

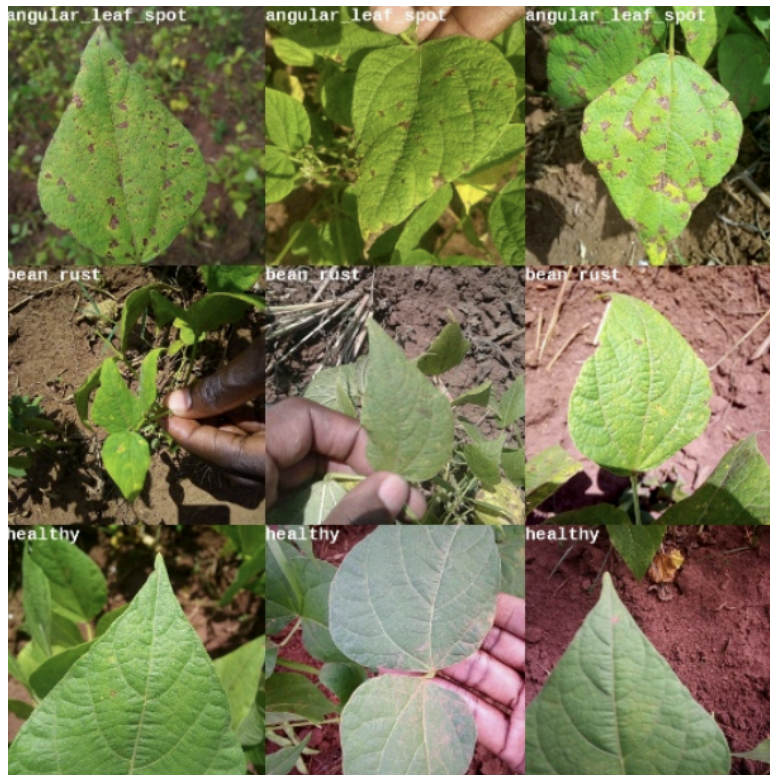
Summary of Notebooks

Notebook 1: Image Classification with Hugging Face

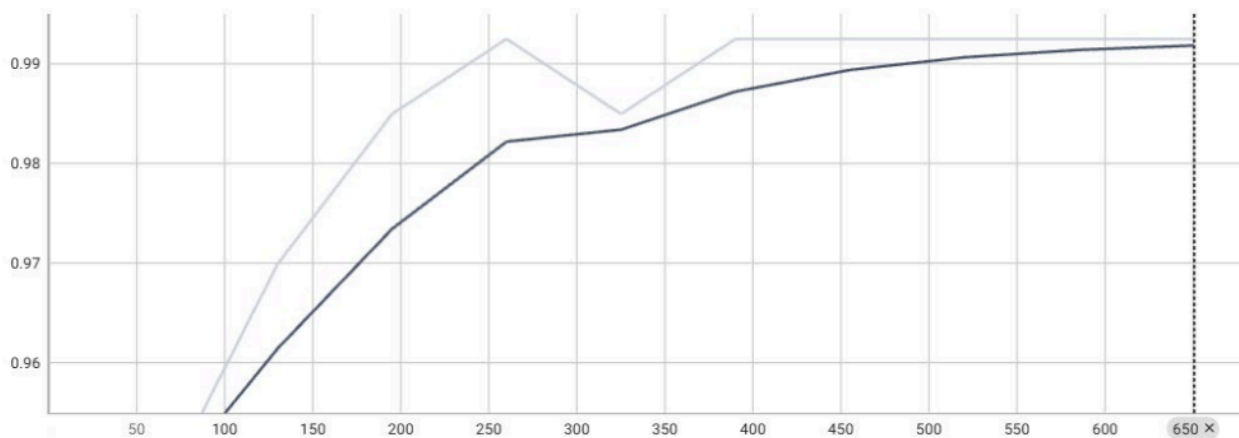
- **Model:** ViT (Vision Transformer)

```
ViTImageProcessor {  
  "do_convert_rgb": null,  
  "do_normalize": true,  
  "do_rescale": true,  
  "do_resize": true,  
  "image_mean": [  
    0.5,  
    0.5,  
    0.5  
  ],  
  "image_processor_type": "ViTImageProcessor",  
  "image_std": [  
    0.5,  
    0.5,  
    0.5  
  ],  
  "resample": 2,  
  "rescale_factor": 0.00392156862745098,  
  "size": {  
    "height": 224,  
    "width": 224  
  }  
}
```

- **Process:** Feature extraction → DataLoader setup → Training with PyTorch Trainer → Evaluation
- **Input:** Images (batch size = 16)



- **Output:** Classification (Healthy or Not)
- **Hardware:** CUDA (GPU)
- **Performance:**
 - Accuracy: **0.992**
 - Macro F1-score: **0.992**



- **Improvements:** Hyperparameter tuning

Notebook 2: Text Classification (TF-IDF)

- **Model:** TF-IDF Vectorizer + Naïve Bayes Classifier

```
%%time
from sklearn.feature_extraction.text import TfidfVectorizer

# Preprocess text
X_train_preprocessed = np.array([text_preprocessing(text) for text in X_train])
X_val_preprocessed = np.array([text_preprocessing(text) for text in X_val])
X_test_preprocessed = np.array([text_preprocessing(text) for text in X_test])

# Calculate TF-IDF
tf_idf = TfidfVectorizer(ngram_range=(1, 3),
                        binary=True,
                        smooth_idf=False)
X_train_tfidf = tf_idf.fit_transform(X_train_preprocessed)
X_val_tfidf = tf_idf.transform(X_val_preprocessed)
X_test_tfidf = tf_idf.transform(X_test_preprocessed)
```

```
from sklearn.model_selection import StratifiedKFold, cross_val_score

def get_auc_cv(model):
    """
    Return the average AUC score from cross-validation.
    """
    # Set KFold to shuffle data before the split
    kf = StratifiedKFold(5, shuffle=True, random_state=1)

    # Get AUC scores
    auc = cross_val_score(
        model, X_val_tfidf, y_val, scoring="roc_auc", cv=kf)

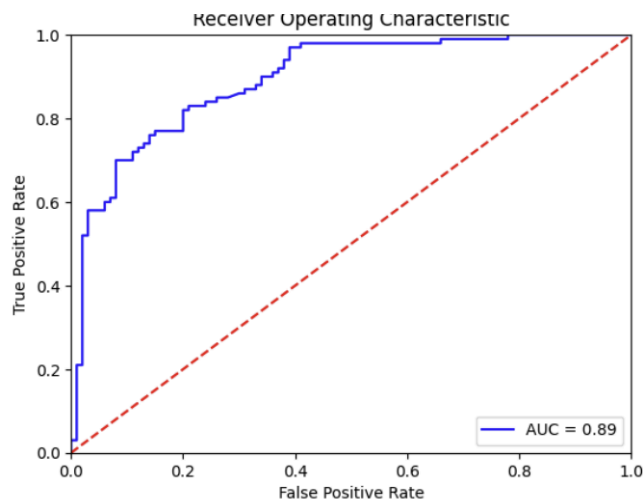
    return auc.mean()
```

- **Process:** Data loading → Train/Test split (60/20/20) → Text processing → Training → Evaluation
- **Input:** Review sentences

	review	label
0	So there is no way for me to plug it in here i...	0
1	Good case, Excellent value.	1
2	Great for the jawbone.	1
3	Tied to charger for conversations lasting more...	0
4	The mic is great.	1

- **Output:** Classification (0 or 1)
- **Hardware:** CUDA (GPU) or CPU

- **Performance:**
 - Accuracy: **0.78**
 - AUC: **0.8938**



Notebook 3: Text Classification (BERT)

- **Model:** BERT

```
# Load model
model = AutoModelForSequenceClassification.from_pretrained("bert-base-cased", num_labels=2)
```

- **Process:** Data loading → Tokenization → Training → Evaluation
- **Input:** Review sentences

	review	label
0	So there is no way for me to plug it in here i...	0
1	Good case, Excellent value.	1
2	Great for the jawbone.	1
3	Tied to charger for conversations lasting more...	0
4	The mic is great.	1

- **Output:** Classification (0 or 1)
- **Hardware:** CUDA (GPU)
- **Performance:**

- F1-score: **0.88**
- Recall: **0.87**

```
{'eval_loss': 0.3862229883670807,
 'eval_accuracy': 0.885,
 'eval_runtime': 0.8504,
 'eval_samples_per_second': 235.179,
 'eval_steps_per_second': 58.795,
 'epoch': 2.0}
```

- **Improvements:** More epochs, hyperparameter tuning

Notebook 4: Text Classification (PhayaThaiBERT)

- **Model:** PhayaThaiBERT (Pretrained)

```
checkpoint = "clicknext/phayathaibert"
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
```

- **Process:** Data loading → Tokenization → DataLoader setup → Training → Evaluation
- **Input:** Review sentences

	review_body	star_rating
0	ร้านอาหารใหญ่มากกกกกกกก โทเลี้ยวเข้ามาเจอห้องนี้...	2
1	อาหารที่นี่เป็นอาหารจีนแคะที่หากินยากในบ้านเรา...	3
2	ปอเปี๊ยะสด ทุกวันนี้รู้สึกว่หาากินยาก (ร้านที่...	2
3	ร้านคัพเค้กในเมืองไทยมีไม่มาก หลายคนอาจจะสงส์...	4
4	อร่อย!!! เดินผ่านDigital gatewayทุกวัน ไม่ยักร...	4

- **Output:** Rating (0 to 5)
- **Hardware:** CUDA (GPU)
- **Performance:**
 - Accuracy: **0.58**
 - Recall: **0.40**

F1-score: 0.5070952837216167

Accuracy: 0.564

Classification Report:

	precision	recall	f1-score	support
1	0.33	0.20	0.25	5
2	0.43	0.16	0.23	19
3	0.53	0.73	0.62	141
4	0.59	0.71	0.65	243
5	0.40	0.02	0.04	92
accuracy			0.56	500
macro avg	0.46	0.36	0.36	500
weighted avg	0.53	0.56	0.51	500

- **Improvements:** More epochs, hyperparameter tuning