons. This layer reduces the dimensionality from 25088 to 64, concentrating features into more manageable size. ReLU activation function is used to maintain non-linearity.

DROP1: To prevent the dependance on particular neurons and prevent overfitting, random 20% neurons are set to zero.

OUTPUT: Final layer with 5 neurons, corresponding to the 5 different flower species. To convert output into probability, softmax activation function is used.

2.5. Initial CNN Model Result

Above discussed architecture yield Training accuracy of 0.7768 and Validation accuracy of 0.7274. No result of overfitting is found as graph of both training and validation accuracy have minimum distance.

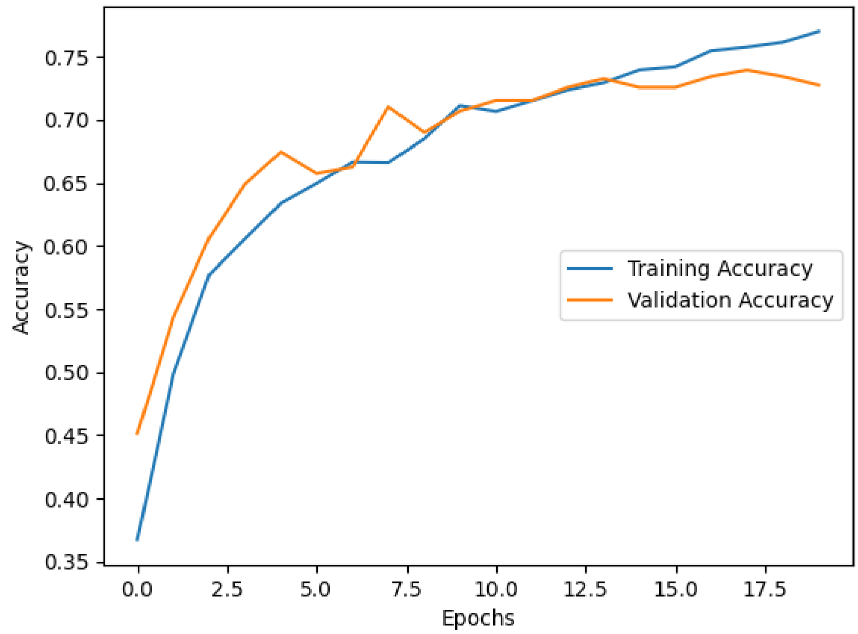


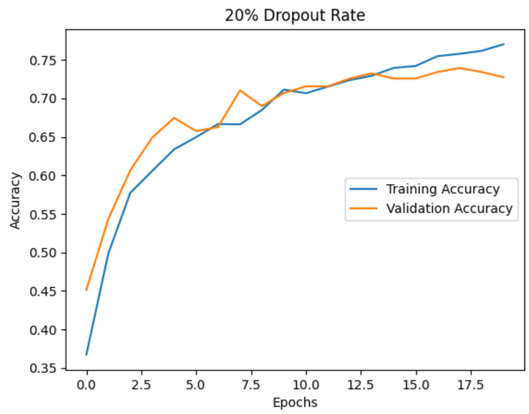
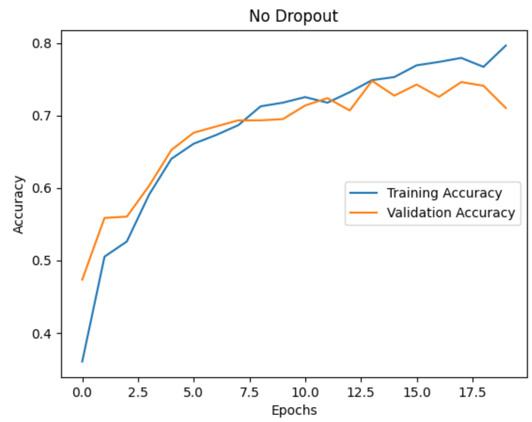
Figure 9: Initial Model Accuracy Graph

2.6. Regularization Techniques & their Impact

For initial start only one regularization technique is used i-e dropout rate.

Dropout: Dropout rate of 20% is used. This randomly ignored half of the neurons during training process so that model doesn't rely on specific neurons and learned all generalized features.

Effect of Dropout on Accuracy

Dropout 20%	No Dropout
Training Accuracy = 0.7768 Validation Accuracy = 0.7274  	Training Accuracy = 0.8034 Validation Accuracy = 0.7104 

When no dropout was used, slight overfitting occur in the model as training accuracy increases from 0.7768 to 0.8034 and validation accuracy decreases from 0.7274 to 0.7104. Model learned detailed patters from training data causes better accuracy for training data but when validation set was used, model struggled as depicted by slight decrease in validation accuracy.

2.6. Hyperparameter Tuning

Hyperparameter tunning was done using some base set parameters due to less computational resources. Throughout tunning same number of convolutional layers and their filters, fully connected layers, activation function, epochs, optimizer and loss measurement metrics were used.

2.6.1. Varying units in Fully Connected Layer

Neurons in fully connected dense layer was altered starting from 64 neurons and then doubled up to 512 neurons. Improvement in validation accuracy was observed. After 256 neurons validation accuracy slightly decreases. Neurons with highest validation accuracy was chosen for further hyperparameter tunning.

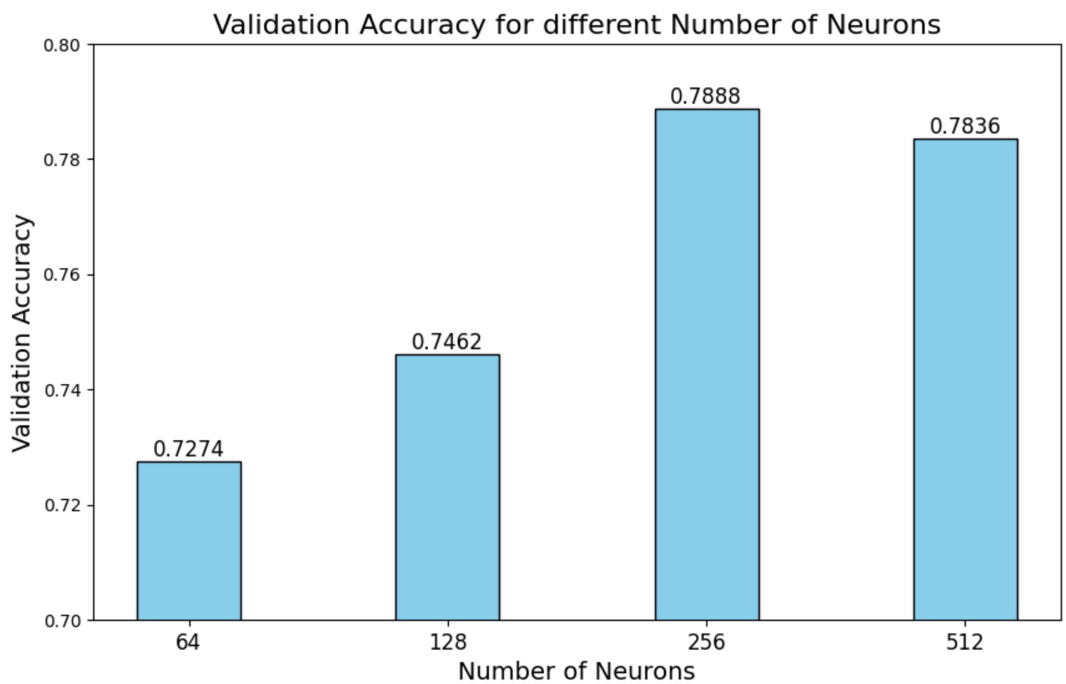


Figure 10: Hyperparameter Tunning (Neurons)

2.6.2. Varying Learning Rate

Learning rate of 0.0001 was checked for same number of epochs. Due to low learning rate convergence speed was low, at the end of 20 epochs validation accuracy of 0.7138 was obtained. Default learning rate was then used for further hyperparameter tunning.

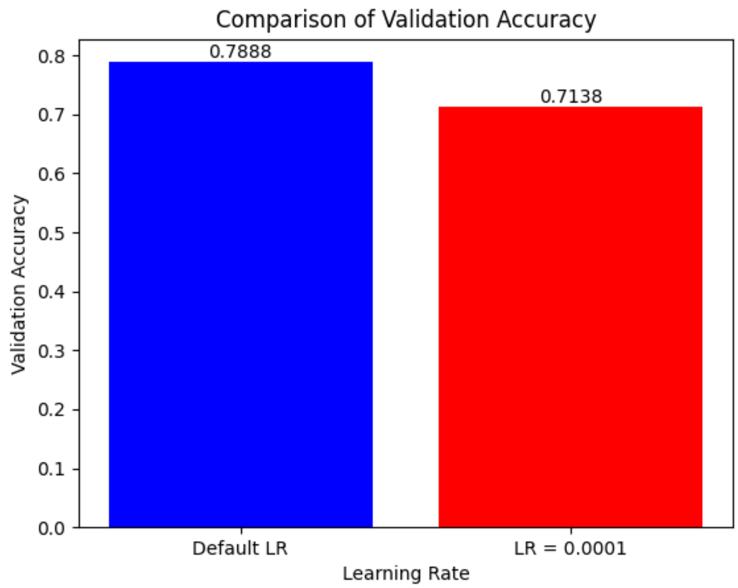


Figure 11: Hyperparameter Tunning (Learning Rate)

2.6.3. Batch Size

Batch size was altered from 32 to 128 but decrease in validation accuracy was observed. With large batch size, sometimes model generalizes poorly. Also, throughout iteration, same learning rate was used which might affect the training dynamics (Keskar, 2017).

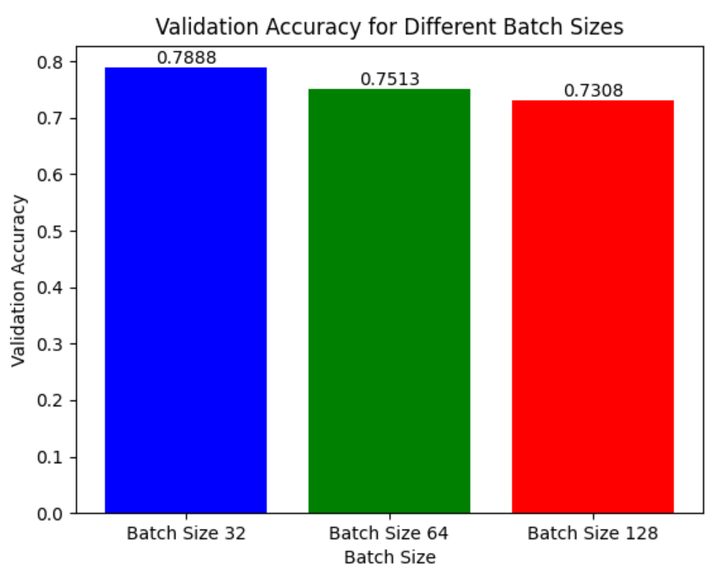


Figure 12: Hyperparameter Tunning (Batch Size)

2.6.4. Kernel size

No improvement in validation accuracy was found when Kernel size changed from 3 x 3 to 5 x 5.

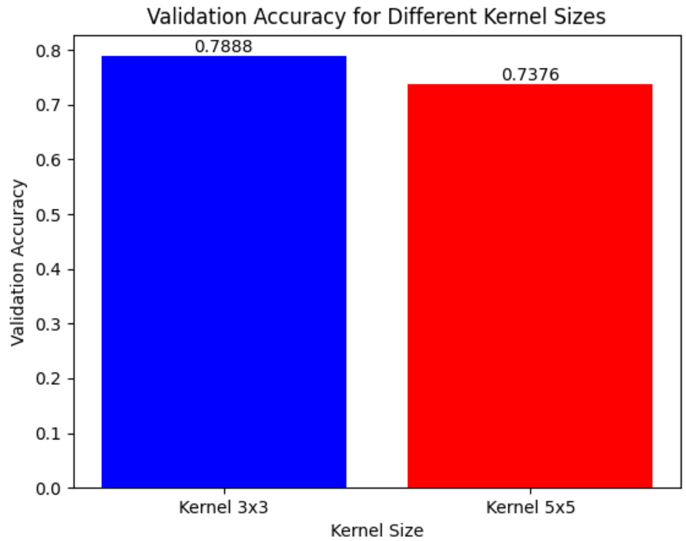


Figure 13: Hyperparameter Tunning (Kernel Size)

2.6.5. Batch Normalization

Batch normalization with momentum of 0.9 was used after every max pooling in convolutional layer. Validation accuracy doesn’t improve for same number iterations.

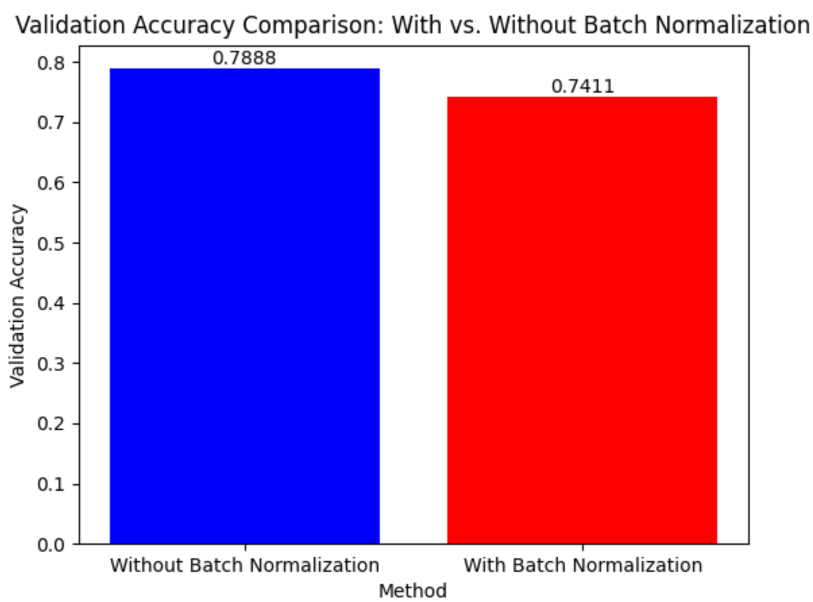


Figure 14: Hyperparameter Tunning (Batch Normalization)

2.6.6. Stride & Padding

Stride of 2 for filter to move 2 pixels at a time in both horizontal and vertical direction and padding = "same" to ensure same spatial dimension of output feature as the input feature map was used. In this case, still no improve in validation accuracy was found.

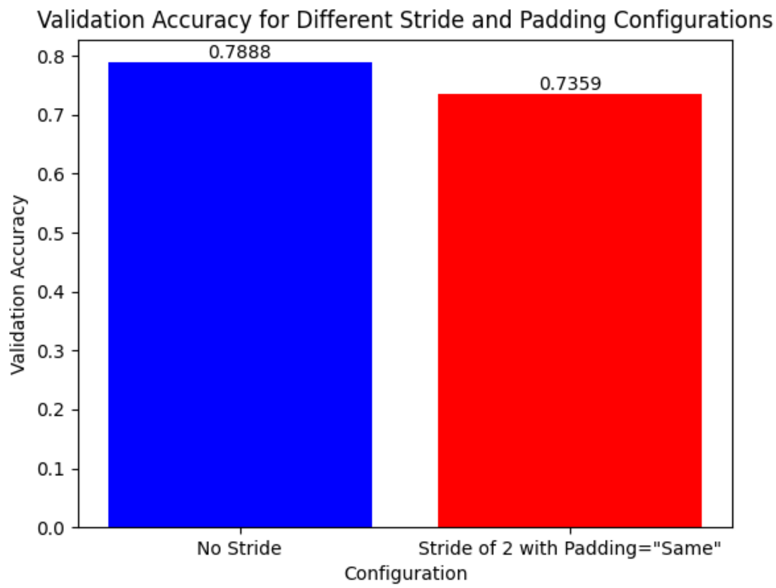


Figure 15: Hyperparameter Tunning (Stride & Padding)

2.6.7. Effect of Hyperparameter on Model

Number of Neurons in fully connected layer, learning rate, kernel size and strides have the strongest effect on the performance of model as variation in these parameters drastically varies the validation accuracy as depicted in figures above.

2.7. Final Model Architecture

2.7.1. Convolutional Layer

Conv #	Filters	Kernel Size	Activation Function
Conv1	32	3 x 3	ReLU
Conv2	64	3 x 3	ReLU
Conv3	128	3 x 3	ReLU

2x2 Max Pooling Layer after each convolutional layer is used to reduce the spatial dimensions for computation ease.

2.7.2. Flatten Layer

To pass the extracted features into the fully connected dense layer, flatten layer is used that convert 3D feature maps to 1D vector.

2.7.3. Fully Connected Dense Layer

Layer	Neurons	Activation Function
First Dense Layer	256	ReLU
Output Layer	5	Softmax

Dropout rate of 20% is applied after first dense layer to prevent overfitting. Also 5 neurons in output layer is referred to the type of flower probability over five flower classes.

2.7.4. Compilation Stage

- Optimizer used was Adam with default learning rate
- Batch size of 32 was used
- 20 epochs were used
- To guide the training process, a multi classification loss cross entropy was used

2.7.5. Results of Final Model

Both training and validation accuracy increases rapidly during the initial epochs with both curves closely tracking each other. Slight dip in the middle for validation accuracy was observed which recover and progress further. Training accuracy continue to increase and ended at 0.82 while validation accuracy also increases slightly but remain lower than training accuracy, reaching 0.78.

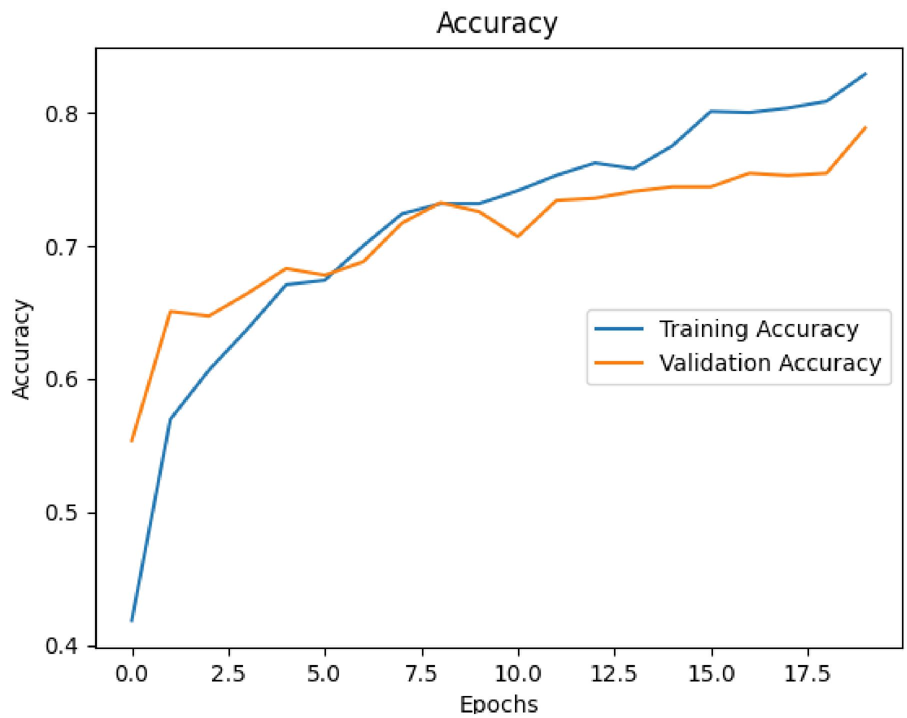


Figure 16: Final Model Accuracy Graph

2.8. Overfitting

In the middle, there's a slight hike where loss drastically increases. This hike suggest that model might starting to overfit the data and that model is beginning to memorize training examples rather than learning general patterns. The gap between training and validation accuracy in later stages also suggest overfitting but relatively mild. Overall, there's small degree of overfitting as evident by the loss graph.

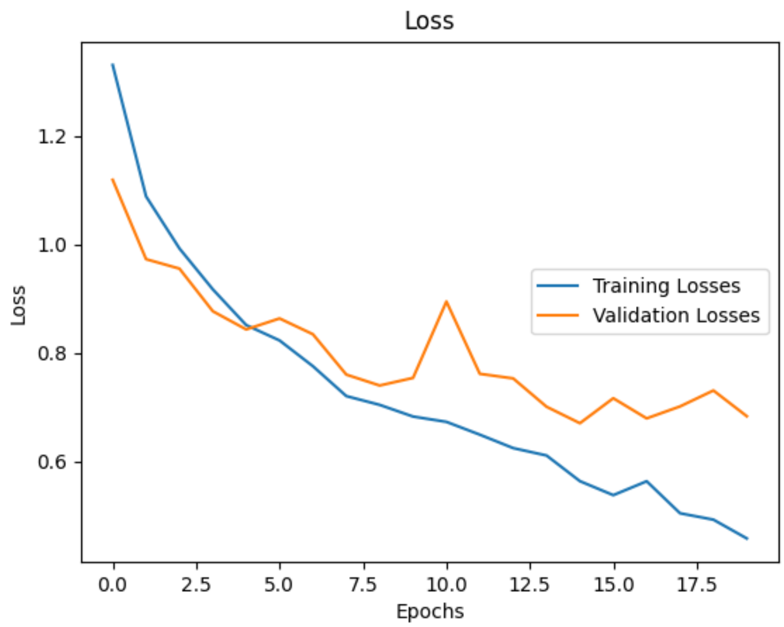


Figure 17: Final Model Loss Graph

3.1. Literature Review on Ethical Applications of AI

This work revolves around the examination of three journal articles that talk about the ethical application of AI specifically focusing on the topic of Socially Responsible AI. This topic is an approach that focuses on the societal impact of Artificial Intelligence. It makes sure that AI system contribute positively toward the society and outcast harmful effects.

3.1.1. Article 1: AI for Social Good: A Vision & a Strategy (Evans, 2024)

Aims

This article talks about the strategies that could address the social issues and positively impact it by utilizing AI. It suggests on establishing protocols for implication of AI towards societal benefit.

Key Conclusion

- Notable social wellness of AI is its utilization in the field of medicine where technology can be used to improve diagnosis, treatments and predict any disease outbreak.
- Environmental issues such as climate change or disaster response has potential AI applications. One such example of AI application is Google's project sunroof that estimate the solar energy potential.
- Paper also talks about the access of AI to marginalized group and creating equitable AI system.

3.1.2. Article 2: Artificial Intelligence & Future of Work: How AI Impacts Employment & Labor Markets (Cazzaniga, 2024)

Aims

This articles debate about the impact of AI and automation on labor market and employment. Article also predicts the potential effects on job displacement and wave inequality casted by AI.

Key Conclusion

- Increased demand of high-skilled workers and reduced opportunities for low-skilled workers as their tasks are easily automated may worsen wage inequality.
- AI can lead to the alteration of dynamics of employment patterns like conventional shifts type jobs.

- Few case studies and economic model on empirical data is also used by author to analyze preceding points.

3.1.3. Article 3: Ensuring Fairness & Accountability in Machine Learning System (Pu Chen, 2023)

Aims

Article aims on investigating various challenges related to fairness and biasness in machine learning systems and their impact on different demographic groups. Article also study to ensure machine learning systems are held accountable for their decisions.

Key Conclusion

- Article talk about various metrics that are used for the assessment of machine learning system such as demographic parity that ensure equal favorable outcome across different groups, equalized odds that ensure same rate of false positive and false negative across groups and calibration to ensure accurate prediction across all groups.
- Pre-processing techniques, in-processing techniques and post-processing techniques are discussed to address to bias in machine learning models
- Through Improved model interpretability, periodic audit and transparency reports, machine learning system can hold accountable

3.1.4. Successes in Socially Responsible AI

- Artificial Intelligence has helped doctors in diagnosing of diseases quickly and accurately by analyzing patients' data and medical images. AI for example can detect early signs of cancer in scan that might missed by human eye. Moreover, by analyzing genetic information AI can predict how certain drugs can affect the patient, this information can be used to customized treatments for better results.
- AI is advancing in protection of environment by tracking deforestation. AI powered cameras and sensors can analyze potential endangered species thus protecting the wildlife population.
- Advancement in AI tools such like Pymetrics (Hunkenschroer, 2022) and HireVue reduce bias by analyzing recruiters' data based on skills rather than demographic factor to make hiring process fair.

3.1.5. Challenges in Ethics of AI

- With the extent of using AI in industries, some careers are at risk of being displaced. For example, the application of robots in production processes indicates that manufacturers will not require human beings to operate these machines so some will lose their jobs.
- It is crucial to understand that traditionally, AI systems are mainly created to identify, collect and analyze huge volumes of personal data. This can lead to privacy concerns every time data is used wrongly or poorly guarded. For example, the application of artificial intelligence in the surveillance will track individuals' behaviors without their knowledge.
- One important issue is that the machine learning algorithms, even if they do not imply any prejudice, may codify and reproduce prejudice and bias we have in social data. For example, if a system that learns from data comes across racist or sexist information, it will be partial to making racist or sexist decisions.

3.1.6. Suggestion to overcome Gaps

- Develop programs that can help the affected employees to undergo training. This might include offering courses in growing areas that are yet to be popular, vocational counselling and job networking over promising areas of employment.
- Strengthen and enact better privacy regulation that make organizations accountable for protection of the consumer's information and for limiting its use. This could include limitations regarding data collection, storage, and sharing. Invest in encryption and anonymization for better data security.
- Improve approaches in Artificial Intelligence to help eliminate bias. Conduct audits of the AI systems regularly to ensure they are fair and good for use. This helps in ensuring that bias or any other issue that may be likely to arise over time does not strain these AI systems to be unjust.

References

- Anon., n.d. *Critical Value Calculator.* [Online]
  Available at: https://www.criticalvaluecalculator.com/coefficient-of-determination-calculator
  [Accessed 08 2024].
- Bernico, M., n.d. *O'REILLY.* [Online]
  Available at: https://www.oreilly.com/library/view/deep-learning-quick/9781788837996/802030f3-98cb-4e44-b597-a785fae9bd67.xhtml
  [Accessed 08 2024].
- Bonnin, R., n.d. *O'REILLY.* [Online]
  Available at: https://www.oreilly.com/library/view/machine-learning-for/9781786469878/5ebb37f2-b0b3-4414-9f28-1d10854e64ab.xhtml
  [Accessed 08 2024].
- Cazzaniga, M., 2024. Artificial Intelligence and the Future of Work.
- Evans, W. D., 2024. AI for Social Good: A Vision & a Strategy. *Businesses.*
- FatihahSahidan, N., 2019. Flower and leaf recognition for plant identification using. *Indonesian Journal of Electrical Engineering and Computer Science.*
- Hunkenschroer, A. L., 2022. Ethics of AI-Enabled Recruiting and Selection: A Review and Research. *Journal of Business Ethics.*
- Keskar, N. S., 2017. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima.
- Meliboyev, A., n.d. *Research Gate.* [Online]
  Available at: https://www.researchgate.net/figure/The-single-CNN-architecture-The-network-includes-following-layers-input-layer-one_fig1_358477638
  [Accessed 08 2024].
- Pu Chen, L. W. a. L. W., 2023. Ensuring Fairness & Accountability in Machine Learning System. *Applied Sciences.*