Galaxy Morphology Classifier

- David Thornton

- B00152842

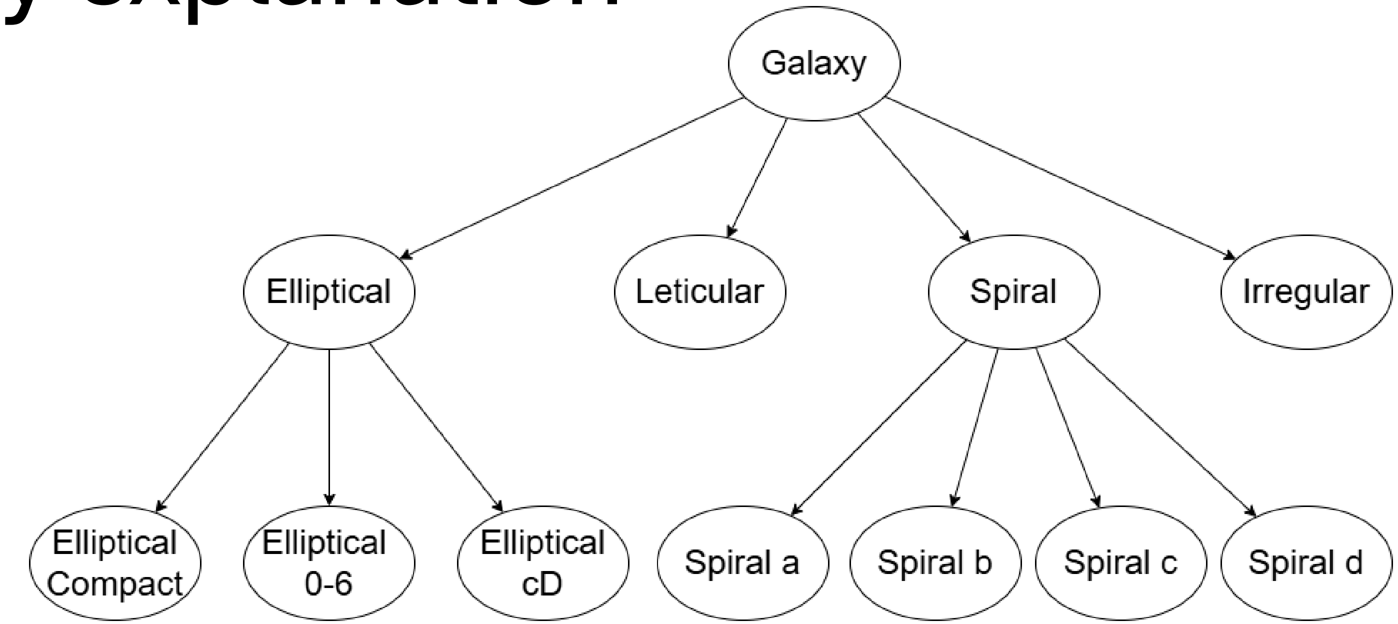
Project Overview

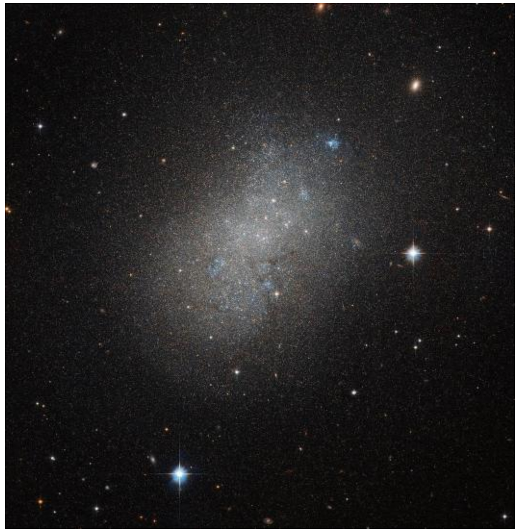
- Galaxy Morphology Classifier
- Hierarchical classification and image feature extraction techniques
- Provides insight in astronomy research

Motivation

- Can be identified by people with relatively high accuracy
- Amount of data makes this impractical
- Machine learning can speed up process
- Current work
  - Neglects hierarchical approaches
  - Lacks any explainability or insight

Galaxy explanation

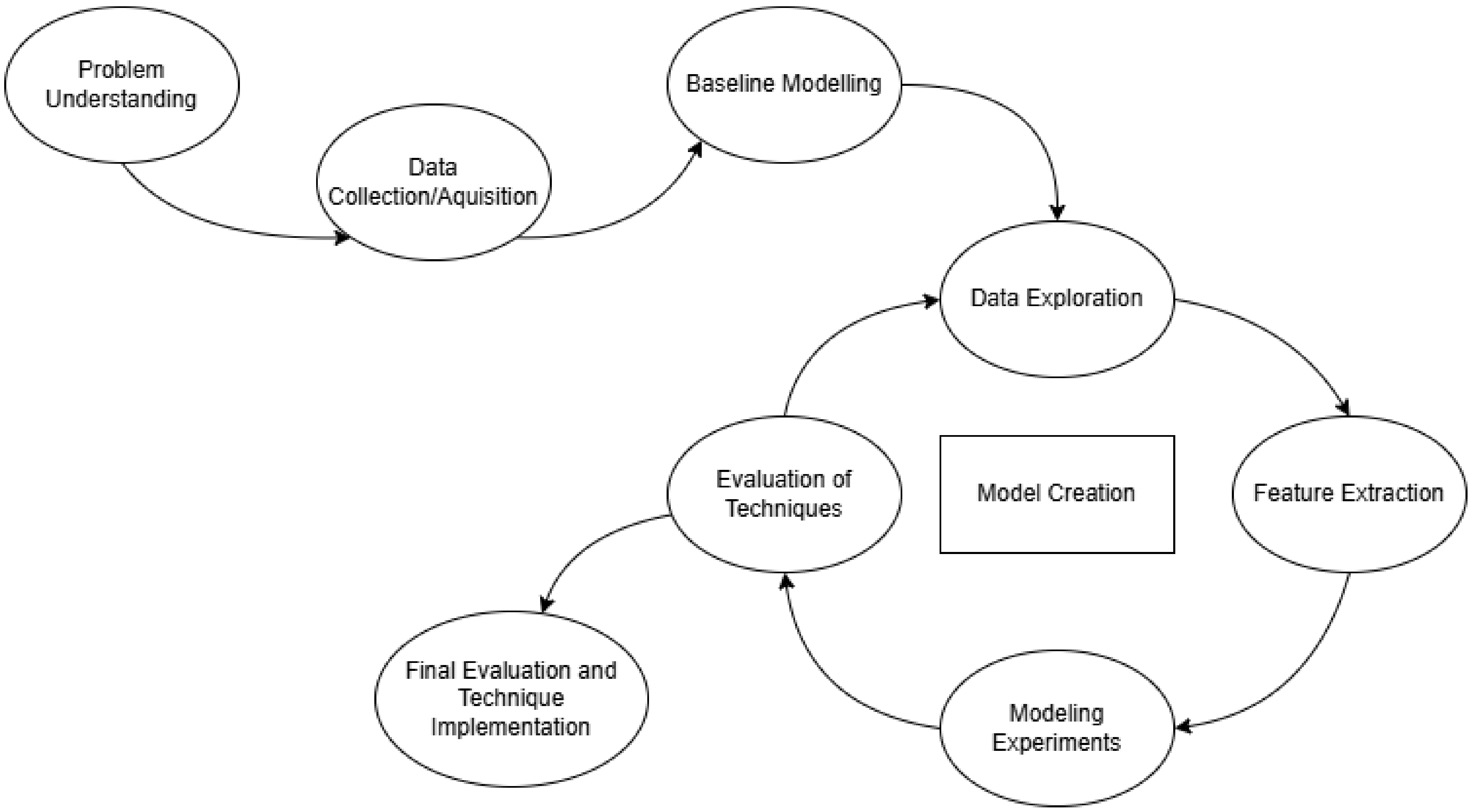




Aim and Research Questions

- Aim
  - To develop a hierarchical galaxy morphology classification model
- Research Questions
  - How do flat classification models perform?
  - To what level do hierarchical models outperform flat models?
  - What feature extraction techniques are suitable?
  - {Optional} How effective is implementing explainability at informing decision making?
- Objectives
  - Research what has been done in current literature
  - Experiment with hierarchical and flat approaches, compare
  - Experiment with different feature extraction techniques, compare
  - Evaluate explainability of approaches and interpret results
  - Use most optimal techniques to produce accurate model

Modified Crisp-DM



Dataset

- EFIGI data set
- Chosen for
  - Label quality
  - Detailed class labels
  - Usefulness of additional labels
- Main Limitations
  - Severe data imbalances
  - Dataset size

Feature Extraction

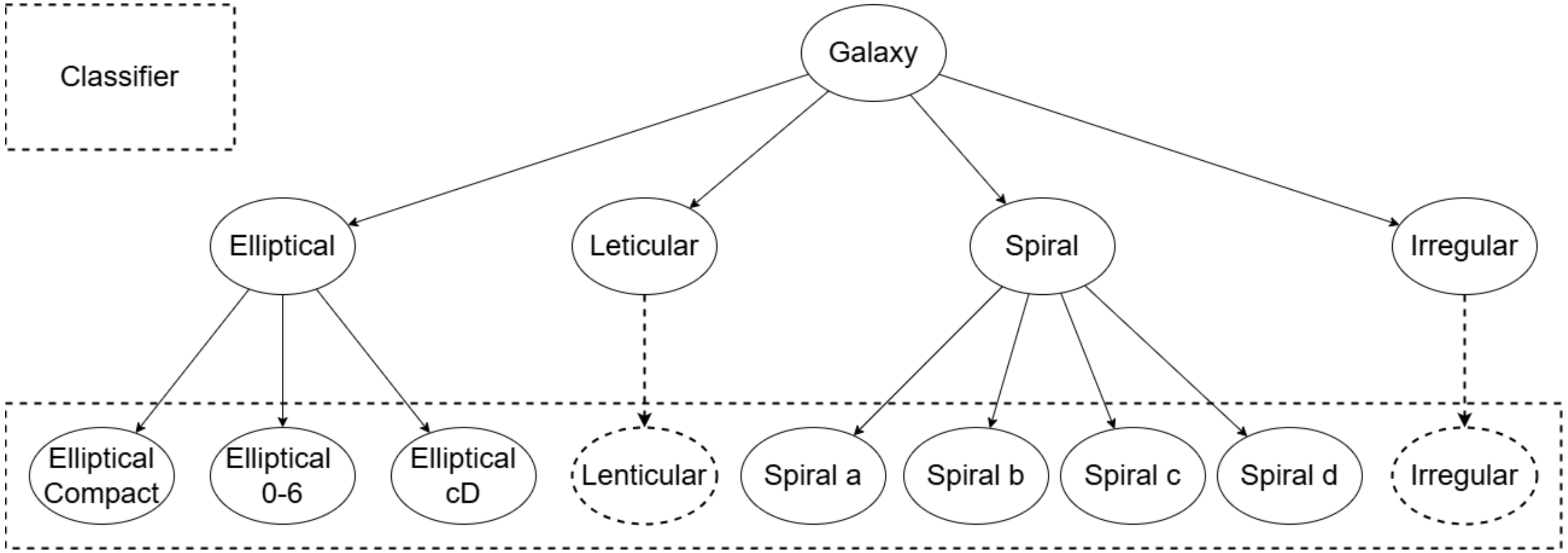
Local Binary Pattern

- Extracted using Skimage local binary pattern class method
- Represents the textural pattern of the image
- Compares the brightness of pixels with all the pixels in a given radius
- Uses these differences to make a code

CNN Automatic feature extraction

- Implemented with Pytorch
- Pretrained Resnet18 model
- Trained on raw image data
- Single stage training with classifier heads

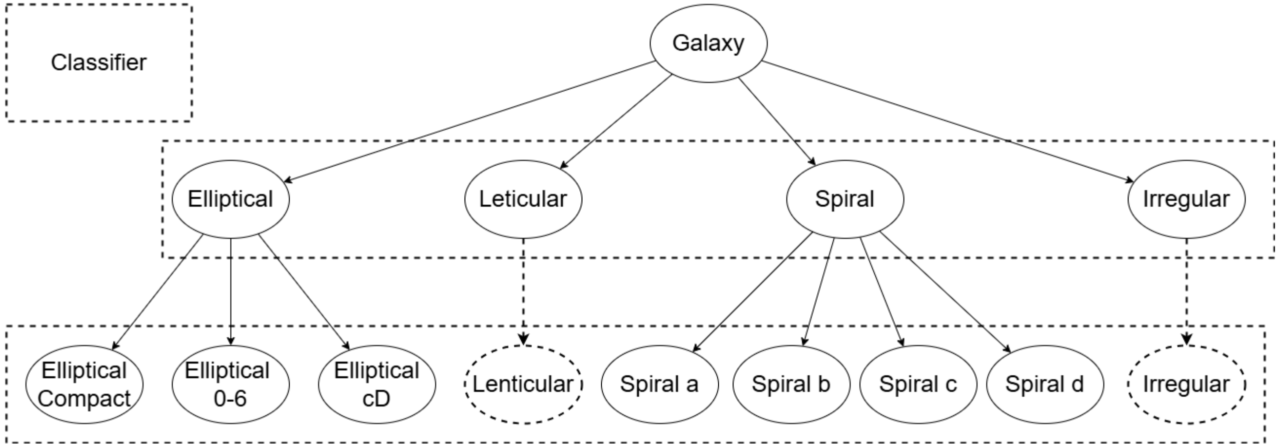
Flat Approach



- Single linear sequential classifier head
- No coarse class prediction
- Basic cross entropy for loss

Classifier Per Level Approach

- 2 linear sequential classifier heads
  - 1 predicts coarse classes
  - 1 predicts fine classes
- Combined cross entropy and consistency penalty for loss
- Additional method for combined fine class predictions



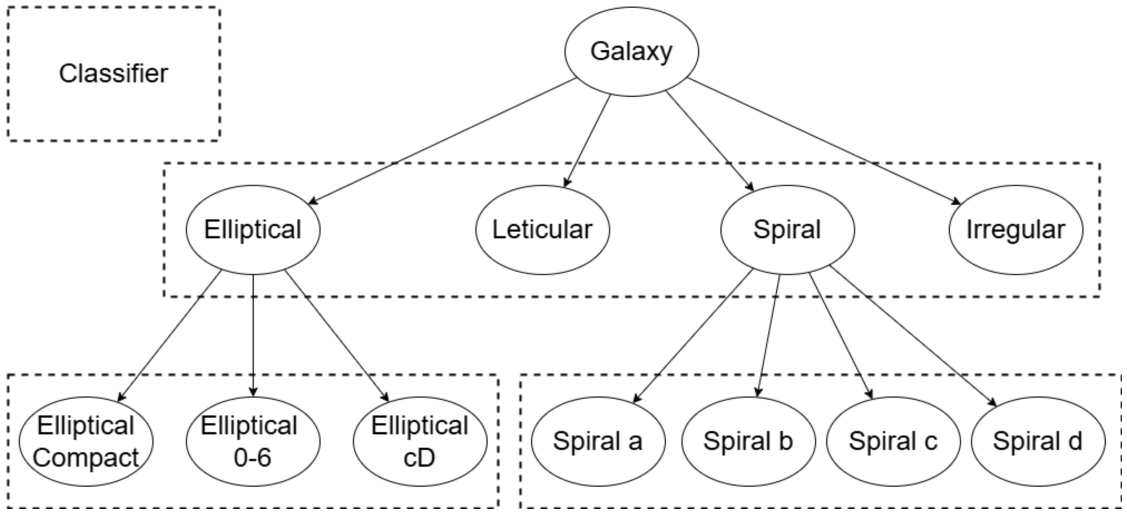
$$\mathrm{Loss}_t = Loss_f + Loss_c \times \alpha + Penalty \times \beta$$

$$\mathrm{P(Combined\ Fine)} = P(Coarse) \times P(Fine|Coarse)$$

$$\mathrm{P(Fine|Coarse)} = \frac{P(Fine)}{\sum P(FineInGroup)}$$

Classifier Per Node Approach

- 3 linear sequential classifier heads
  - 1 predicts coarse classes
  - 2 predict fine classes where coarse has multiple fine
- Combined cross entropy for loss
- Combined fine class loss needed



$$\mathrm{Loss}_t = \sum Loss_i$$

$$\mathrm{P(Final\ Fine)} = P(Coarse) \times P(Fine|Coarse)$$

Training and Testing Procedure and Baseline Modelling

Training Procedure

- Split train test
- 70% train, 10% validation, 20% test
- Sklearn’s train test split with stratified hyperparameter
- Pytorch’s Dataset and Data loader

Baseline Modelling

- 3 baseline models
  - Flat Classifier Model
  - Classifier per Level
  - Classifier per Node
- Cross Entropy Loss with class weights
- Adam optimizer with learn rate 0.01

Evaluation Metrics

- Macro average F1
- Goal is overall accuracy across *all classes*
- Class specific accuracy also assessed with F1
- Hierarchical measures for each also considered

$$\mathrm{MA} = \frac{\sum_{i=1}^{n} F_1}{n}$$

$$F_1 = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i}$$

$$\mathrm{hPrecision_i} = \frac{\left|Y_{\text{pred}}^{\text{hier}} \cap Y_{\text{true}}^{\text{hier}}\right|}{\left|Y_{\text{pred}}^{\text{hier}}\right|}$$

$$\mathrm{Precision}_i = \frac{TP_i}{TP_i + FP_i}$$

n = number of classes

TP = true positive rate

FP = false positive rate

FN = false negative rate

$$\mathrm{hRecall_i} = \frac{\left|Y_{\text{pred}}^{\text{hier}} \cap Y_{\text{true}}^{\text{hier}}\right|}{\left|Y_{\text{true}}^{\text{hier}}\right|}$$

$$\mathrm{Recall}_i = \frac{TP_i}{TP_i + FN_i}$$

Results from Baselines

Flat

Class	Precision	Recall	F1-score	Support
0	0.3333	0.7500	0.4615	4
1	0.6949	0.9111	0.7885	45
2	0.5833	0.7778	0.6667	9
3	0.8571	0.6729	0.7539	107
4	0.7023	0.6815	0.6917	135
5	0.7302	0.6970	0.7132	198
6	0.6646	0.7133	0.6881	150
7	0.8563	0.8232	0.8394	181
8	0.7808	0.9048	0.8382	63
Accuracy			0.7466	892
Macro avg	0.6892	0.7702	0.7157	892
Weighted avg	0.7543	0.7466	0.7473	892

Classifier Per Layer

Fine classes

Class	Precision	Recall	F1-score	hF1-score	Support
0	0.7500	0.7500	0.7500	0.6667	4
1	0.8333	0.7778	0.8046	0.8939	45
2	0.4615	0.6667	0.5455	0.7778	9
3	0.7143	0.8879	0.7917	0.9112	107
4	0.7315	0.5852	0.6502	0.6926	135
5	0.7418	0.6818	0.7105	0.8384	198
6	0.6707	0.7333	0.7006	0.8633	150
7	0.8537	0.7735	0.8116	0.8287	181
8	0.6951	0.9048	0.7862	0.9048	63
Accuracy				0.7399	892
Macro avg	0.7169	0.7512	0.7279	0.8337	892
Weighted avg	0.7462	0.7399	0.7386		892

Coarse classes

Class	Precision	Recall	F1-score	Support
0	0.8361	0.8793	0.8571	58
1	0.7287	0.8785	0.7966	107
2	0.9871	0.9247	0.9549	664
3	0.7125	0.9048	0.7972	63
Accuracy			0.9148	892
Macro avg	0.8161	0.8968	0.8515	892
Weighted avg	0.9269	0.9148	0.9184	892

Classifier Per Node

Fine classes

Class	Precision	Recall	F1-score	hF1-score	Support
0	0.7500	0.7500	0.7500	0.8890	4
1	0.7213	0.9778	0.8302	0.9889	45
2	0.8333	0.5556	0.6667	0.7222	9
3	0.7500	0.8411	0.7930	0.9040	107
4	0.7938	0.5704	0.6638	0.6852	135
5	0.7157	0.7374	0.7264	0.8687	198
6	0.6543	0.7067	0.6795	0.8533	150
7	0.8642	0.7735	0.8163	0.8370	181
8	0.7500	0.9048	0.8201	0.8968	63
Accuracy				0.7489	892
Macro avg	0.7592	0.7575	0.7495	0.8427	892
Weighted avg	0.7555	0.7489	0.7466		892

Coarse classes

Class	Precision	Recall	F1-score	Support
0	0.7778	0.9655	0.8615	58
1	0.7563	0.8411	0.7965	107
2	0.9872	0.9322	0.9589	664
3	0.7568	0.8889	0.8175	63
Accuracy			0.9204	892
Macro avg	0.8195	0.9069	0.8586	892
Weighted avg	0.9296	0.9204	0.9231	892

Comparative Analysis of Baselines

- Baselines indicate promise for hierarchical architectures
  - Better Macro F1 scores
- Baselines indicate promise for classifier per node
  - Better F1 accuracy for underrepresented classes

Classifier	Flat	Classifier per Level	Classifier per Node
Coarse Marco F1	N/a	85	85
Fine Macro F1	71	74	75
Coarse F1 range	N/a	79 - 95	80 - 96
Fine F1 range	46 - 83	55 - 81	66 - 83
Hierarchical Macro F1	N/a	83	84
Hierarchical F1 range	N/a	67 - 91	69 - 98

Plans for Future Work

- Alleviate class imbalances
  - Combine with other datasets
  - Data creation
- Feature extraction
  - Geometric ratio based features
  - Lower dimensionality features to represent image aspects
- Data preparation
  - Dimensionality reduction (PCA)
  - Feature ranking (explainability tool)
  - Feature selection
- Model implementations
  - SVC, tree based, and neural network based
  - In flat and hierarchical implementations

Gantt Chart

Task Name	Start	Finish
Proposal	Mon 29/09/25	Fri 17/10/25
Literature Review	Mon 20/10/25	Sat 28/02/26
Dataset Selection	Thu 09/10/25	Fri 31/10/25
Understanding of Datasets	Thu 09/10/25	Sun 15/03/26
Developing Base Models	Sat 01/11/25	Tue 02/12/25
Feature Extraction Experiments	Wed 12/11/25	Sun 29/03/26
Modelling Experimentation	**Wed 12/11/25**	**Sun 26/04/26**
Model selection	Wed 12/11/25	Sun 19/04/26
Hyperparameter Tuning	Mon 19/01/26	Sun 26/04/26
Preprocessing	Mon 02/02/26	Sun 12/04/26
Evaluating Performance	Mon 24/11/25	Sun 10/05/26
Documentation of Experimentation	Mon 24/11/25	Sun 26/04/26
Finalisation of Thesis	Mon 27/04/26	Sun 10/05/26
Project Demonstration Preperation	Thu 30/04/26	Sun 10/05/26