Analyzing Fashion-MNIST dataset “Fashion Forward”

W200 – Project 2, Nov 2023

Trevor Lang, Ryan Powers, Carmen Liang

https://github.com/UC-Berkeley-I-School/Project2_Powers_Lang_Liang

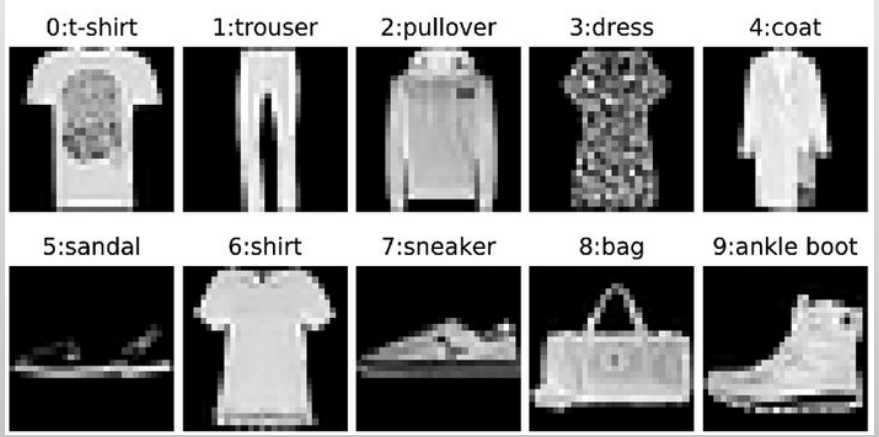
Introduction: “Decoding the Dress Code with Machine Learning”

Project Overview:

- Develop and compare machine learning models for classifying images from the Fashion-MNIST dataset.
- Compare performance of Random Forest (ensemble learning) and Convolutional Neural Network(CNN) algorithms.

Dataset:

- Fashion-MNIST dataset



Overall Question:

- How does the distribution of images across different classes in the Fashion-MNIST dataset affect the performance of Random Forest and Convolutional Neural Network(CNN) algorithms?

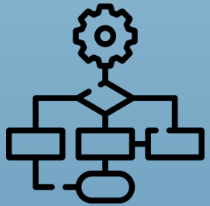
Steps we took to Analyze the Data: “The Fashionable Challenge”

1. Data Loading and Preprocessing:

The Fashion-MNIST dataset was loaded from the data repository

The data was then divided into training, testing, and validation sets

Reshaped and normalized the images and labels to optimize learning.



2. Model Implementation:

Random Forest utilized scikit-learn library

CNN utilized Keras/TensorFlow.

3. Model Training and Evaluation:

Both models were trained and evaluated on the training and validation sets.

We calculated various performance metrics (accuracy, precision, recall, F1 score, and AUC).

4. Comparison of Models:

How the distribution of images across different classes in the dataset affected the performance of the two algorithms

Assumptions we made:

1. **Data Quality**: Fashion-MNIST dataset was high quality with no missing or erroneous data.
2. **Model Suitability**: Random Forest and CNN were suitable models for the image classification task.
3. **Data Distribution**: Distribution of images across different classes in the Fashion-MNIST dataset was representative of real-world scenarios**.**

Expected Outcomes:

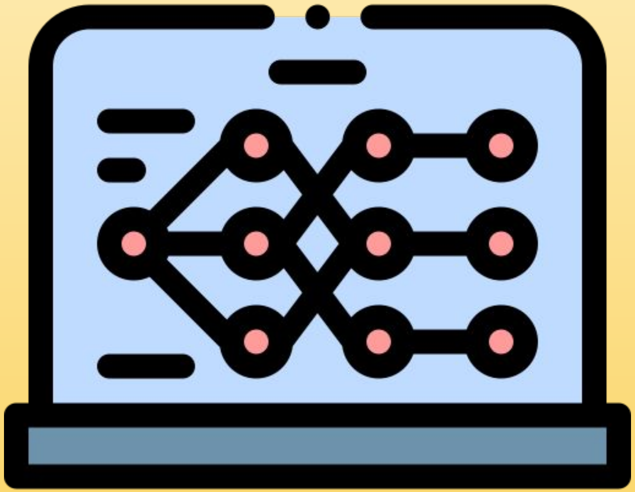
- Gain insights into factors affecting image classification performance
- Compare effectiveness of Random Forest and CNN on Fashion-MNIST
- Identify strengths and weaknesses of each algorithm
- Contribute to understanding of image classification in machine learning

Data Preparation: “ Behind the scenes”

Welcome to the backstage of our project! Here, we've prepared the Fashion-MNIST dataset for our models. We've split the data into training, testing, and validation sets, ensuring a diverse range of images for each phase. We've reshaped the images to fit the requirements of our models and normalized the pixel values. Finally, we've converted the class vector labels into binary class matrices using one-hot encoding. This meticulous preparation is the foundation of our machine-learning journey!

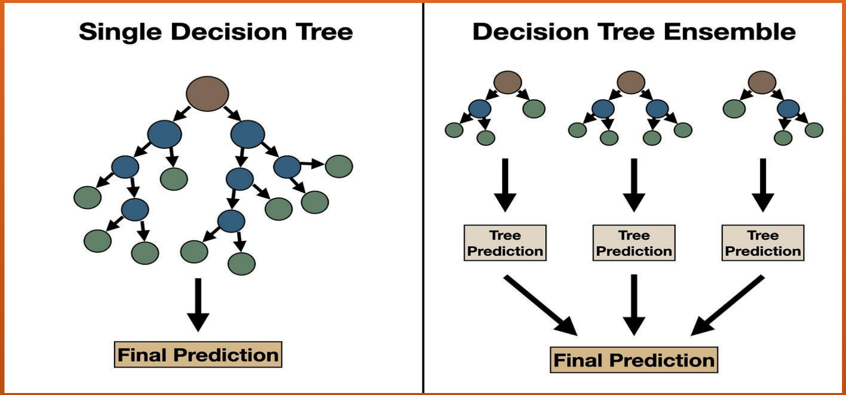
CNN Model Architecture: “ The Visionary”

Let's dive deep into the architecture of our Convolutional Neural Network (CNN) model. It's a complex structure with multiple layers, including two 2D convolutional layers, two dropout layers to reduce overfitting, a flattened layer to reshape the output volume for dense layers, a 256-unit Dense layer, and a final Dense softmax layer for the 10 classes. We've compiled the model with categorical cross-entropy loss, Nadam optimizer, and accuracy, precision, and recall metrics.



What is a Random Forest Model: “ The Council of Fashion Experts”

Next, we have our Random Forest (RF) model. This model is a collection of decision trees, each trained on a random subset of the features and training data. The RF model learns the underlying patterns in the data and uses these patterns to make predictions on new, unseen data. We've evaluated the model using various metrics such as accuracy, precision, recall, and F1 score.

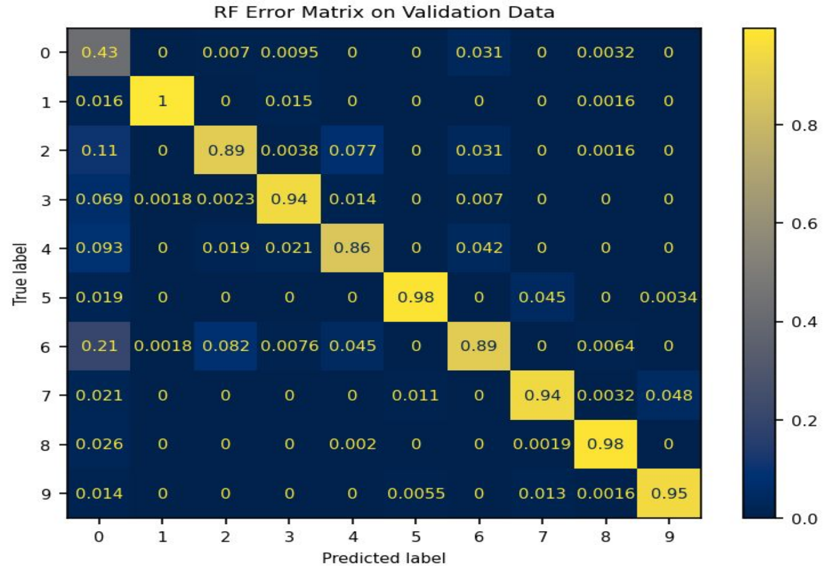


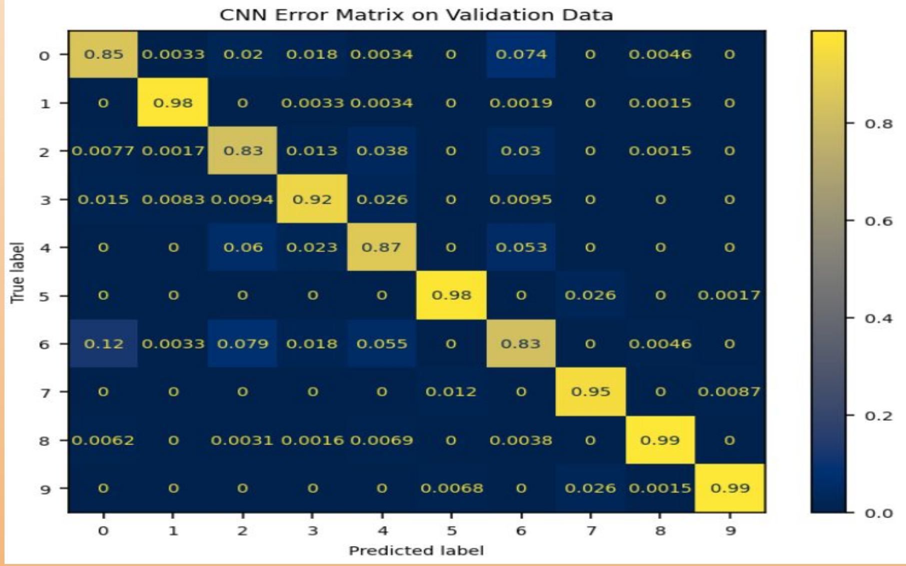
Why we chose RF: In real-world scenarios, machine learning engineers and data scientists frequently utilize random forests because they are highly precise and modern computer systems are capable of handling large datasets that were previously unmanageable.

Key Figures/Plots/Charts and Graphs

1. **Confusion Matrix**: This table helps us understand how well the model performed for each class, and where it may have had some difficulty.
2. **Histogram of Pixel Values**: Tells us about the distribution of pixel values in the images used to train the model. It helps us understand what the images look like.
3. **Error/Confusion Matrix**: A confusion matrix is like a tally board, counting how often your model got things right (true positives) and wrong (false positives, negatives). It helps you see where it stumbles and how to improve its accuracy.
4. **Random Sample of Images with Predicted Labels**: This plot showed that the model could classify most images correctly but struggled with some ambiguous images.
5. **CNN Plots**: These graphs show the model's performance during training, including how the accuracy changes over time. They help us understand how the model learns and improves.

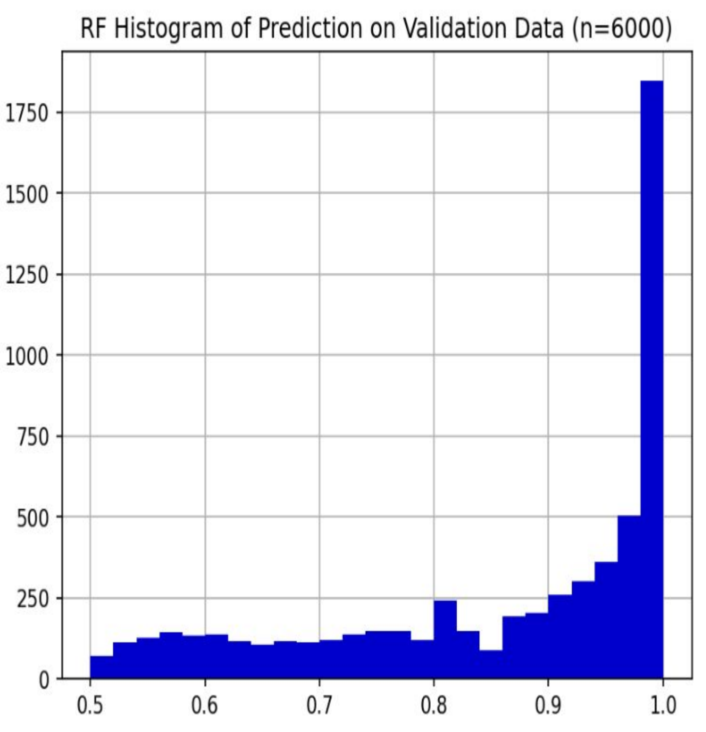
Key Figures – Confusion Matrix



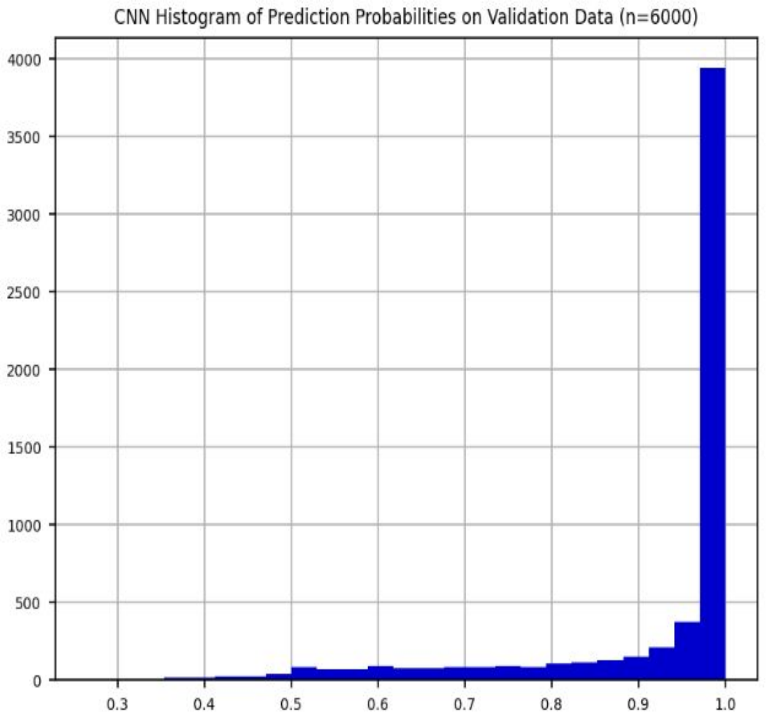


The confusion matrix for the Random Forest model showed that the model performed well for most classes but struggle to identify classes 0 (“top”), 2 (“pullover”), 4 (“coat”), and 6 (“shirt”)

Key Figures – Histogram of Pixel Values



The histograms of the pixel values showed that most pixel values were concentrated around 0 and 1, suggesting that the images were primarily black and white with few shades of gray.



Key Figures– Random Sample of Images with Predicted Labels

This plot showed that the model could classify most images correctly but struggled with some ambiguous images.

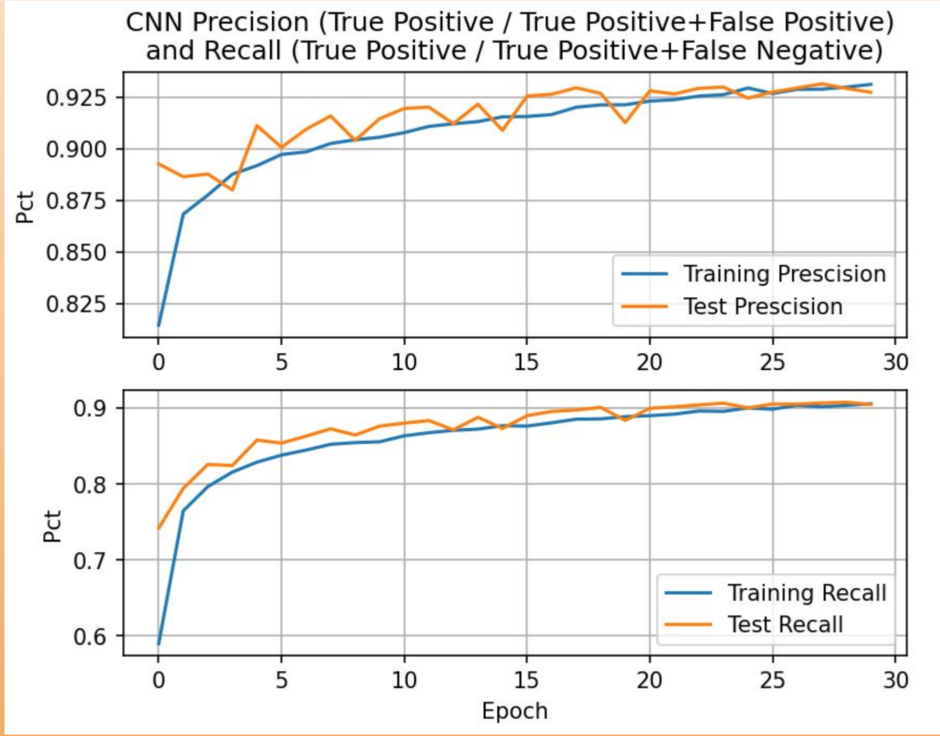
RF Accuracy: 0.82

RF Precision (True Positive / True Positive+False Positive): 0.9309413547755834

RF Recall (True Positive / True Positive+False Negative): 0.8003333333333333

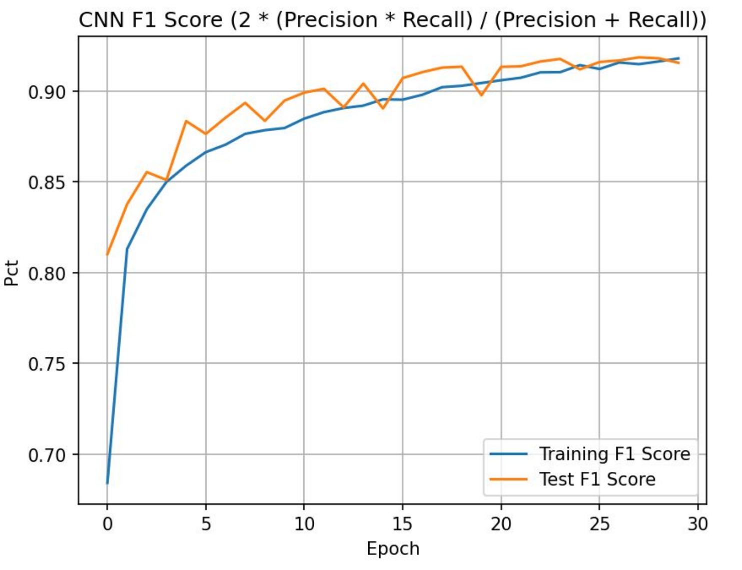
RF F1 Score (2 * Precision*Recall / Precision+Recall): 0.8607107846291333

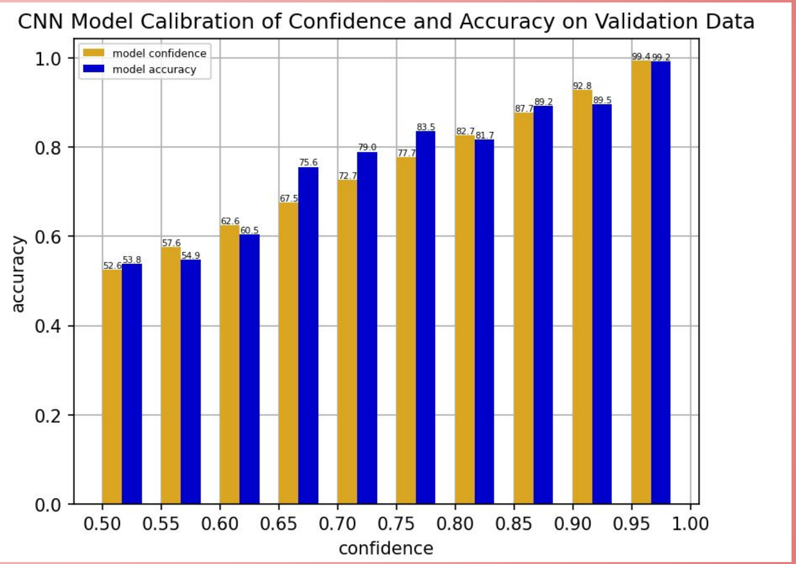
Key Figures – CNN Plots



The CNN plots during and post-training showed the model's performance over time and its final accuracy of over 92.5% on unseen data.

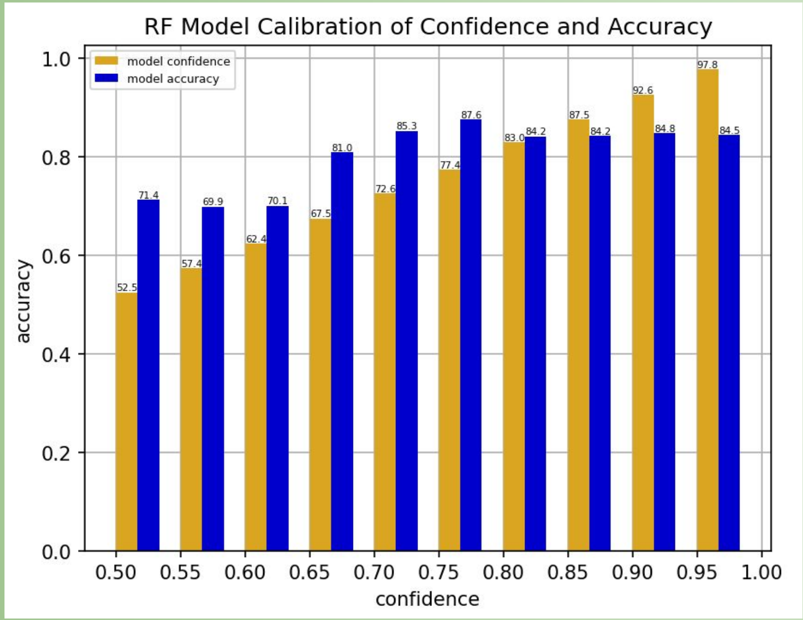
The CNN Model: Performance Plots

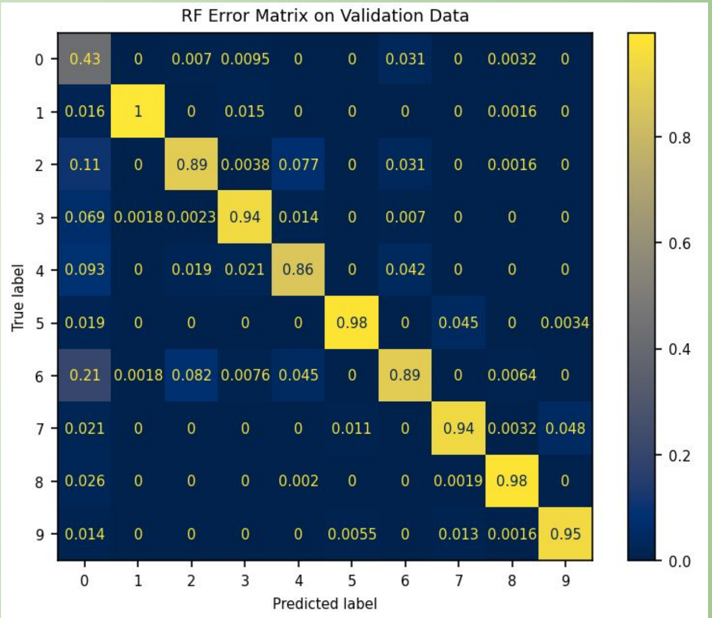




Let's embark on a visual journey through the training of our CNN model. During training, we've monitored the loss, accuracy, precision, recall, and F1 score. The plots show the values increasing with each epoch, indicating good model learning. We've also evaluated the model's calibration using the validation dataset. The plots show that the average confidence and accuracy values are reasonably close for each confidence interval, indicating a well-calibrated model. It's a fascinating visual representation of our model's learning journey!

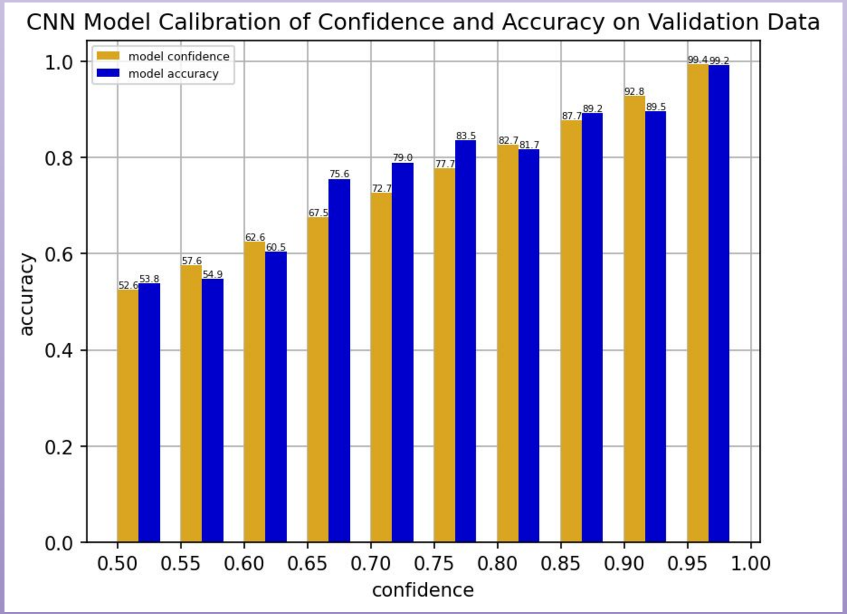
The RF Model: Performance Plots

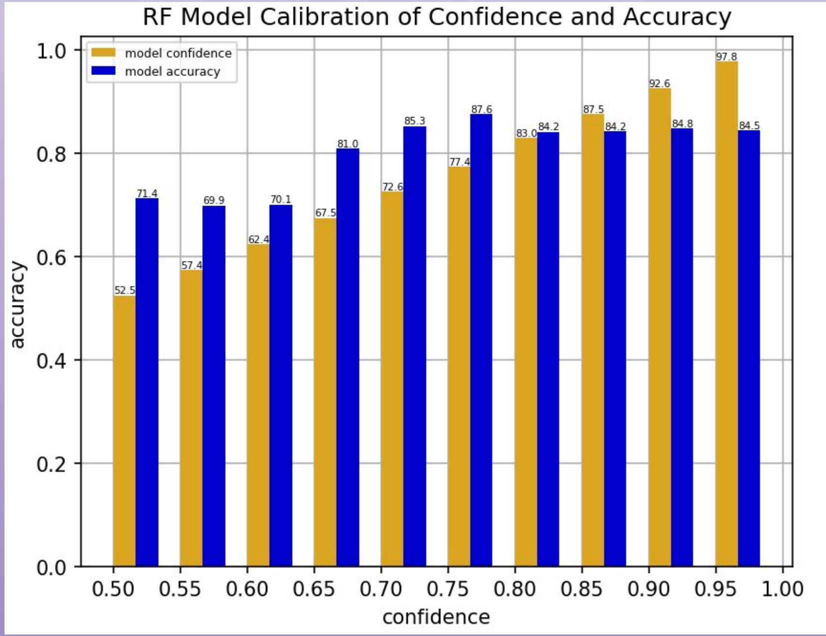




Now, let's visually stroll through the performance of our RF model. After the training cycle, we computed the overall values on the validation set. The plots show the average confidence and accuracy values, which are farther from each confidence interval than expected. We've also evaluated the model's performance using a confusion matrix. The matrix shows that the model has a more difficult time identifying certain classes. It's a tale of a forest's learning journey, with its twists and turns!

CNN VS RF: “ Who will be Crowned Data Sciences next top algorithm”





Here's where the excitement builds up! We've compared the performance of the CNN and RF algorithms on the Fashion-MNIST dataset. While both models performed admirably, the CNN model outperformed the RF model in accuracy. However, the choice between the two depends on the project’s specific requirements and the dataset's characteristics. It's a thrilling battle of algorithms, each with its strengths and weaknesses!

Model Comparison:

Feature	***Random Forest***	***Convolutional Neural Network (CNN)***
Model Type	Ensemble Learning	Deep Learning
Image Processing	No	Yes
Feature Selection	Built-in	Requires Manual Feature Engineering
Hyperparameter Tuning	Less Complex	More Complex
Interpretability	High	Low
Overfitting	Less Prone	More Prone
Performance on Fashion-MNIST	High Accuracy	Highest Accuracy
Real-World Pros	Interpretable results, good for small datasets, robust to outliers	State-of-the-art performance, can handle large and complex datasets
Real-World Cons	May not be as accurate as CNNs, performance can depend on feature selection, limited to classification tasks	Requires significant computational resources, black-box nature makes debugging difficult, can be sensitive to noise
Fashion-MNIST Pros	Accurate enough for most practical applications, interpretable results can help understand feature importance	Achieves state-of-the-art accuracy, can potentially be used for other image processing tasks in the fashion domain
Fashion-MNIST Cons	May be computationally expensive for large datasets, feature selection can be challenging, interpretability may be limited for complex models	May be overkill for relatively simple tasks like Fashion-MNIST, black-box nature may limit trust in its predictions

Conclusion: “ The Grand Finale in time for Winter Session”

Based on our analysis, it was evident that the way images were distributed across various classes in the Fashion-MNIST dataset had an impact on the performance of both the Random Forest and CNN algorithms. Nevertheless, it was observed that the CNN model performed better in handling the distribution and resulted in higher accuracy as compared to the Random Forest model. This indicates that CNNs could be a more appropriate choice for image classification tasks, particularly when dealing with complex and diverse images.

CNN model achieved a final test accuracy of over 92.5%, and the RF model achieved an accuracy of approximately 82%. The journey was filled with learning, challenges, and exciting discoveries. We've referenced several resources during our project, which you can explore for further reading.

Thank you for joining us on this exciting journey into machine learning!"

References:

1. http://yann.lecun.com/exdb/mnist/
2. https://github.com/zalandoresearch/fashion-mnist
3. https://developers.google.com/machine-learning/crash-course/ml-intro
4. https://developers.google.com/machine-learning/glossary
5. https://www.tensorflow.org/api_docs/python/tf/
6. https://scikit-learn.org/stable/modules/classes.html
7. https://www.flaticon.com/free-icon/neural-network_6310976
8. https://towardsdatascience.com/10-decision-trees-are-better-than-1-719406680564