Ninjacart - CV Classification Business Case

- ❖ Topic: Classifying images as potato, onion, tomato or noise.
- ❖ Duration: 1 week

Why this case study?

From the company's perspective:

- Ninjacart is India's largest fresh produce supply chain company. They are pioneers in solving one of the toughest supply chain problems of the world by leveraging innovative technology.

- The biggest challenge Ninjacart faces is to supply fresh products from the farmers to the businesses within 12 hours. They need to ensure that the correct vegetables are selected for delivery.

- Ensuring a correct product delivery is an integral component of their automation process, a robust classifier that can distinguish between images of different types of vegetables, while also correctly labeling images that do not contain any one type of vegetable as noise becomes critical.

- By leveraging data science and Computer vision, Ninjacart can classify various vegetables and noises.

From the learner's perspective:

- Engaging with this case offers a practical understanding of how Computer vision helps in tackling real-world problems
- Computer Vision with different image manipulation algorithms can significantly fasten the classification problem with high accuracy.

- This exercise hones the participant's skills in data augmentation in images, dealing with pre-trained CNN models and their evaluation, implementing callbacks, and overcoming overfitting.

- Additionally, learners gain hands-on experience in tackling real-world problems, transforming raw data into actionable insights that can guide business strategies.

Dataset Explanation: Ninjacart_data

This dataset contains a folder **train,** which has a total of 3135 images, split into four folders as follows:

- Tomato : 789
- Potato : 898
- Onion : 849
- Indian market : 599

This dataset contains another folder **test** which has a total of 351 images, split into four folders

- Tomato : 106
- potato : 83
- onion : 81
- Indian market : 81

What is Expected?

Assuming you are a data scientist at NinjaCart, you are responsible for analyzing the dataset to develop a program that takes an image as input and recognizes the vegetable item(s).

Submission Process:

Upon concluding the case study...

- Importing the dataset and doing usual exploratory analysis steps like checking the structure & characteristics of the data
- Exploratory Data Analysis.
    - Plotting class distribution & Visualizing Image dimensions with their plots
    - Splitting the dataset into train, validation, and test set
- Creating model architecture and training
    - Defining the CNN Classifier model from scratch
    - Improve model using finetuning
    - Use of pretrained CNN models
    - Plotting the model training metrics and confusion matrix
- Testing your best model so far
    - Testing on the test set & Random image samples prediction
    - Summary & Insights
- Transform your Jupyter Notebook into a PDF (Utilize the Chrome browser's Print command for this).
- Adhere to the submission guidelines and upload the PDF on the specified platform.
- Note that once your work is submitted, there's no provision to edit or modify your submission.

General Guidelines:

This scenario mirrors real-world challenges and embodies the tasks data scientists frequently grapple with. Embrace this opportunity to dive deep and simulate a professional experience.

During the course of this study, it's possible to face hurdles or even feel daunted:

- Re-evaluate the problem statement periodically to assure alignment with objectives.
- Deconstruct multifaceted tasks into simpler, achievable steps.
- If faced with code errors or issues, turn to online forums or official documentation. Problem-solving acumen is indispensable for data scientists.
- Collaborate with colleagues. Engaging in the discussion forum can offer diverse perspectives, aiding in overcoming obstacles or sparking new ideas.

- Revisit lectures or explore external resources for topics you're unsure about.
- For any overarching issues or if the problem statement appears ambiguous, don't hesitate to contact your Instructor.

Remember, every challenge faced is an opportunity to grow. Approach this case with enthusiasm, diligence, and an open mind.

What does 'good' look like?

1. Define Problem Statement and perform Exploratory Data Analysis

	Hint	Approach
a. Definition of problem	Start by crystallizing the problem statement. What's the objective of NinjaCart? Why is classifying vegetables so important?	The main aim is to classify the correct vegetable from different images
b. Observations on Data	A thorough understanding of the dataset structure is key. Observe the shape of data, data type, size of the individual image, range of pixel values, number of channels	a. Visualize the data, use the dataset directory to create a list containing all the image paths in the training folder. b. You can use matplotlib or tensorflow to plot a grid sample of the images you fetched from the list of image paths.
c. Plotting Data	Begin the analysis with individual variables. For continuous attributes, use distribution plots, and for categorical ones, use bar or count plots.	a. For continuous variables, use histograms or density plots. For categorical variables, use countplots. Tools like Seaborn make these visualizations straightforward. This helps in understanding the distribution of individual variables. b. Plot a few of the images of each class to check their dimensions

		c. Verify the count of images in each train and test folder by plotting histogram . Check each folder to see if the number of images matches the reported number.

2. Data Preprocessing

	Hint	Approach
a. Irregular size treatment	• Perform rescaling and resizing of the data	a. Rescale image to range (0-1) and resize image to a square dimension
b. Class imbalance treatment	• Use data augmentation	a. Use tensorflow function for rotating the images to different angles for class with less samples. b. Use tensorflow function for random flipping the data samples
c. Plotting data	Analysis for image data for each class	a. Verify the count of images in each train and test folder by plotting histogram . b. Check each folder to see if the number of images matches the reported number.

3. Model building

	Hint	Approach
a. Data Splitting	Before model building, split the dataset into training, validation set and test set	The provided data does not contain separate training and validation folders. For us to do hyperparameter tuning of our models, it is important to divide the dataset into an 80-20 split for training and validation respectively.

b. CNN scratch	a.Create your own CNN mode b. Check for loss and overfitting	a. Use the tensorflow layer to find all the necessary functions to create your own model b. Use categorical loss and if the model is overfitting use batch normalization, dropout or simplify model
c. Pretrained model	a. Use transfer learning concept	a. Use a model of your choice (could be vgg, resnet, and mobilenet) and train it with appropriate batch size. b. Using the pre-trained weights of popular networks is a great way to do transfer learning since the size of our original dataset is small.

4. Results Interpretation & Stakeholder Presentation

	Hint	**Approach**
a. Interpret the model performance	Test accuracy	a. Obtain the testing accuracy to see how well your model generalizes. b.Compare and check the performance of multiple models using a confusion matrix.
b. Visual Representations	Visuals can often convey information more effectively than numbers alone.	a. Implement a TensorBoard callback to log each of our model metrics for each model during the training process. [recommended google colab or jupyter notebook] b. Plot the train/valid accuracy and loss graph for each model.
c. Feedback Loop	It's essential to ensure continuous monitoring and adaptability.	If the model is still overfitting, Try to tune your model by applying -:

		- BatchNormalization and Dropout - Callbacks: EarlyStopping, ModelCheckpoint and TensorBoard callback - Data Augmentation