2024 인공지능 특강 하계 실습

인공지능융합원장 김광수
kim.kwangsu@skku.edu

개요

1. 프로그램명 : 2024 비전공 교원 대상 인공지능(AI) 실습특강 및 교류
2. 목표 : 데이터를 다루는 능력과 AI모델 활용 능력 강화
3. 일 시 : 2024.7.29.(월) ~ 8.1.(목) / 총 4일 오전 10 ~ 17시 (점심시간 12:00 ~ 13:00)
4. 장 소 : 자연과학캠퍼스 화학관 330102호
5. 대 상
    1) 기존 비전공교원대상 인공지능 특강 수강교원 중 희망자
    2) 교원 추천 대학원생, 연구원, 비전임교원
6. 강의교원 : AI 전임교원 2명 (소프트웨어학과 김광수, 박진영 교수)
7. 주관 : 성균융합원, 인공지능융합원
8. 지원 : 4단계 BK21 대학원혁신지원사업

일정

구분	2024.7.29.(월)	7.30.(화)	7.31.(수)	8.1.(목)
하계 (실습기반)	기계학습, 딥러닝 분야(실습) 분반 (소프트웨어학과, 김광수 교수)		자연어처리 분야(실습) 분반 (소프트웨어학과, 박진영 교수)	

시 간		프 로 그 램
기계학습. 딥러닝 분야 실습 (김광수 교수)		
7. 29.	10:00~10:10	개회 및 환영사 (유지범 총장)
	10:10~17:00	AutoML 활용 이미지 예측, 객체 검출 및 테이블 데이터 처리
7. 30.	10:00 ~ 17:00	Foundation Model 활용 이미지 분류 및 라벨링
자연어처리 분야 실습 (박진영 교수)		
7. 31.	10:00 ~ 17:00	ChatGPT API를 활용한 Text Classification, 및 Evaluation
8. 1.	10:00 ~ 17:00	ChatGPT API를 활용한 Text Summarization, Evaluation 및 Table data 분석

※12:00~13:00 중식 및 네트워킹 (복지회관 3층 구시재)

AutoML

소프트웨어학과 김광수 교수 (kim.kwangsu@skku.edu)
인공지능융합연구실 장승우 (sewo@skku.edu)

Machine learning

"Machine learning is the science of getting computers to act without being explicitly programmed."

ML in Nutshell

Process of Weeks and Months

Raw Data

Preprocessing

- Data Cleaning
- Data Normalization
- Data Standardization
- Feature Selection
- Dimensionality Reduction
- Splitting Data
- Handling Imbalanced Data
- Feature Scaling

Algorithm and Hyperparameter Selection

- Choose a Validation Strategy
- Select algorithms
- Select modeling techniques
- Tune hyperparameters

Model

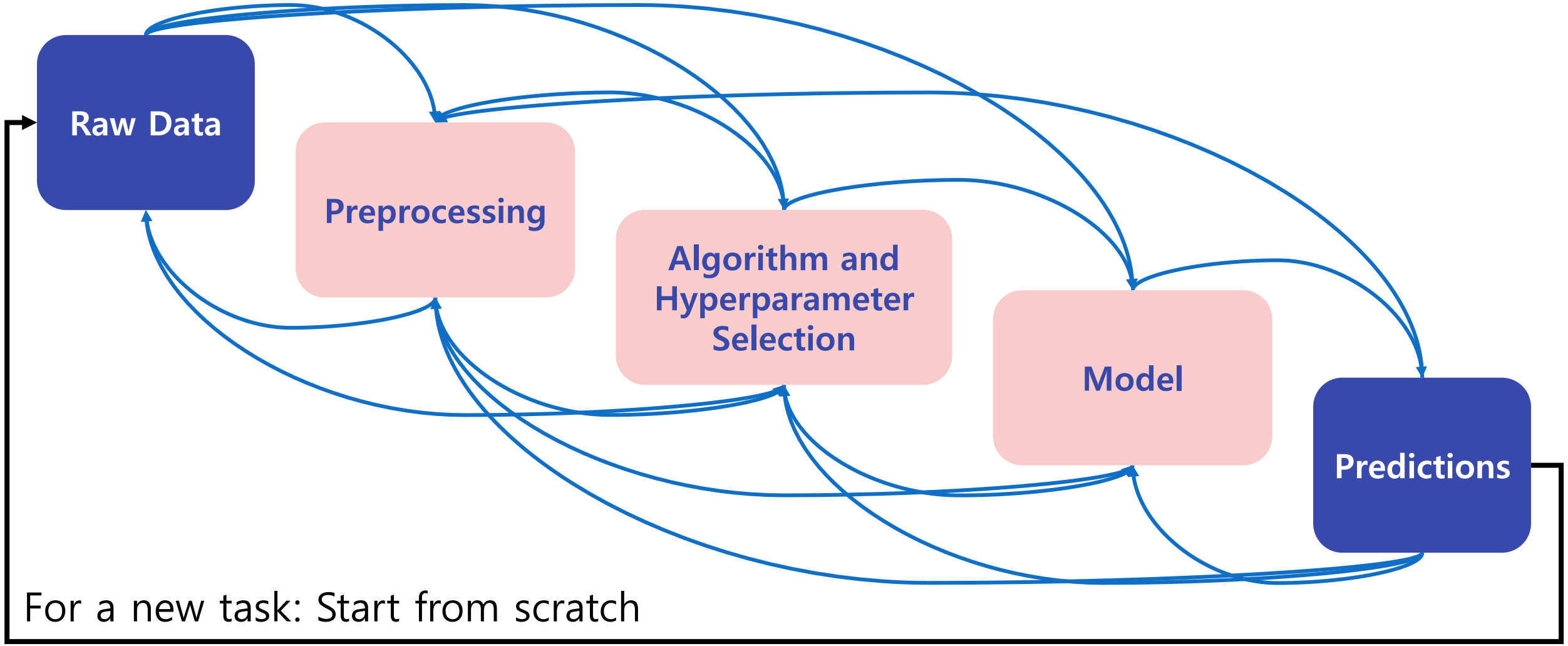
- Select optimization metric
- Validate partitioning
- Analyze ROC curves
- Analyze Confusion matrix
- Score and compare models

Predictions

Challenges in Applying AI/ML these days

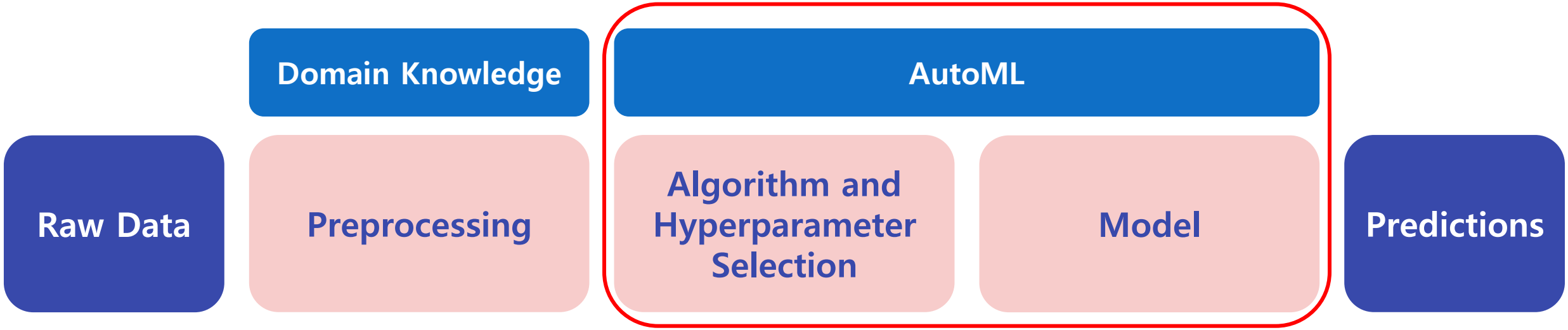
1. Required expertise in ML and AI
2. Long development time for new AI applications
3. Few experts are available on the job market
4. Unstructured and error-prone development of AI application

Why does ML development take a lot of time?



What is AutoML?

- Automated Machine Learning provides methods and processes to make Machine Learning available for non-Machine Learning experts



Advantages of AutoML

1. More efficient research and development of ML applications
   - AutoML has been shown to outperform humans on subproblems
2. More systematic research and development of ML applications
   - No (human) bias or unsystematic evaluation
3. More reproducible research
   - Since it is systematic
4. Broader use of ML methods
   - Less required ML expert knowledge
   - Not only limited to computer scientists

Applications

Contents lists available at ScienceDirect

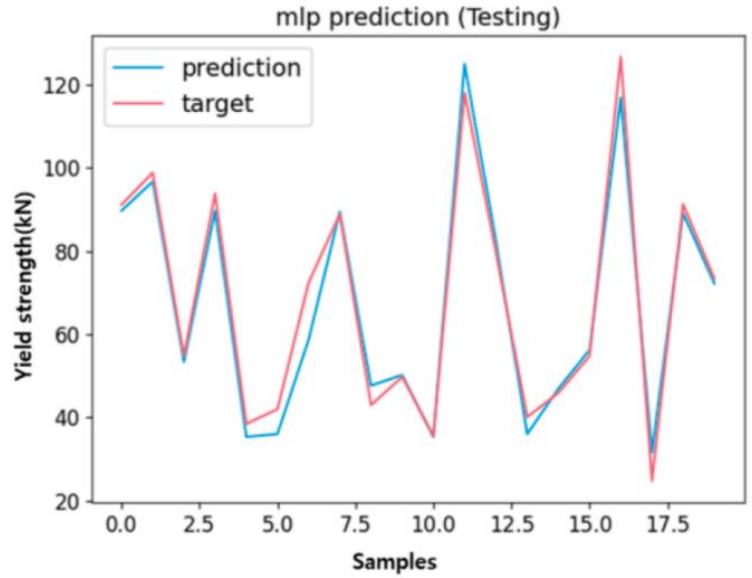
Construction and Building Materials

journal homepage: www.elsevier.com/locate/conbuildmat

Analysis of load-bearing capacity factors of textile-reinforced mortar using multilayer perceptron and explainable artificial intelligence

Youngjae Song [a], Kwangsu Kim [a], Seunghee Park [b,*,1], Sun-Kyu Park [b], Jongho Park [c,*,2]

[a] College of Computing and Informatics, Sungkyunkwan University, Suwon 16419, Republic of Korea
[b] Department of Civil, Architectural and Environmental System Engineering, Sungkyunkwan University, Suwon 16419, Republic of Korea
[c] Global Frontiers of Resilient EcoSmart City, Sungkyunkwan University, Suwon 16419, Republic of Korea



- Textile-based concrete composites show complex behaviors affected by variables like textile geometry and curing methods. To design effective TRM (textile-reinforced mortar) strengthening methods, a precise performance evaluation system is needed. We use AutoML to assess how external factors impact TRM performance.

Applications

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Predicting biomass composition and operating conditions in fluidized bed biomass gasifiers: An automated machine learning approach combined with cooperative game theory

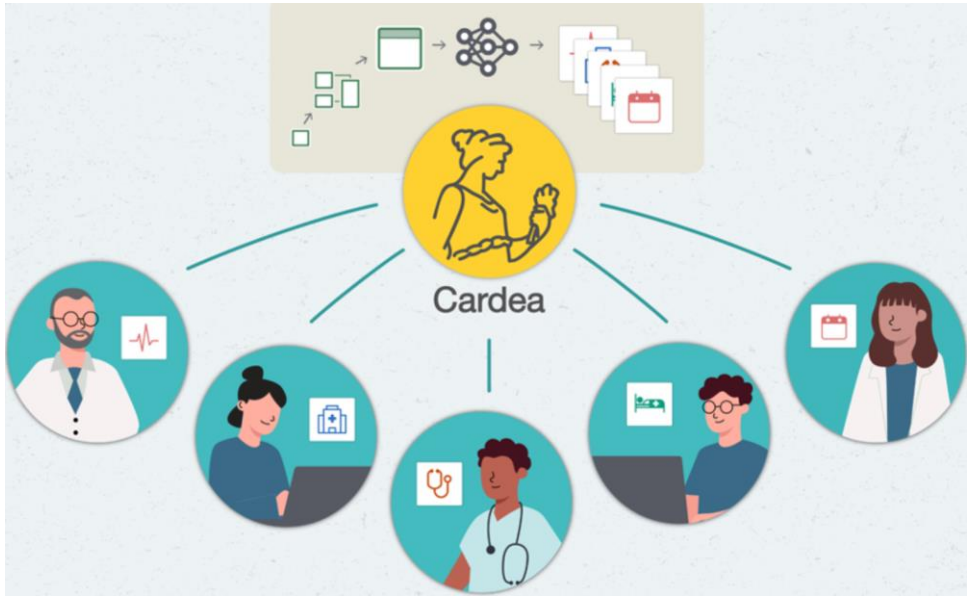
Jun Young Kim [a,*], Ui Hyeon Shin [b], Kwangsu Kim [c]

[a] School of Chemical Engineering, Sungkyunkwan University, 2066 Seobu-Ro, Jangan, Suwon, 16419, Republic of Korea
[b] Department of Artificial Intelligence, Sungkyunkwan University, 2066 Seobu-Ro, Jangan, Suwon, 16419, Republic of Korea
[c] College of Computing and Informatics, Sungkyunkwan University, 2066 Seobu-Ro, Jangan, Suwon, 16419, Republic of Korea

Models	CatBoost		WeightedEsemble_L2		ExtraTreeMSE		NeuralNetFastAI	
Indicators	R^2	**RMSE**	R^2	**RMSE**	R^2	**RMSE**	R^2	**RMSE**
Cell.	0.753	0.044	0.774	0.042	0.714	0.048	0.720	0.047
Hem.	0.753	0.052	0.722	0.055	0.660	0.061	0.586	0.068
Lignin	0.684	0.049	0.675	0.050	0.671	0.050	0.626	0.053
Temp.	0.731	0.444	0.723	0.451	0.729	0.446	0.704	0.466
Pressure	0.651	0.471	0.704	0.434	0.629	0.485	0.659	0.465
ER	0.886	0.051	0.874	0.053	0.869	0.054	0.860	0.056
SBR	0.673	0.355	0.624	0.381	0.653	0.365	0.643	0.371
U_g	0.377	0.292	0.410	0.284	0.472	0.269	0.460	0.272
Average	0.689	0.220	0.688	0.219	0.675	0.222	0.657	0.225

- A study used AutoML to select the best machine learning model for predicting biomass gasification outcomes, with AutoML showing strong performance. SHAP was applied for model interpretability, offering a new approach for efficient and insightful process monitoring.

Applications



Healthcare – Identifying Transplantation Prospects:

University of Pittsburgh Medical Center previously had to contact 10,000 prospective patients, who were than narrowed down using 1,500 different criteria.

Instead of months, the model development now takes hours to reach 10,000 potential candidates instead of just 75 previously.

Applications

Financial Services – Fraud Detection:

PayPal, a well-known online payment platform, utilized AutoML tool to enhance its fraud detection model. As a result of implementing AutoML, the accuracy of their model increased from 89% to 94.7%.complex attack patterns not previously observed.

AutoGluon

AutoML Libraries

AutoGluon Overview

- AutoGluon is an AutoML Toolkit that **democratizes machine learning**
- Automatically builds accurate models for Image, Text, Time Series, and Tabular data with **a single line of code**
- Open source & easy to install
- AutoGluon is adopted by over 100 companies
  Intel, NVIDIA, IBM, Capcom, Mitsubishi, …

AutoGluon and AutoML

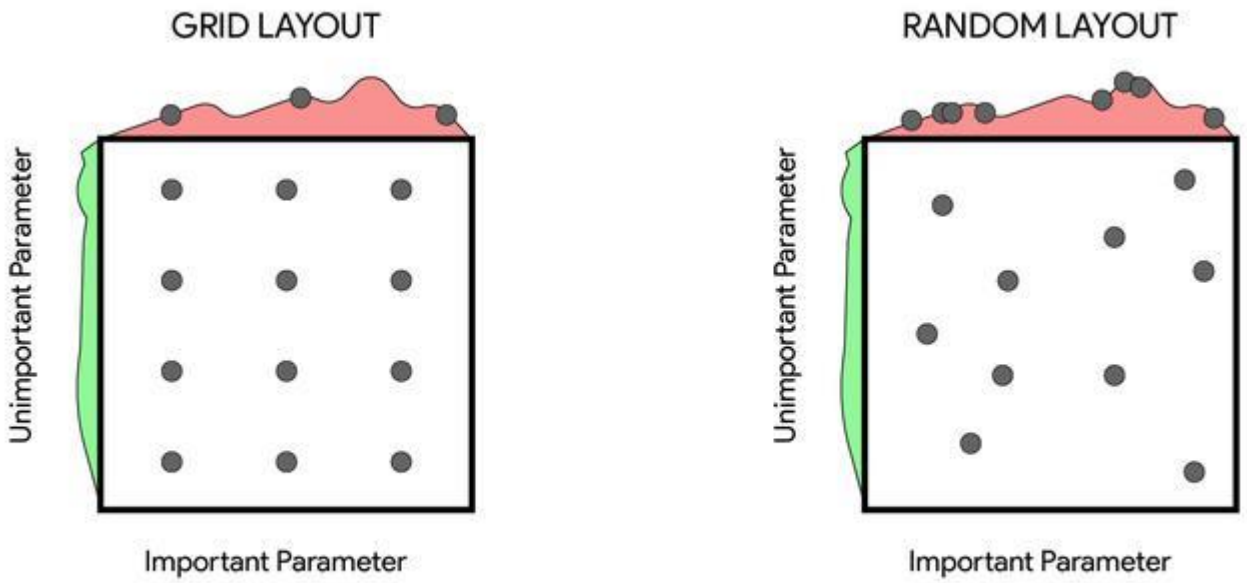
Prior work: **AutoML = Model + Hyperparameter Selection (CASH)**

AutoGluon: Rely on strategies that win prediction contests

1. Data preprocessing
2. Emphasis on modern deep learning techniques
3. Lots of tricks and optimizations
4. Ensembling via multi-layer stacking

Hyperparameter tuning: Benefit

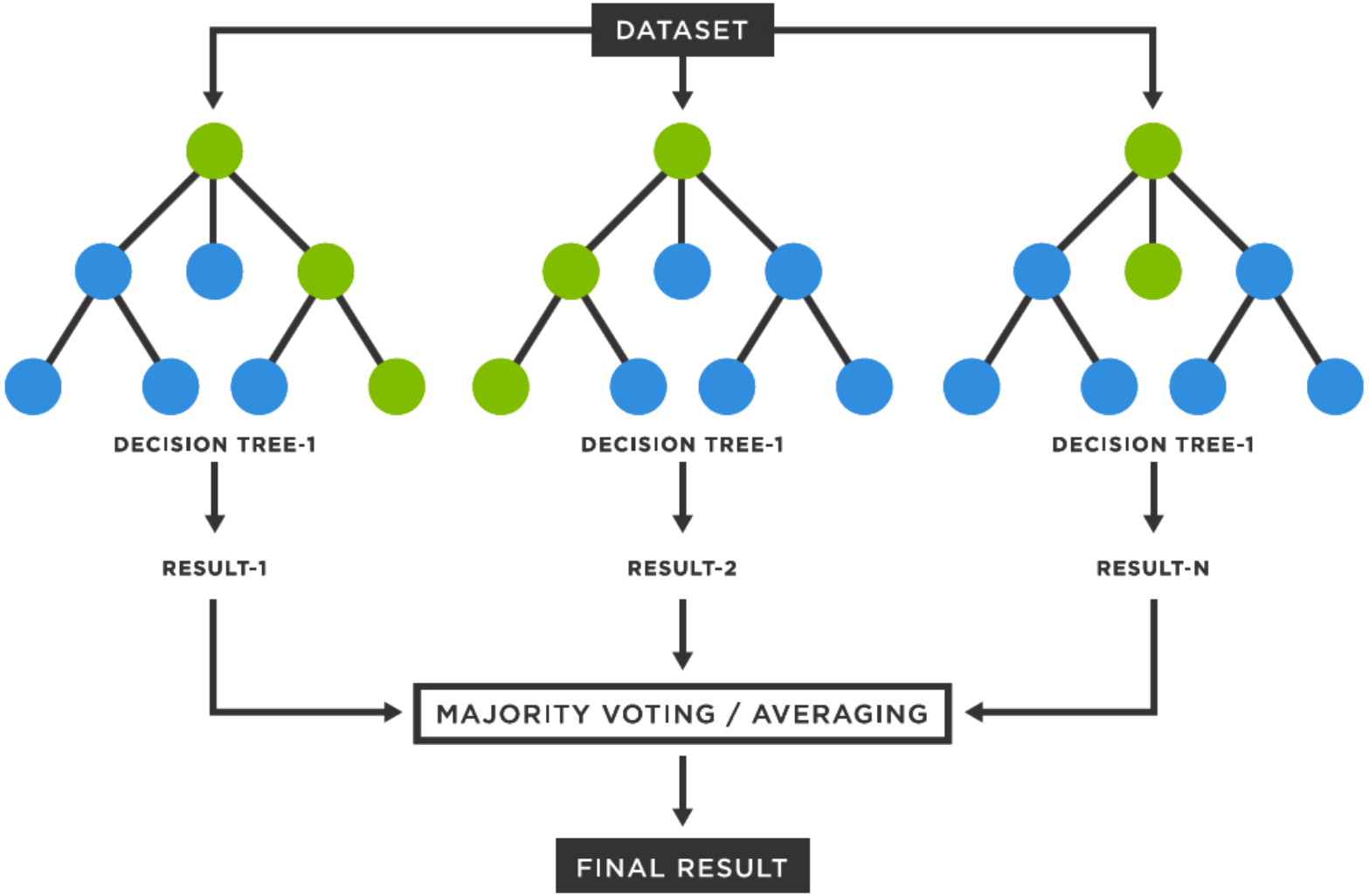
Finding the hyperparameter values of a learning algorithm that produce the best model.



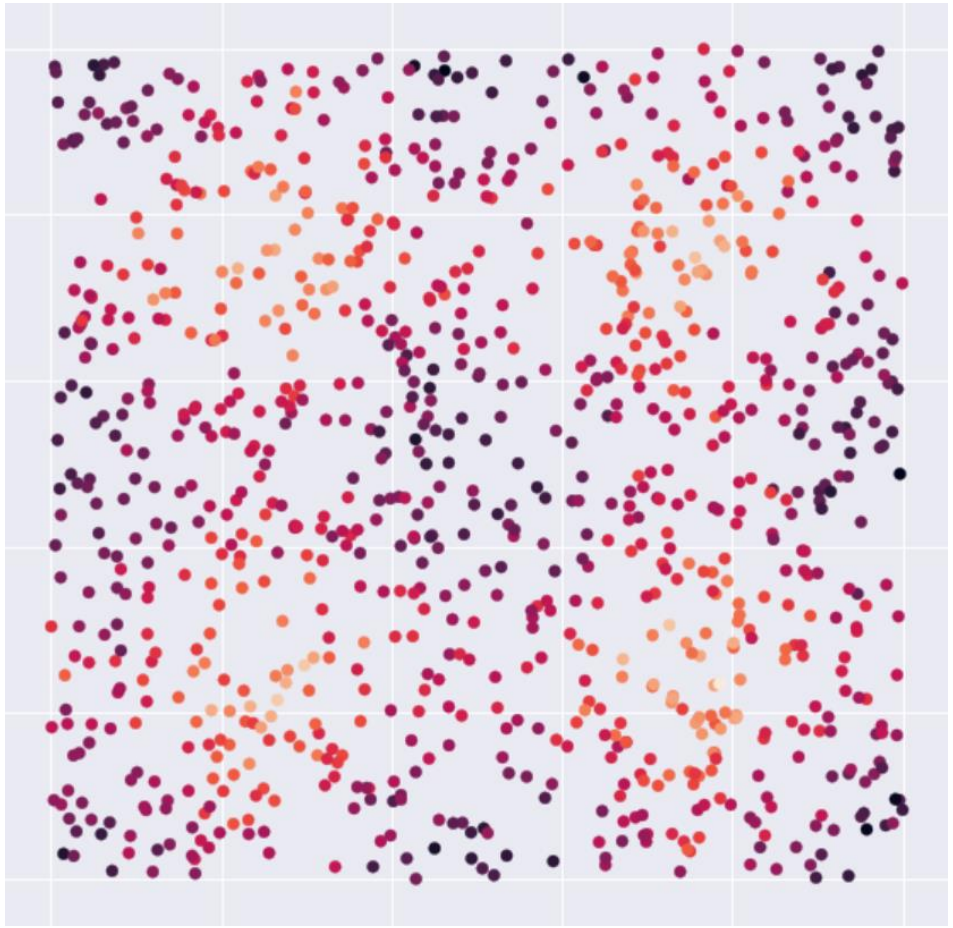
Hyperparameter tuning: Drawbacks

- Requires many repeated model trainings
- Most of the models being trained are thrown away and don't contribute to the final result
- The more hyperparameter tuning done, the more the final model is overfit on the validation data
- Less helpful when ensembling

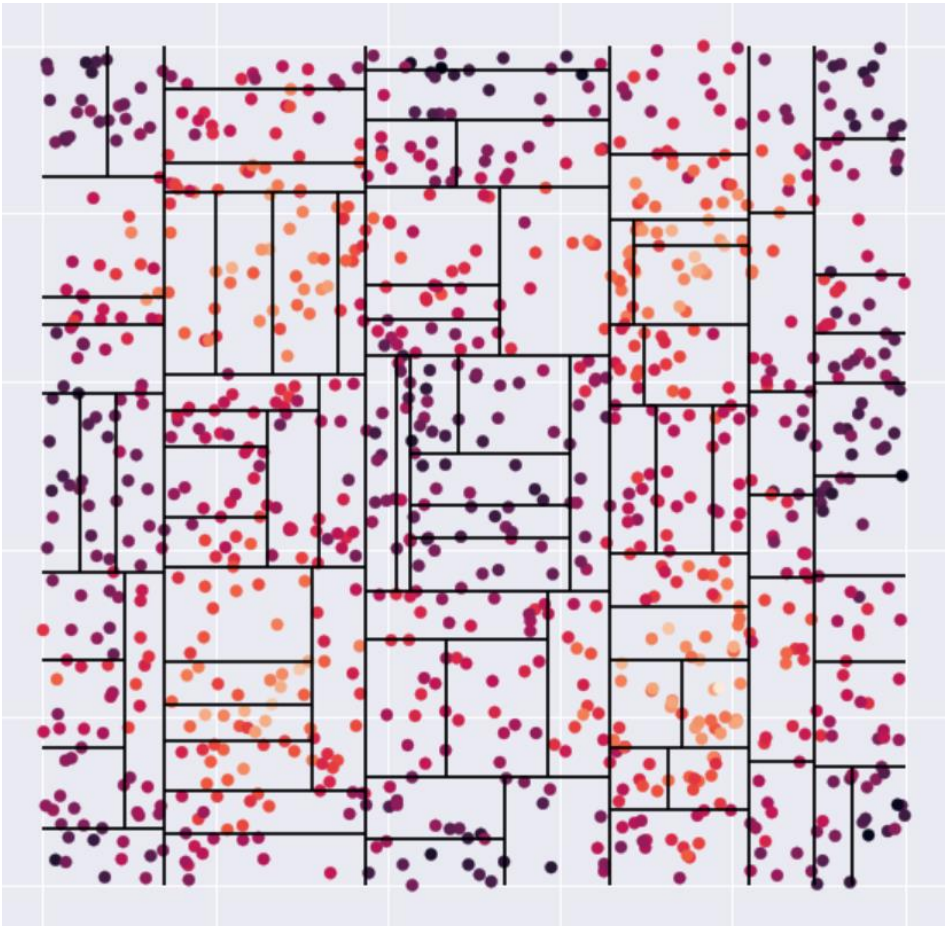
Introduction to ensemble: Random forest



Introduction to ensemble: Random forest

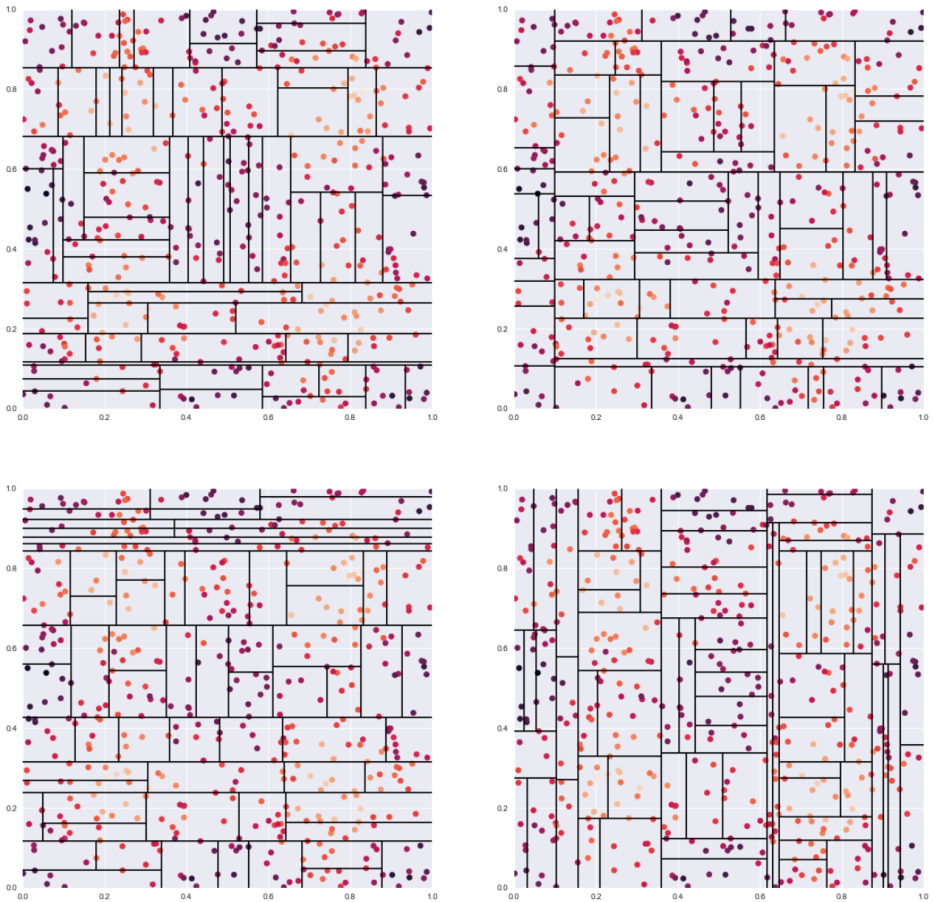


Data

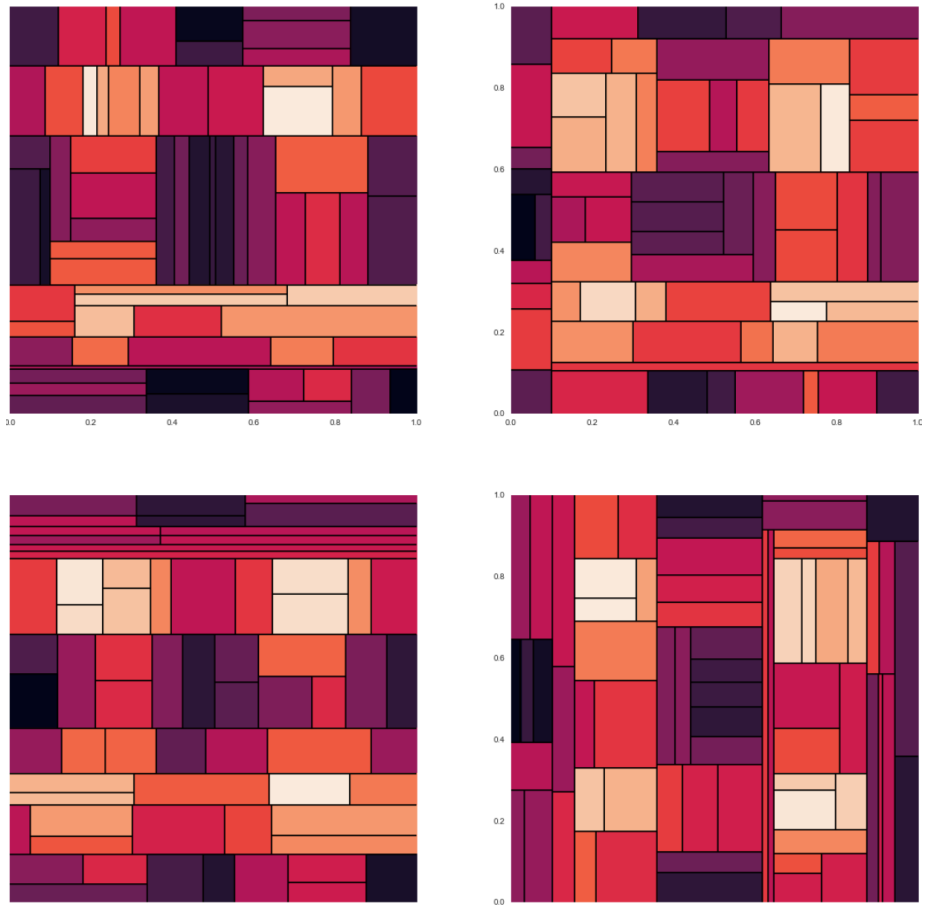


A decision tree

Introduction to ensemble: Random forest

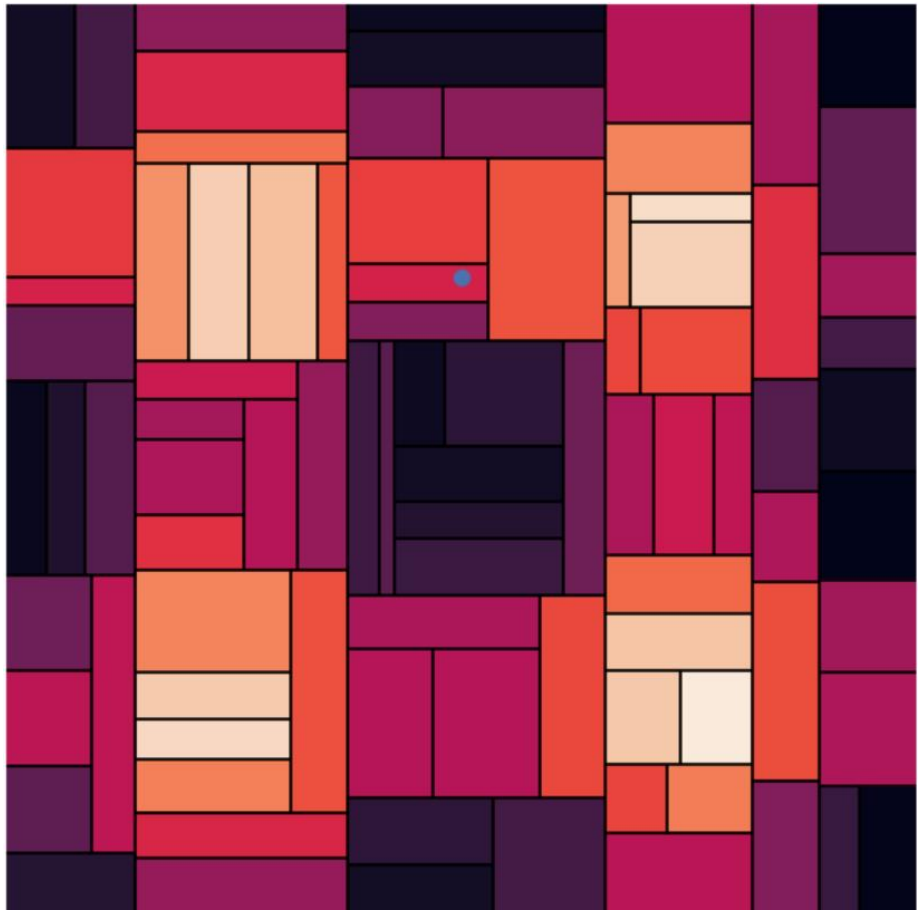


Many decision tree

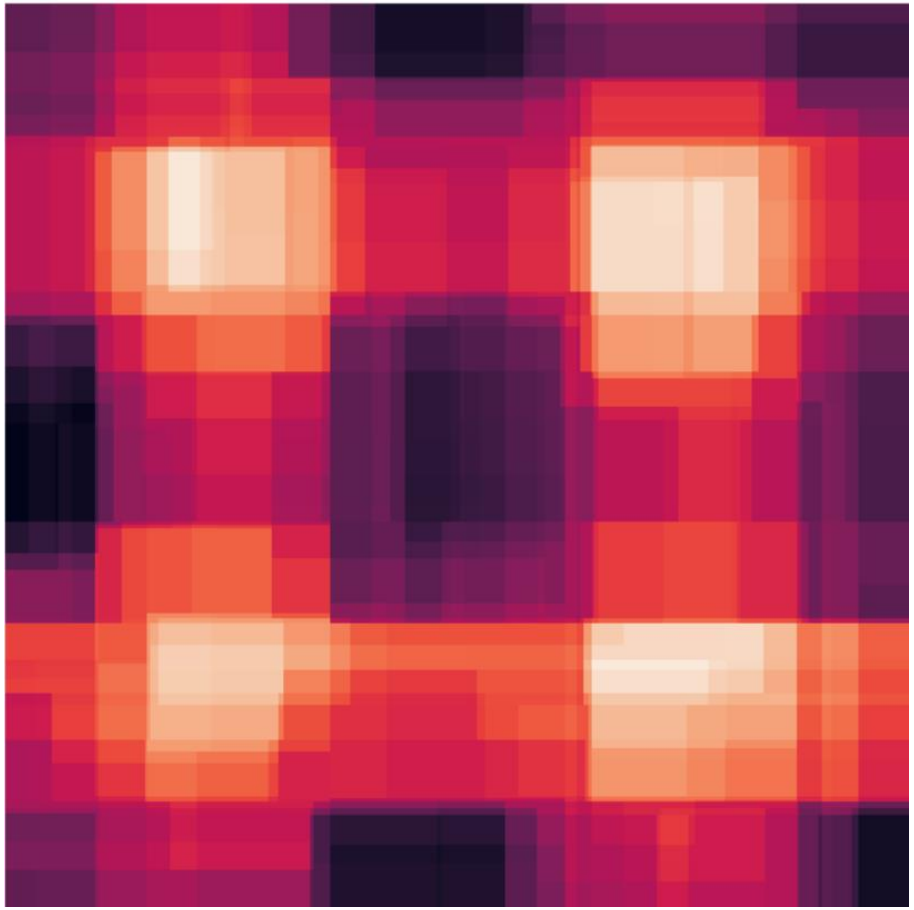


Predicting

Introduction to ensemble: Random forest



Decision tree



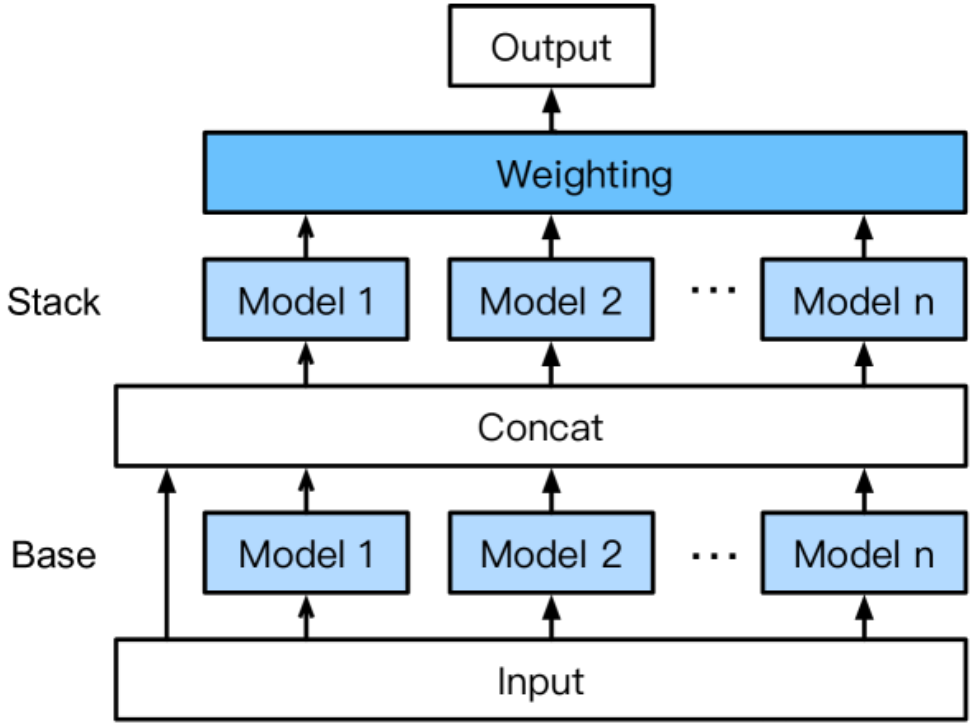
Random forest

Ensembling: Takeaways

- Many are better than few
- Models that learn differently can boost accuracy even if they are worse in isolation
- Hyperparameter tuning to find the best single model does not necessarily help find the best combination of models

Multi-layer stack ensembling

- Stack model uses predictions of every base model as extra features
- Layer L+1 stacker model uses layer L predictions as extra features
- Stacker is trained with held-out predictions of lower-layer models



AutoGluon and AutoML

To obtain the upcoming results:

- AutoGluon did not hyperparameter tune
- AutoGluon used the same hyperparameters for every problem
- AutoGluon used near default hyperparameters for all of its models
- AutoGluon trained using the same 3 lines of code for every problem

Pairwise comparisons: AutoML & Kaggle (2020)

Framework	Wins	Losses	Failures	Champion	Avg. Rank	Avg. Rescaled Loss	Avg. Time (min)
AutoGluon	-	-	**1**	**23**	**1.8438**	**0.1385**	201
H2O AutoML	4	26	8	2	3.1250	0.2447	220
TPOT	6	27	5	5	3.3750	0.2034	235
GCP-Tables	5	20	14	4	3.7500	0.3336	**195**
auto-sklearn	6	27	6	3	3.8125	0.3197	240
Auto-WEKA	4	28	6	1	5.0938	0.8001	244

AutoML Benchmark (39 datasets)

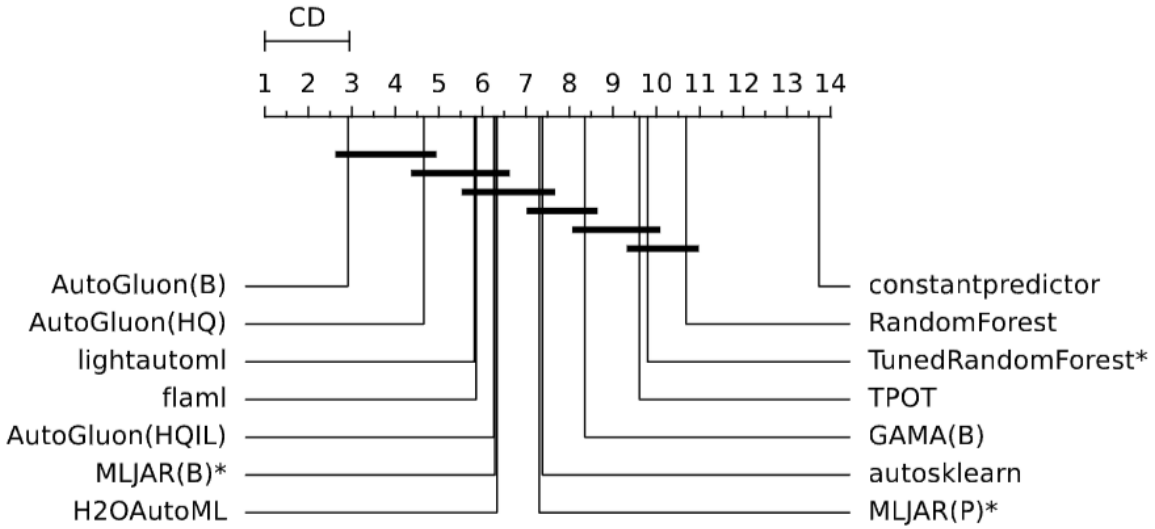
Framework	Wins	Losses	Failures	Champion	Avg. Rank	Avg. Percentile	Avg. Time (min)
AutoGluon	-	-	**0**	**7**	**1.7143**	**0.7041**	**202**
GCP-Tables	3	7	1	3	2.2857	0.6281	222
H2O AutoML	1	7	3	0	3.4286	0.5129	227
TPOT	1	9	1	0	3.7143	0.4711	380
auto-sklearn	3	8	**0**	1	3.8571	0.4819	240
Auto-WEKA	0	10	1	0	6.0000	0.2056	221

Kaggle Benchmark (11 datasets)

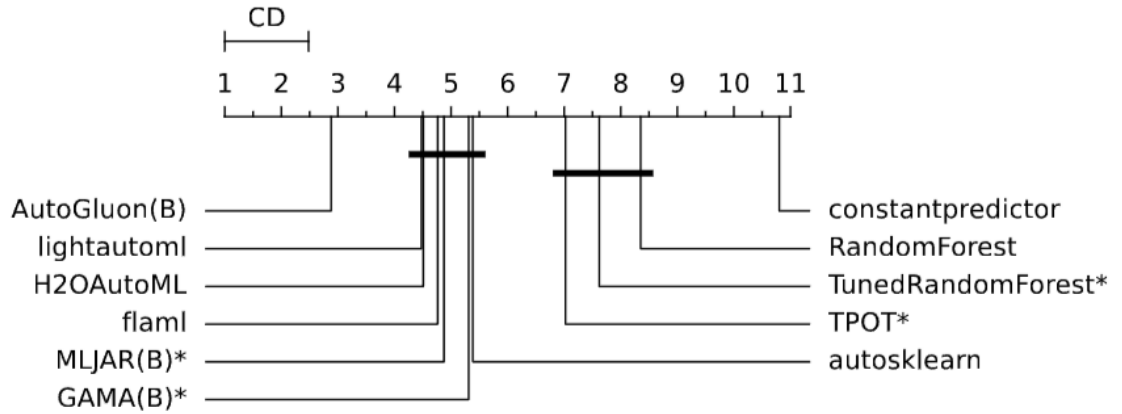
AutoGluon-Tabular: Robust and Accuate AutoML for Structured Data, arXiv, 2020

AutoMLBenchmark (2023)

- 104 classification & regression tasks, 10 folds, 4hours
- Average rank per framework



(g) All tasks, 1 hour



(h) All tasks, 4 hours

AMLB: an AutoML Benchmark, arXiv, 2023

Can AutoGluon win a live competition?

Kaggle

- 2nd out of 2742 (Mercari Price Suggestion – 2380 teams with $100,000 price)
- 2nd out of 172 (California House Prices)
- 20th out of 3537 (PetFinder Pawpularity Contest) – 2022

MachineHack

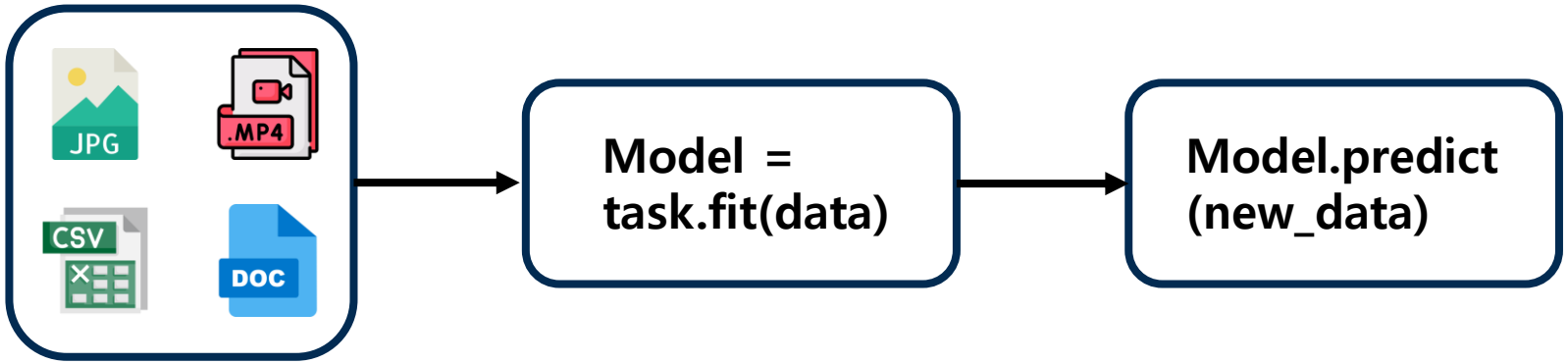
- 1st out of 1544 (Predict Data Scientist Salary in India)
- 1st out of 314 (Product Sentiment Classification)
- 2nd out of 3922 (Predict the Price of Books)

AutoGluon is easy to use

Step 1. Prepare your dataset

Step 2. Load the dataset for training ML

Step 3. Call fit() to get the ML model



AutoGluon is easy to use

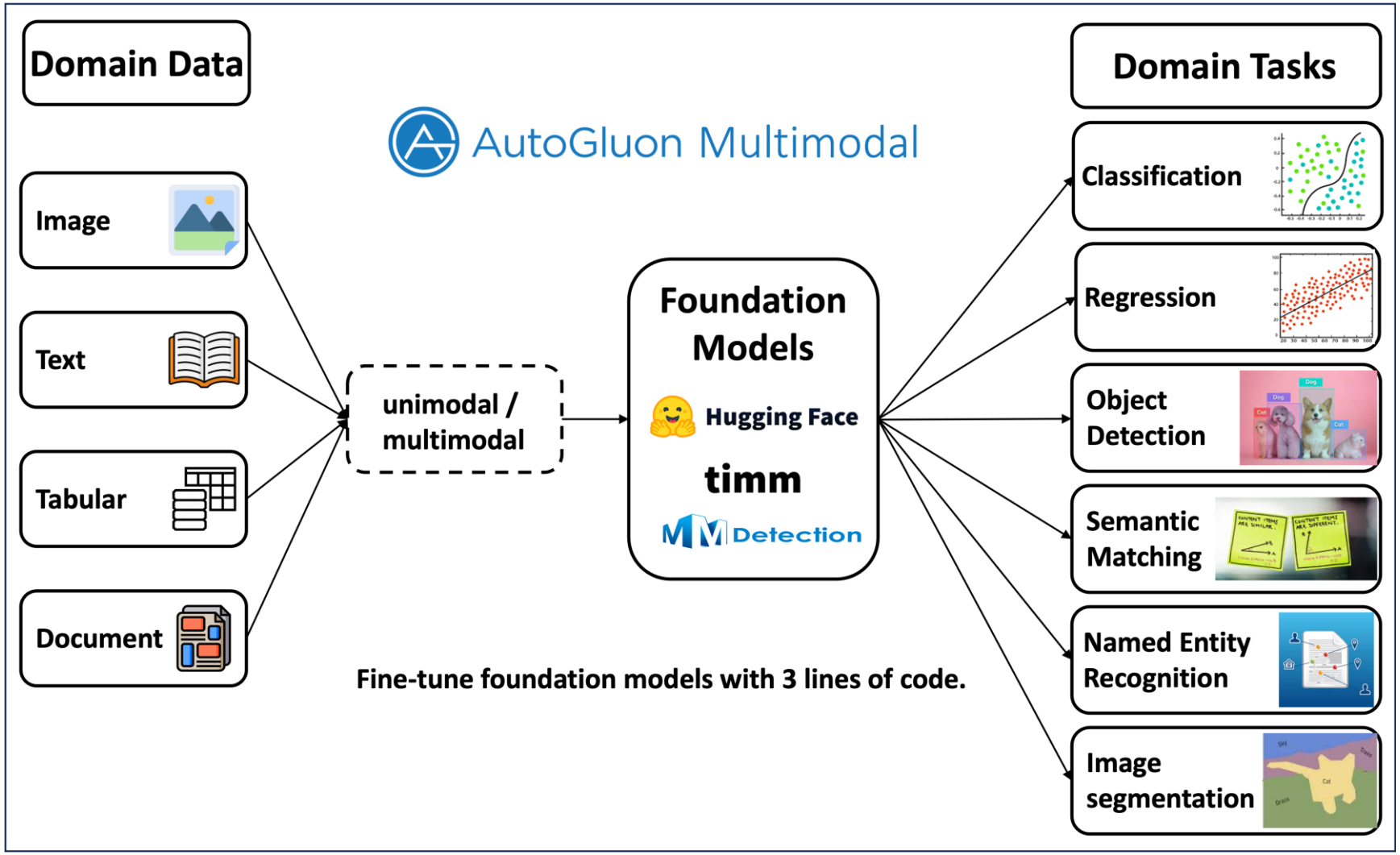
- **Only few lines of code to train and use a model**

```
from autogluon.tabular import TabularDataset, TabularPredictor

data_root = 'https://autogluon.s3.amazonaws.com/datasets/Inc/'
train_data = TabularDataset(data_root + 'train.csv')
test_data = TabularDataset(data_root + 'test.csv')

predictor = TabularPredictor(label='class').fit(train_data=train_data)
predictions = predictor.predict(test_data)
```

- **fit() automatically does the following:**

1. Preprocesses raw data (identifies the type of each feature)
2. Identifies what type of prediction problem this is (binary/multi-class classification or regression)
3. Splits data appropriately (e.g. training/validation sets, k-fold split)
4. Individually trains/tunes various models
5. Assembles ensemble that outperforms all the individual models

Built-in Prediction Tasks



서버 접속 및 설정

WiFi

- WiFi : SKKU_SEMINAR
- Seminar key : seminar240722

SKKU HPC Cluster Service

- 고성능의 HPC Cluster(A100)를 클라우드 컴퓨팅 형식으로 제공
- https://supercom.skku.edu/supercom/index.do

HPC Cluster GPU Server		DELL PowerEdge XE8545	10 EA	AMD EPYC 7763/2EA A100(80GB)Nvidia HMB2e NVLINK /4EA *상세 스펙 비공개

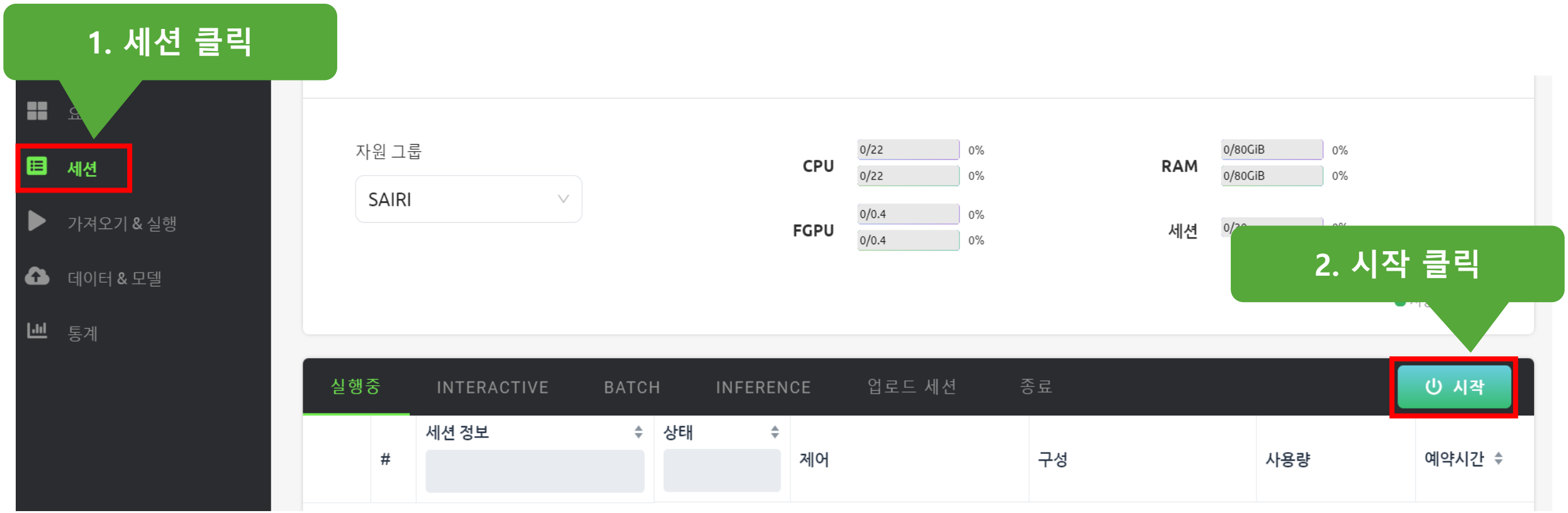
구분	내부 사용자	외부 사용자
1 GPU	1,000원	2,200원

* 각 금액은 1시간 기준 요금임. 사용량 계산시 15분 단위 절사
* 통화는 KRW 기준이며, VAT미포함

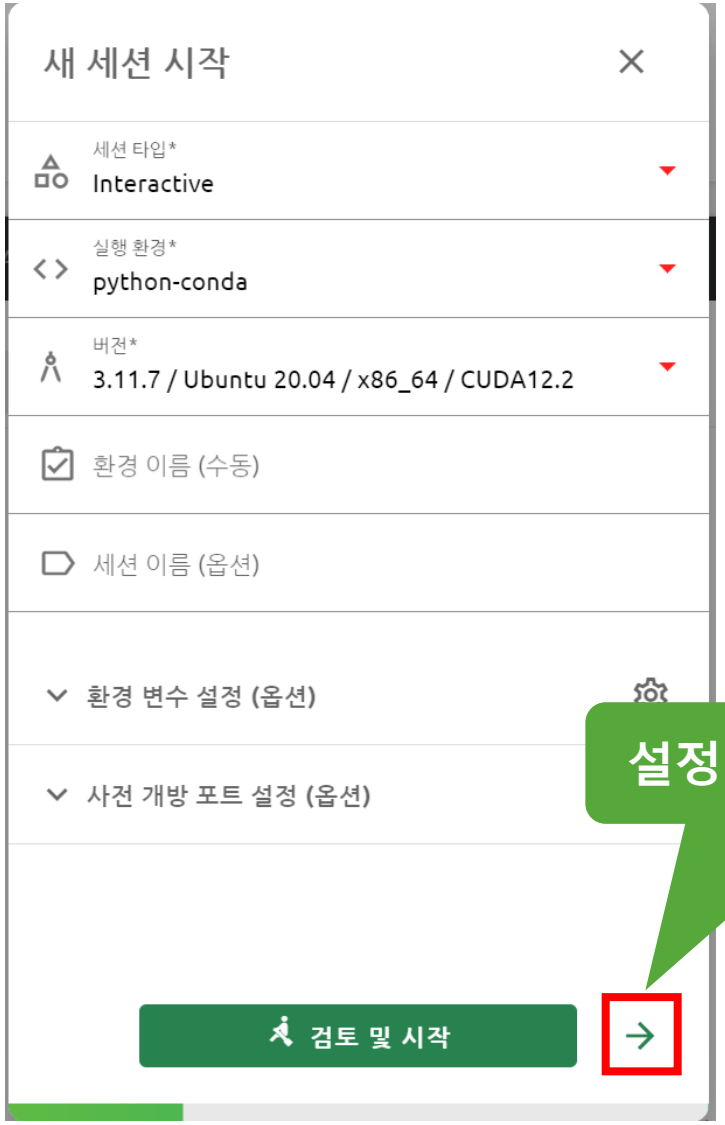
SKKU HPC Cluster Service

- https://spccluster.skku.edu/

SKKU HPC Cluster Service



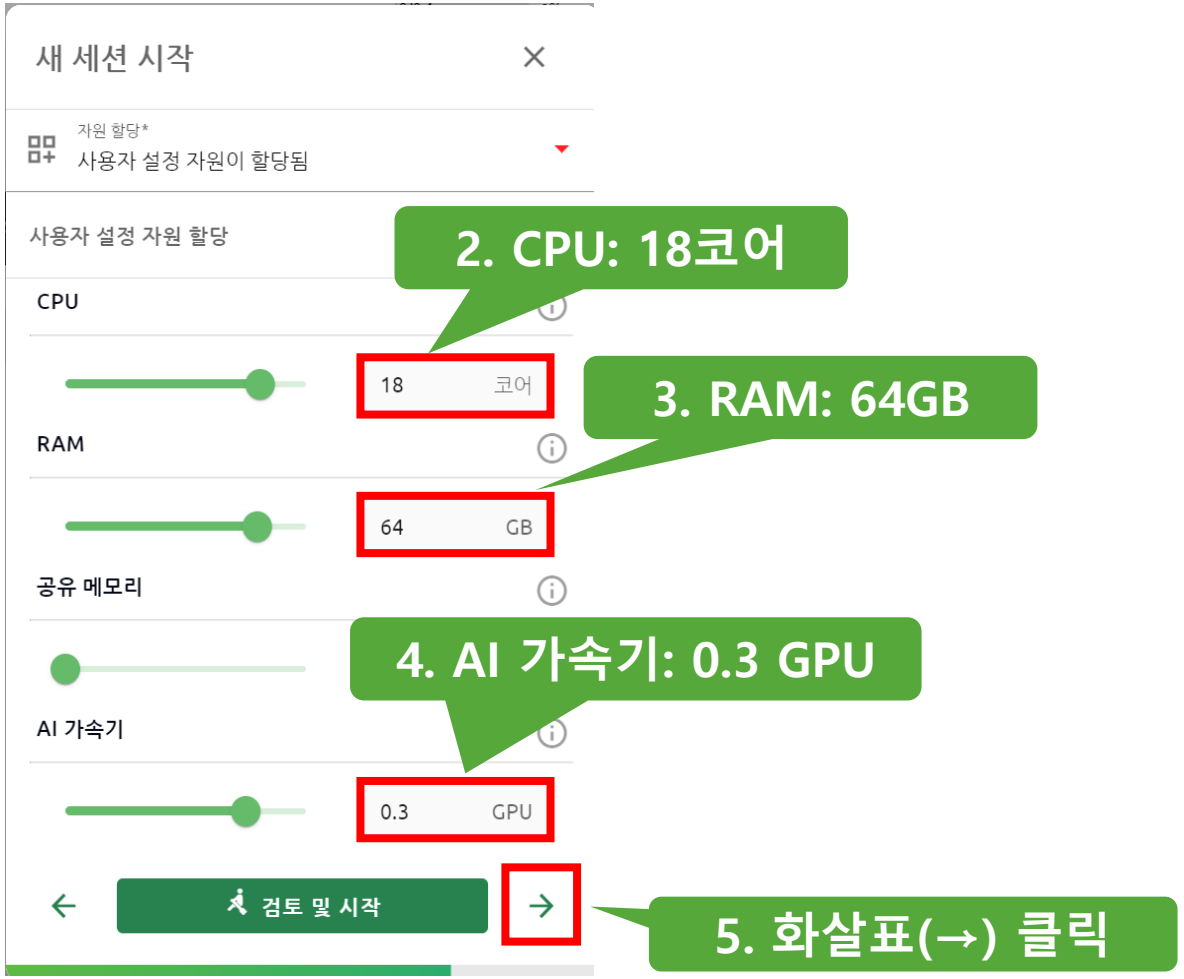
SKKU HPC Cluster Service



- 세션 타입: Interactive
- 실행 환경: python-conda
- 버전 3.11.7 / Ubuntu 20.04 / x86_64 / CUDA12.2

SKKU HPC Cluster Service

SKKU HPC Cluster Service

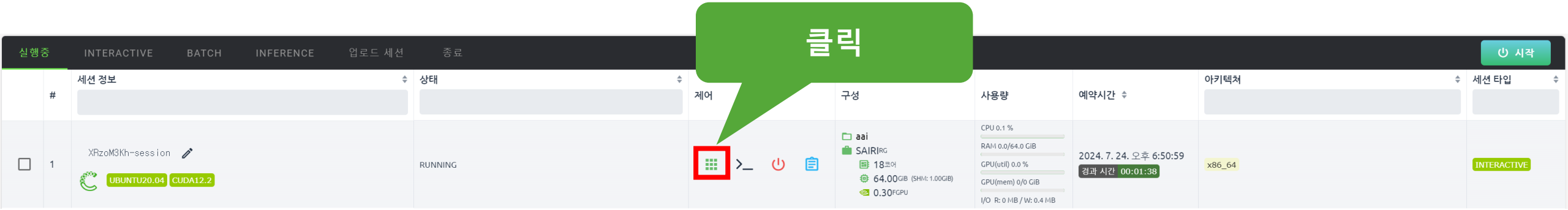


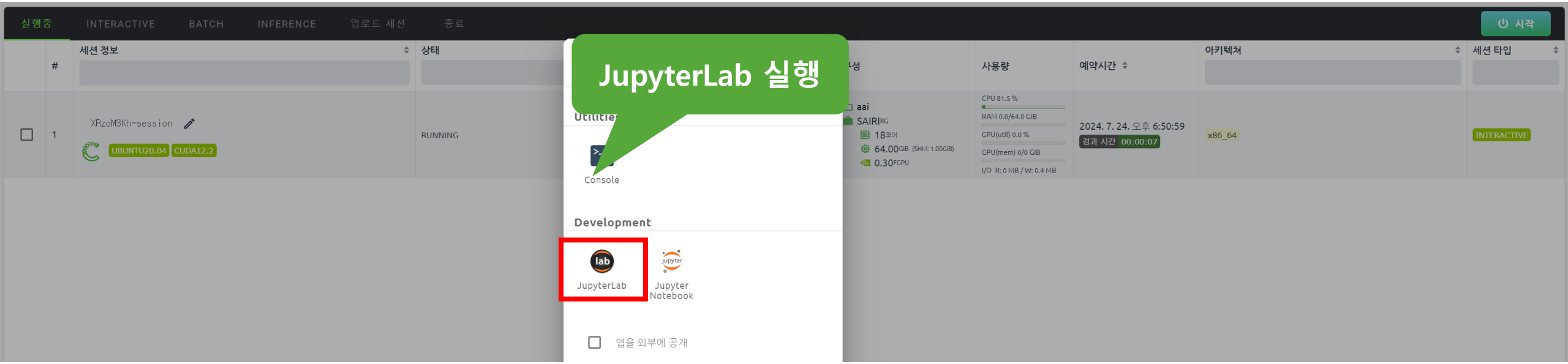
SKKU HPC Cluster Service

- 환경 정보
- 총 자원 할당량
- 마운트된 폴더

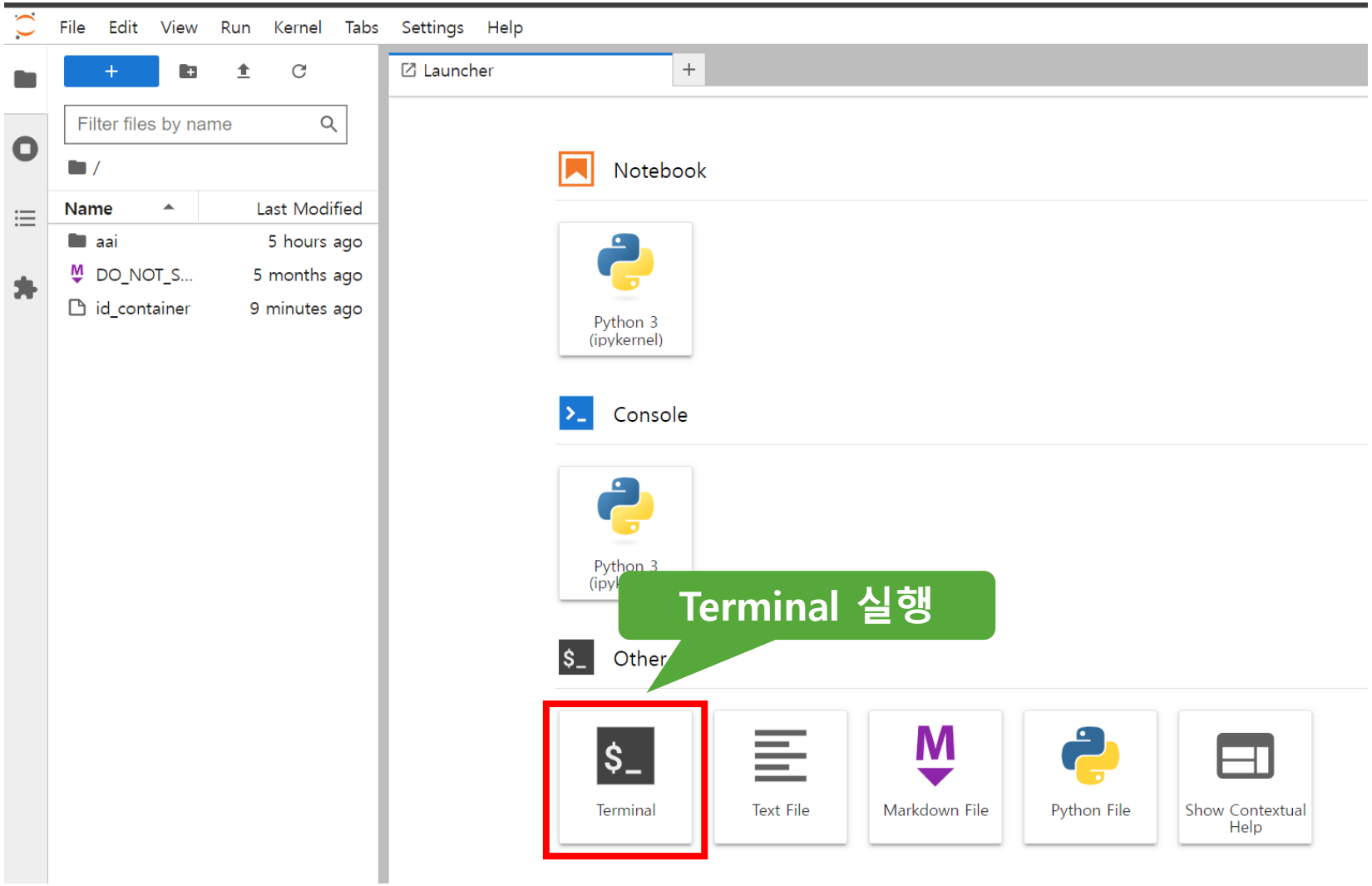
확인 후 시작 클릭

SKKU HPC Cluster Service





Environment setup

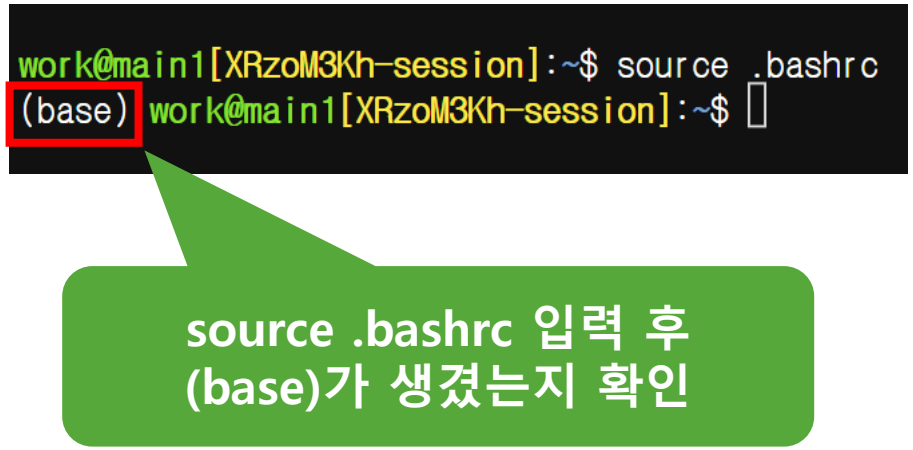


Environment setup

1. conda init 입력

```
work@main1[XRzoM3Kh-session]:~$ conda init
no change     /home/user/anaconda3/condabin/conda
no change     /home/user/anaconda3/bin/conda
no change     /home/user/anaconda3/bin/conda-env
no change     /home/user/anaconda3/bin/activate
no change     /home/user/anaconda3/bin/deactivate
no change     /home/user/anaconda3/etc/profile.d/conda.sh
no change     /home/user/anaconda3/etc/fish/conf.d/conda.fish
no change     /home/user/anaconda3/shell/condabin/Conda.psm1
no change     /home/user/anaconda3/shell/condabin/conda-hook.ps1
no change     /home/user/anaconda3/lib/python3.11/site-packages/xontrib/conda.xsh
no change     /home/user/anaconda3/etc/profile.d/conda.csh
modified      /home/work/.bashrc

==> For changes to take effect, close and re-open your current shell. <==
```

2. source .bashrc 입력



Environment setup

1. cd aai

```
(base) work@main1[XRzoM3Kh-session]:~$ ls
DO_NOT_STORE_PERSISTENT_FILES_HERE.md  aai  id_container
(base) work@main1[XRzoM3Kh-session]:~$ cd aai
(base) work@main1[XRzoM3Kh-session]:~/aai$
```

1. cd aai 입력 후 확인

2. ls

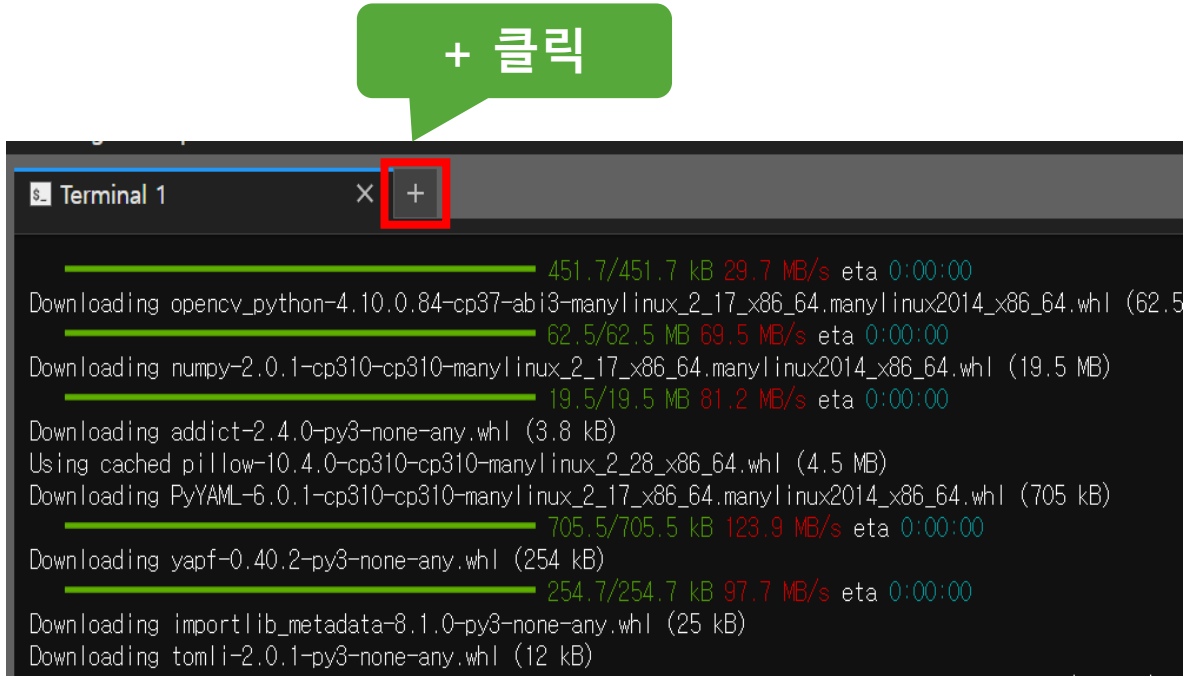
```
(base) work@main1[XRzoM3Kh-session]:~/aai$ ls
AutoGluon  HuggingFace  install_1.sh
```

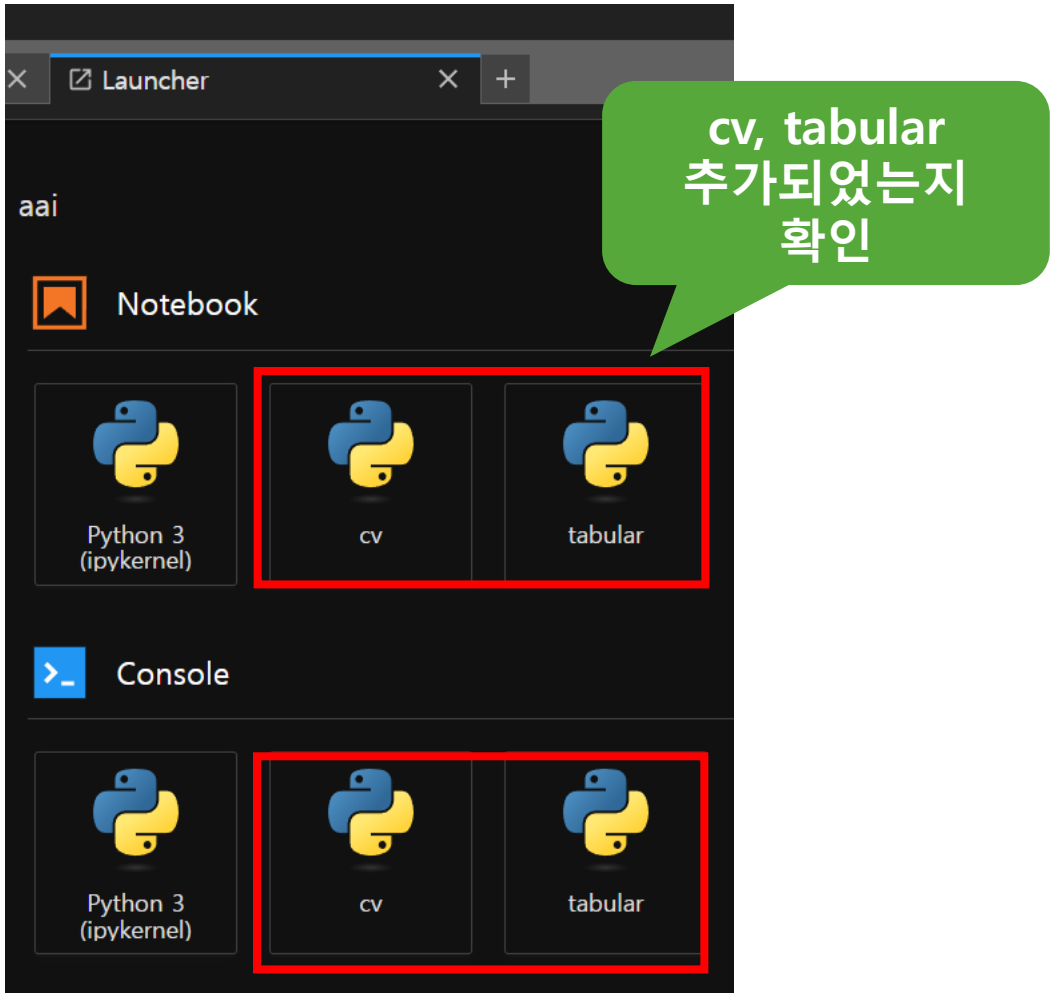
2. ls 입력 후 AutoGluon, HuggingFace, install_1.sh 확인

3. source install_1.sh

```
work@main1[BhQYto7r-session]:~/aai$ source install_1.sh
```

Environment setup

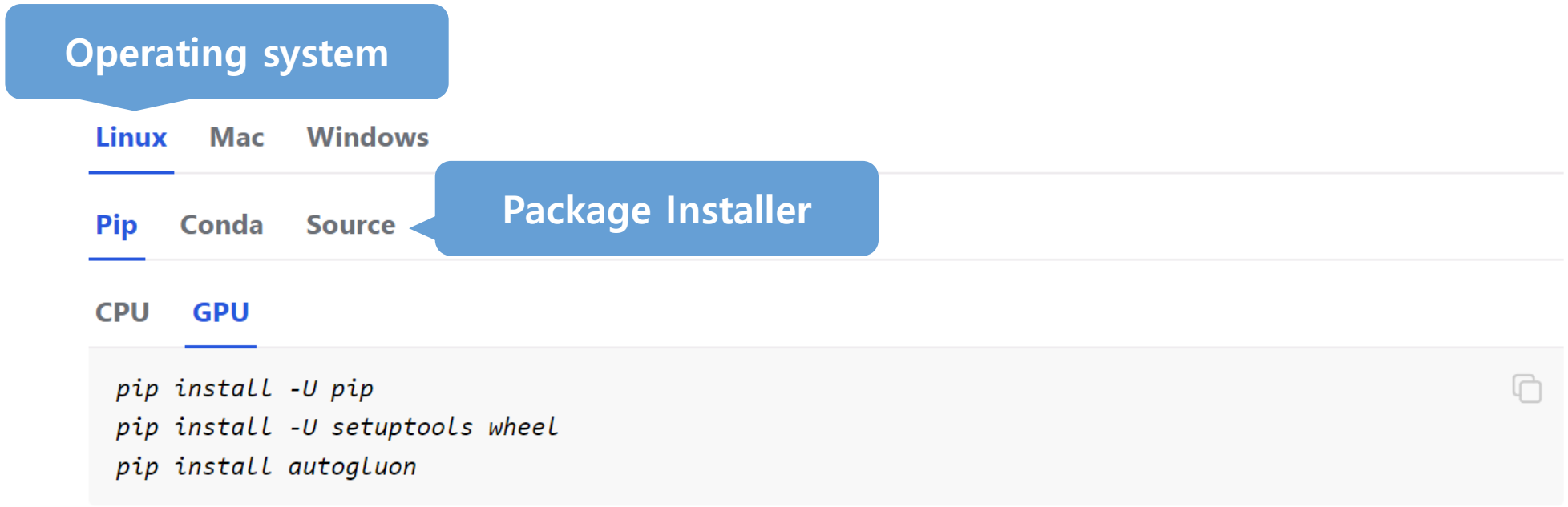




Tabular Classification

Installation

- https://auto.gluon.ai/stable/install.html



- The Conda install may have subtle differences in installed dependencies that could impact performance and stability, and we recommend trying pip if you run into issues with Conda.

Preparing Data

```
import pandas as pd

df_train = pd.read_csv('https://autogluon.s3.amazonaws.com/datasets/titanic/train.csv')
df_test = pd.read_csv('https://autogluon.s3.amazonaws.com/datasets/titanic/test.csv')
target_col = 'Survived'
```

- Pandas: data analysis and data manipulation library for the Python
- pd.read_csv: Read a comma-separated values (csv) file into DataFrame

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Preparing Data

```
[5]  1  df_train.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

- DataFrame.head: Returns the first n rows for the object based on position.

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.head.html

Automated Dataset Overview

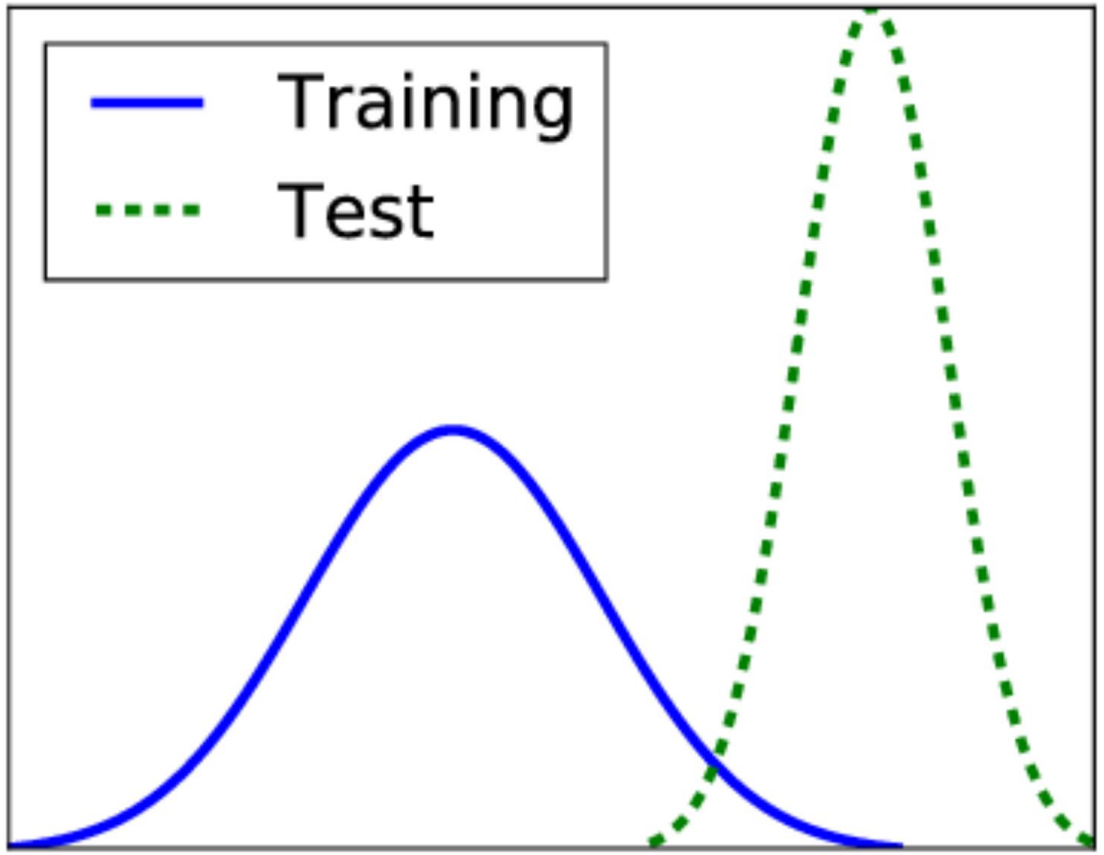
```
1  import autogluon.eda.auto as auto
2
3  auto.dataset_overview(train_data=df_train, test_data=df_test, label='Survived')
```

train_data **dataset summary**

	count	unique	top	freq	mean	std	min	25%	50%	75%	max	dtypes	missing_count	missing_ratio	raw_type	variable_type	special_types
Age	714	88			29.699118	14.526497	0.42	20.125	28.0	38.0	80.0	float64	177	0.198653	float	numeric	
Cabin	204	147	B96 B98	4								object	687	0.771044	object	category	
Embarked	889	3	S	644								object	2	0.002245	object	category	
Fare	891	248			32.204208	49.693429	0.0	7.9104	14.4542	31.0	512.3292	float64			float	numeric	
Name	891	891	Braund, Mr. Owen Harris	1								object			object	category	text
Parch	891	7			0.381594	0.806057	0.0	0.0	0.0	0.0	6.0	int64			int	category	
PassengerId	891	891			446.0	257.353842	1.0	223.5	446.0	668.5	891.0	int64			int	numeric	
Pclass	891	3			2.308642	0.836071	1.0	2.0	3.0	3.0	3.0	int64			int	category	
Sex	891	2	male	577								object			object	category	
SibSp	891	7			0.523008	1.102743	0.0	0.0	0.0	1.0	8.0	int64			int	category	
Survived	891	2			0.383838	0.486592	0.0	0.0	0.0	1.0	1.0	int64			int	category	
Ticket	891	681	347082	7								object			object	category	

Covariate Shift Analysis

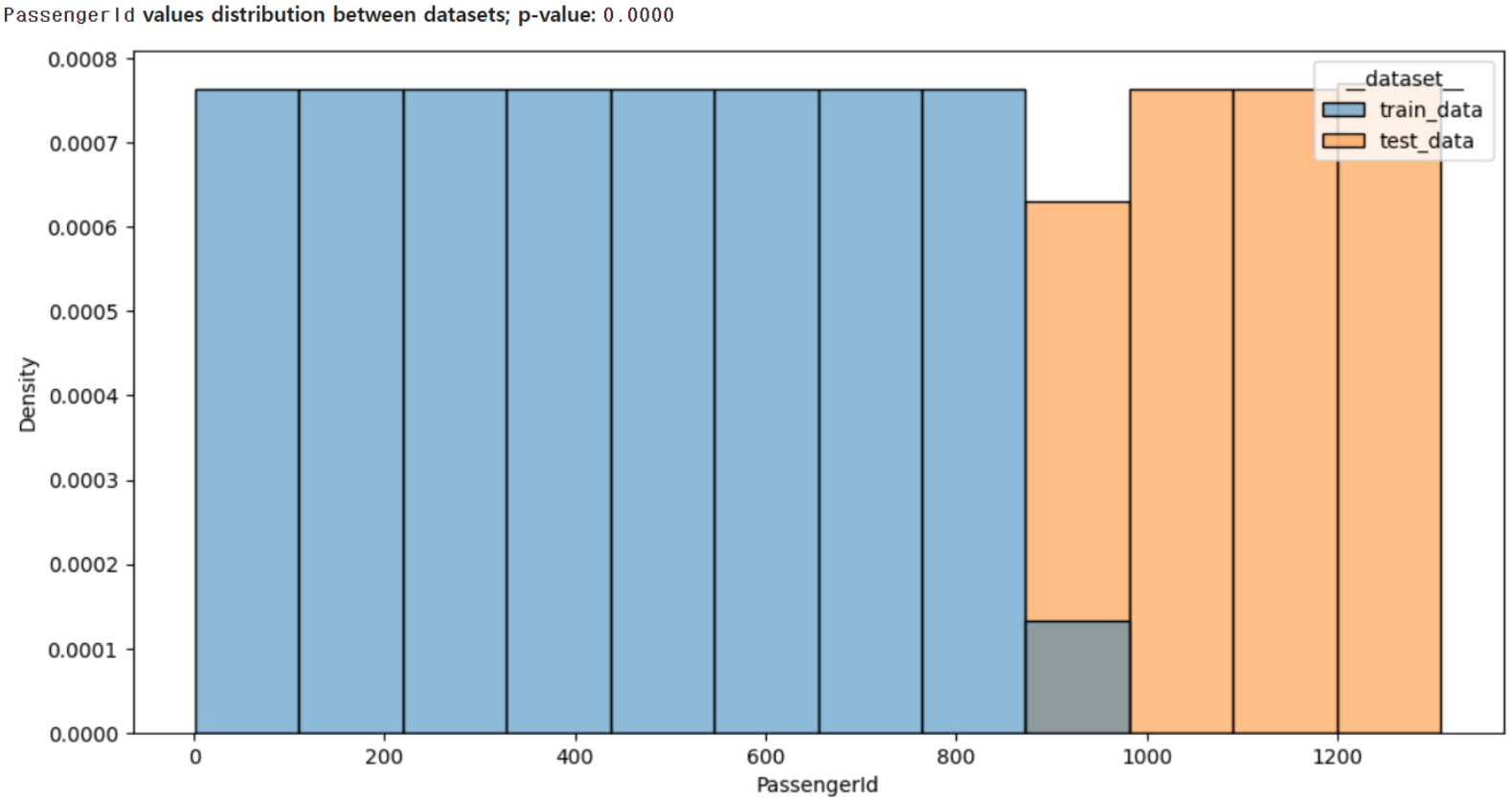
- Covariate shift is a phenomenon in machine learning where the distribution of the independent variables in the training and testing data is different.



Covariate Shift Analysis

```
import autogluon.eda.auto as auto

auto.covariate_shift_detection(train_data=df_train, test_data=df_test, label='Survived')
```

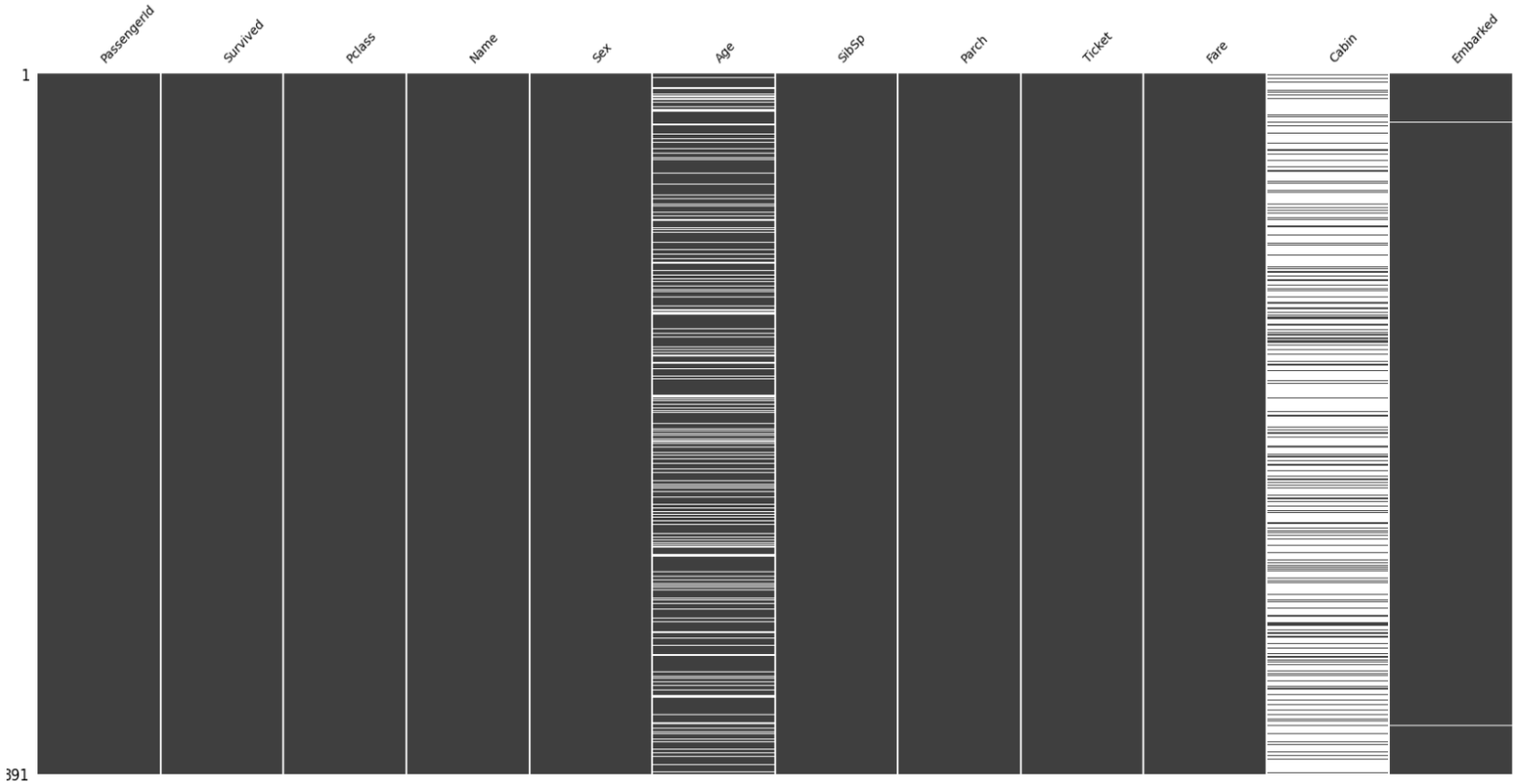
```
df_train = df_train.drop(columns='PassengerId')
df_test = df_test.drop(columns='PassengerId')
```



https://auto.gluon.ai/0.7.0/tutorials/eda/eda-auto-covariate-shift.html

Missing values analysis

```
import autogluon.eda.auto as auto
auto.missing_values_analysis(train_data=df_train)
```



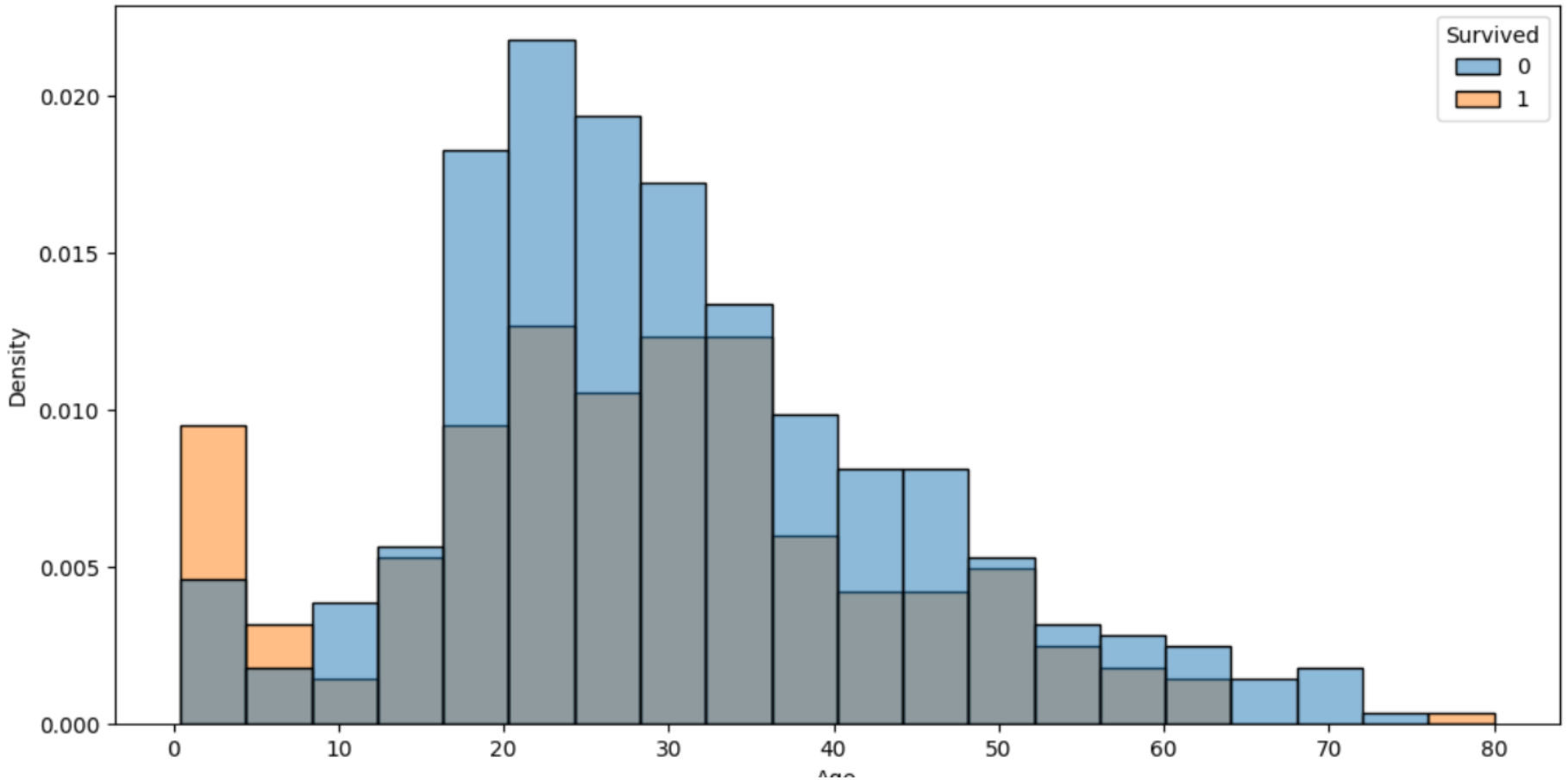
Missing Values Analysis

	missing_count	missing_ratio
Age	177	0.198653
Cabin	687	0.771044
Embarked	2	0.002245

https://auto.gluon.ai/0.7.0/tutorials/eda/eda-auto-analyze-interaction.html

Feature Interaction Charting

```
auto.analyze_interaction(x='Age', hue='Survived', train_data=df_train, test_data=df_test)
```



https://auto.gluon.ai/0.7.0/tutorials/eda/eda-auto-analyze-interaction.html

Training

```
from autogluon.tabular import TabularPredictor

predictor = TabularPredictor(label='Survived').fit(df_train)
```

```
class autogluon.tabular.TabularPredictor(label: str, problem_type: str | None = None,
    eval_metric: str | Scorer | None = None, path: str | None = None, verbosity: int = 2,
    log_to_file: bool = False, log_file_path: str = 'auto', sample_weight: str | None = None,
    weight_evaluation: bool = False, groups: str | None = None, **kwargs)          [source]
```

AutoGluon TabularPredictor predicts values in a column of a tabular dataset (classification or regression).

- label(str) – Name of the column that contains the target variable to predict.
- problem_type – Type of prediction problem, i.e. is this a binary/multiclass classification or regression problem.
- eval_metric – Metric by which predictions will be ultimately evaluated on test data.

https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.html

Training

```
from autogluon.tabular import TabularPredictor

predictor = TabularPredictor(label='Survived').fit(df_train)
```

```
TabularPredictor.fit(train_data, tuning_data=None, time_limit: float = None, presets:
    List[str] | str = None, hyperparameters: dict | str = None, feature_metadata='infer',
    infer_limit: float = None, infer_limit_batch_size: int = None, fit_weighted_ensemble: bool
    = True, fit_full_last_level_weighted_ensemble: bool = True,
    full_weighted_ensemble_additionally: bool = False, dynamic_stacking: bool | str = False,
    calibrate_decision_threshold: bool | str = False, num_cpus: int | str = 'auto', num_gpus:
    int | str = 'auto', **kwargs) → TabularPredictor                              [source]
```

Fit models to predict a column of a data table (label) based on the other columns (features).

- presets – List of preset configurations for various arguments in fit(). Can significantly impact predictive accuracy, memory-footprint, and inference latency of trained models, and various other properties of the returned predictor.
- tuning_data – Another dataset containing validation data reserved for tuning process.
- hyperparameters – Determines the hyperparameters used by the models.

Monitoring

```
predictor.leaderboard()
```

Validation score

Inference time

Training time

	model	score_val	pred_time_val	fit_time	pred_time_val_marginal	fit_time_marginal	stack_level	can_infer	fit_order
0	WeightedEnsemble_L2	0.854749	0.026934	8.660037	0.001510	0.907350	2	True	13
1	NeuralNetTorch	0.843575	0.018771	7.339141	0.018771	7.339141	1	True	11
2	NeuralNetFastAI	0.837989	0.034112	2.585567	0.034112	2.585567	1	True	10
3	CatBoost	0.832402	0.010626	1.953277	0.010626	1.953277	1	True	7
4	LightGBM	0.826816	0.006652	0.413546	0.006652	0.413546	1	True	4
5	LightGBMLarge	0.821229	0.006938	0.698423	0.006938	0.698423	1	True	12

Output summary of information about models produced during fit() as a pd.DataFrame.

https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.leaderboard.html

Monitoring

```
predictor.feature_importance(df_train)
```

Importance score

Feature

	importance	stddev	p_value	n	p99_high	p99_low
Sex	0.142536	0.013351	0.000009	5	0.170026	0.115047
Ticket	0.068013	0.006762	0.000012	5	0.081937	0.054090
Name	0.065993	0.006116	0.000009	5	0.078587	0.053399
Pclass	0.062177	0.007880	0.000030	5	0.078403	0.045952
Age	0.042873	0.003310	0.000004	5	0.049689	0.036057
SibSp	0.034343	0.005418	0.000072	5	0.045498	0.023189
Cabin	0.020202	0.003272	0.000080	5	0.026939	0.013465
Embarked	0.018855	0.002905	0.000066	5	0.024837	0.012874
Parch	0.017733	0.002159	0.000026	5	0.022178	0.013288
Fare	0.014141	0.002039	0.000050	5	0.018339	0.009943

Calculates feature importance scores for the given model via permutation importance.

https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.feature_importance.html

Predicting

```
# Predict for each row
predictor.predict(df_test)
# Return the class probabilities for classification
predictor.predict_proba(df_test)
# Evaluate various metrics, it needs test_data to have the label column
predictor.evaluate(df_test)
```

- Use trained models to produce predictions of *label* column values for new data.

```
Evaluation: accuracy on test data: 0.7966507177033493
Evaluations on test data:
{
    "accuracy": 0.7966507177033493,
    "balanced_accuracy": 0.7794303797468354,
    "mcc": 0.5641241988480588,
    "roc_auc": 0.8269717624148004,
    "f1": 0.7249190938511327,
    "precision": 0.7417218543046358,
    "recall": 0.7088607594936709
}
{'accuracy': 0.7966507177033493,
 'balanced_accuracy': 0.7794303797468354,
 'mcc': 0.5641241988480588,
 'roc_auc': 0.8269717624148004,
 'f1': 0.7249190938511327,
 'precision': 0.7417218543046358,
 'recall': 0.7088607594936709}
```

https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.predict.html

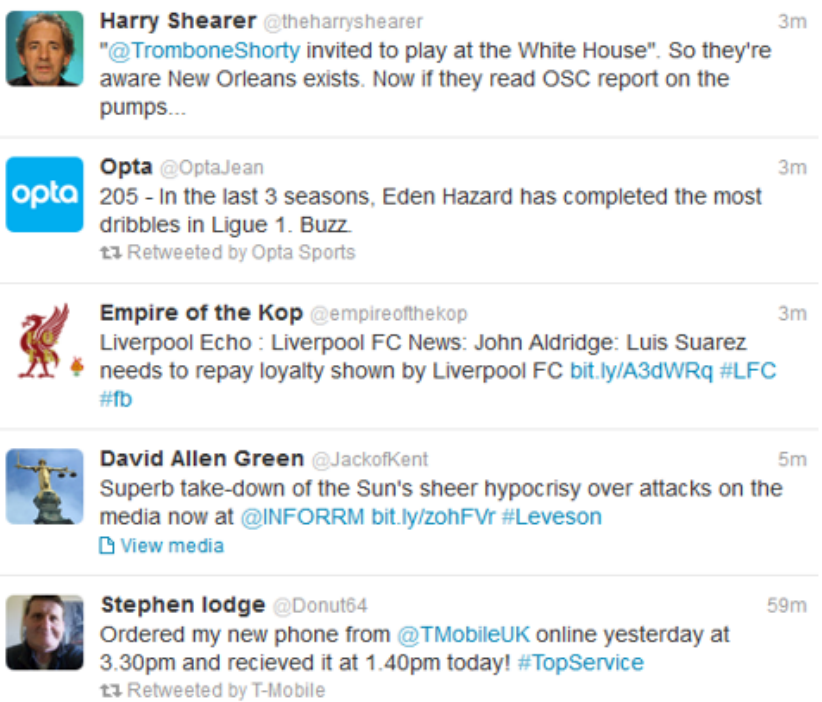
Presets

```
time_limit = 60
metric = 'roc_auc'
predictor = TabularPredictor(target_col, eval_metric=metric).fit(df_train, time_limit=time_limit, presets='best_quality')
predictor.leaderboard(df_test, silent=True)
```

Preset	Model Quality	Use Cases	Fit Time (Ideal)	Inference Time (Relative to medium_quality)	Disk Usage
best_quality	State-of-the-art (SOTA), much better than high_quality	When accuracy is what matters	16x+	32x+	16x+
high_quality	Better than good_quality	When a very powerful, portable solution with fast inference is required: Large-scale batch inference	16x	4x	2x
good_quality	Significantly better than medium_quality	When a powerful, highly portable solution with very fast inference is required: Billion-scale batch inference, sub-100ms online-inference, edge-devices	16x	2x	0.1x
medium_quality	Competitive with other top AutoML Frameworks	Initial prototyping, establishing a performance baseline	1x	1x	1x

https://auto.gluon.ai/stable/api/autogluon.tabular.TabularPredictor.predict.html

Ranking social media news feed



The news feed is a list that allows users to follow updates about individuals from their social network. The feed is typically displayed in chronological order.

Today, social media users are overwhelmed by a large number of posts in their news feed. Moreover, most posts are irrelevant. Ranking news feed posts by relevance has been proposed to help users catch up with the content they may find interesting.

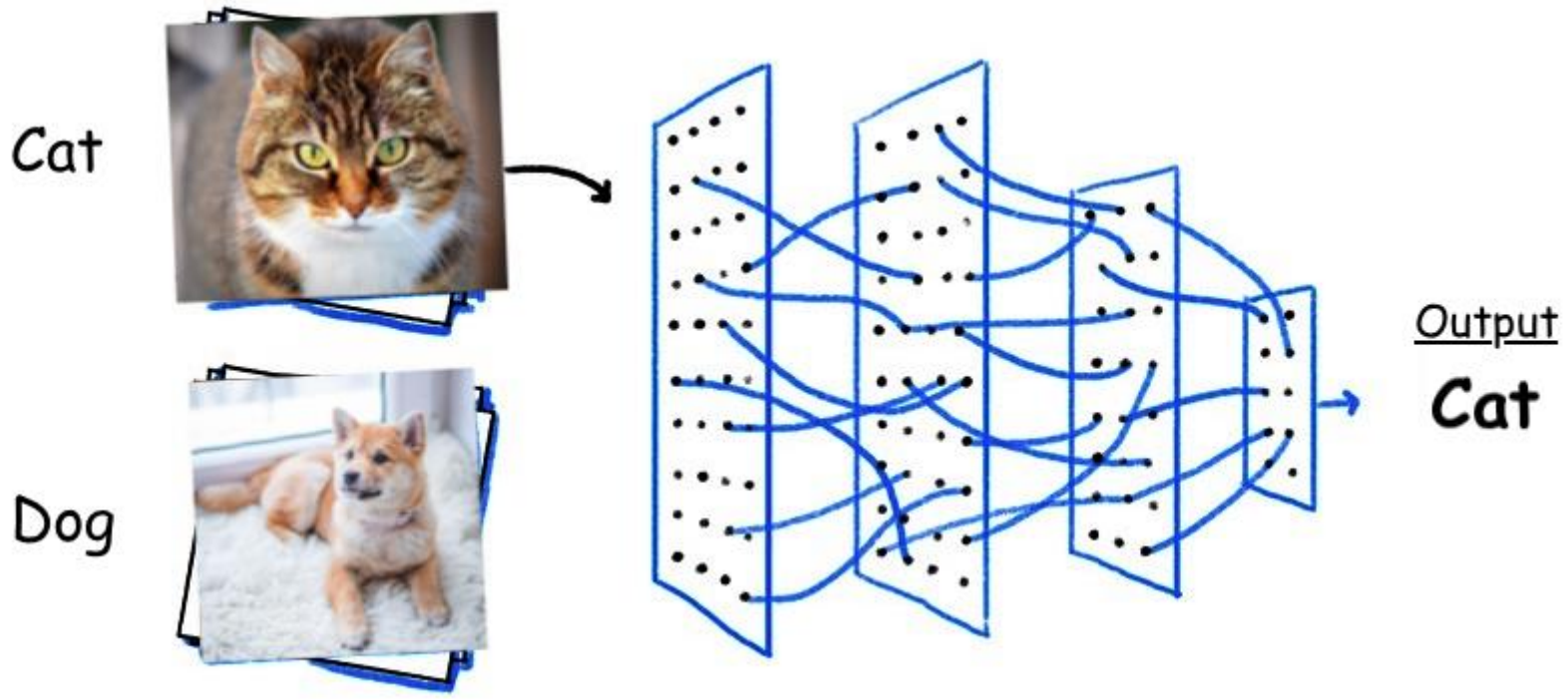
https://github.com/SamBelkacem/Ranking-social-media-news-feed

Ranking social media news feed

Features that may influence relevance		Type	N°
Relevance of the content of t, its hashtags, and mentions to u	Relevance of the keywords of t to u	Int	*f1*
	Relevance of the hashtags of t to u	Int	*f2*
	Presence of u in the mentions of t	Bool	*f3*
Social tie strength between u and u'	Interaction rate of u with tweets of u'	Float	*f4*
	Number of times u mentioned u'	Int	*f5*
Authority of u'	Followers count / Followings count	Int	*f6*
	Seniority in years	Int	*f7*
	Listed (group) count	Int	*f8*
Quality of t	Length (# characters)	Int	*f9*
	Presence of hashtags	Bool	*f10*
	Presence of a URL	Bool	*f11*
	Presence of an image or a video	Bool	*f12*
	Popularity (# retweets, replies, likes)	Int	*f13*

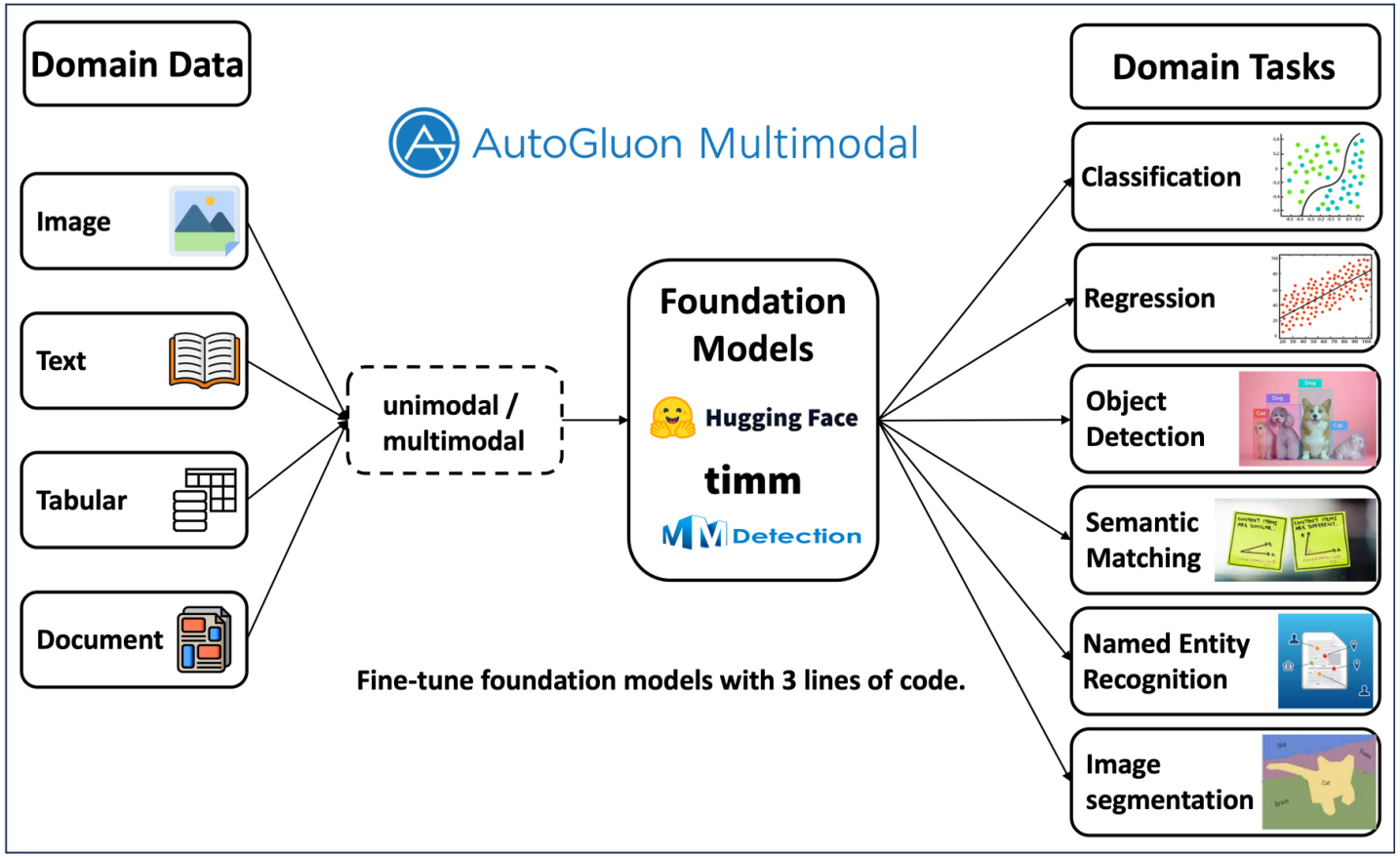
Image Classification

Image Classification



- **Image Classification** is a fundamental task in vision recognition that aims to understand and categorize an image as a whole under a specific label.

Multimodal



- AutoMM seamlessly integrates with popular model zoos such as HuggingFace Transformers, TIMM, and MMDetection, accommodating a diverse range of data modalities, including image, text, tabular, and document data, whether used individually or in combination.

Multimodal Predictor

```
class autogluon.multimodal.MultiModalPredictor(label: str | None = None, problem_type: str |
    None = None, query: str | List[str] | None = None, response: str | List[str] | None =
    None, match_label: int | str | None = None, presets: str | None = None, eval_metric: str |
    Scorer | None = None, hyperparameters: dict | None = None, path: str | None = None,
    verbosity: int | None = 2, num_classes: int | None = None, classes: list | None = None,
    warn_if_exist: bool | None = True, enable_progress_bar: bool | None = None, pretrained:
    bool | None = True, validation_metric: str | None = None, sample_data_path: str | None =
    None) ¶                                                                          [source]
```

- lable – Name of one pd.DataFrame column that contains the target variable to predict.
- problem_type –
  'binary': Binary classification
  'multiclass': Multi-class classification
  'regression': Regression
  'classification': Classification problems include 'binary' and 'multiclass' classification

Multimodal Predictor

```
class autogluon.multimodal.MultiModalPredictor(label: str | None = None, problem_type: str |
    None = None, query: str | List[str] | None = None, response: str | List[str] | None =
    None, match_label: int | str | None = None, presets: str | None = None, eval_metric: str |
    Scorer | None = None, hyperparameters: dict | None = None, path: str | None = None,
    verbosity: int | None = 2, num_classes: int | None = None, classes: list | None = None,
    warn_if_exist: bool | None = True, enable_progress_bar: bool | None = None, pretrained:
    bool | None = True, validation_metric: str | None = None, sample_data_path: str | None =
    None) ¶                                                                          [source]
```

- problem_type –
  'object_detection': Object detection
  'ner' or 'named_entity_recognition': Named entity extraction
  'text_similarity': Text-text semantic matching
  'image_similarity': Image-image semantic matching
  'image_text_similarity': Text-image semantic matching
  'feature_extraction': Extracting feature (only support inference)
  'zero_shot_image_classification': Zero-shot image classification (only support inference)
  'few_shot_classification': Few-shot classification for image or text data
  'semantic_segmentation': Semantic segmentation with Segment Anything Model

https://auto.gluon.ai/stable/api/autogluon.multimodal.MultiModalPredictor.html

Multimodal Predictor

```
class autogluon.multimodal.MultiModalPredictor(label: str | None = None, problem_type: str |
    None = None, query: str | List[str] | None = None, response: str | List[str] | None =
    None, match_label: int | str | None = None, presets: str | None = None, eval_metric: str |
    Scorer | None = None, hyperparameters: dict | None = None, path: str | None = None,
    verbosity: int | None = 2, num_classes: int | None = None, classes: list | None = None,
    warn_if_exist: bool | None = True, enable_progress_bar: bool | None = None, pretrained:
    bool | None = True, validation_metric: str | None = None, sample_data_path: str | None =
    None) ¶                                                                          [source]
```

- presets – Presets regarding model quality, 'best_quality', 'high_quality' (default), and 'medium_quality'.

- eval_metric – Evaluation metric name.

https://auto.gluon.ai/stable/api/autogluon.multimodal.MultiModalPredictor.html

Shopee-IET dataset

Shopee-IET dataset

- Each image in this data depicts a clothing item and the corresponding label specifies its clothing category.
- Subset of the data contains the following possible labels: BabyPants, BabyShirt, womencasualshoes, womenchiffontop

Shopee-IET dataset

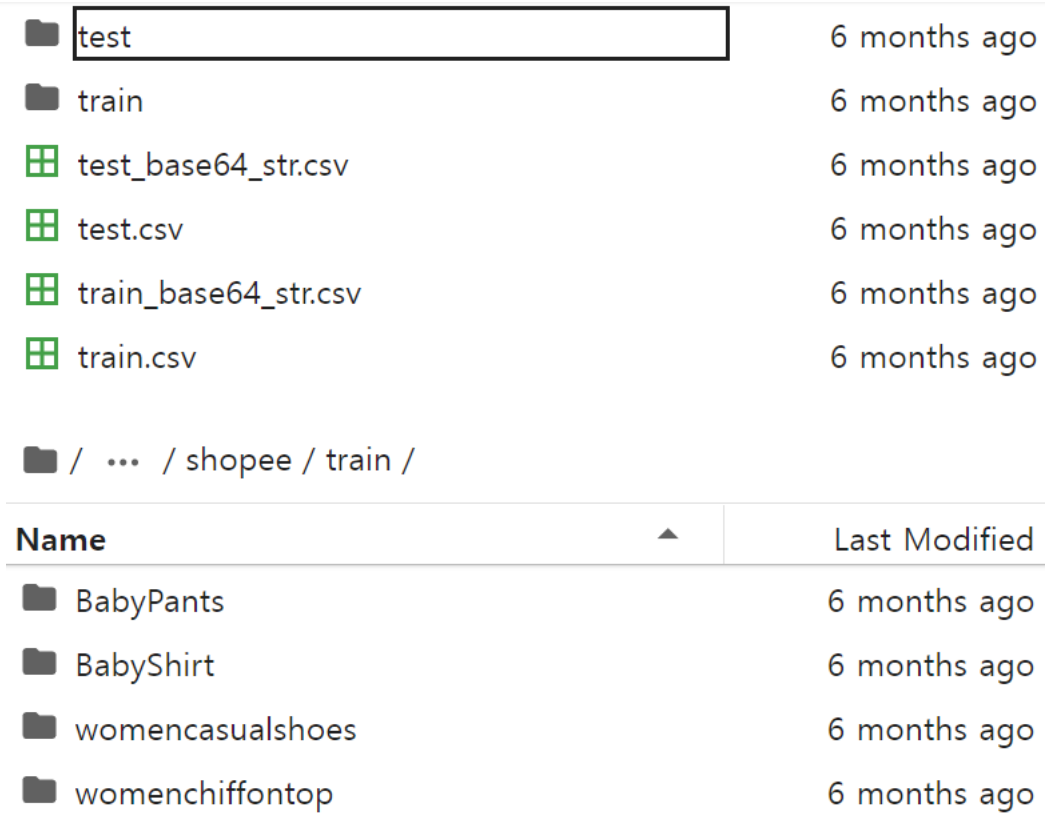


	image	label
1	train/BabyPants/BabyPants_1009.jpg	0
2	train/BabyPants/BabyPants_103.jpg	0
3	train/BabyPants/BabyPants_1038.jpg	0
4	train/BabyPants/BabyPants_1041.jpg	0
5	train/BabyPants/BabyPants_1042.jpg	0
6	train/BabyPants/BabyPants_1047.jpg	0
7	train/BabyPants/BabyPants_1049.jpg	0
8	train/BabyPants/BabyPants_1052.jpg	0
9	train/BabyPants/BabyPants_1055.jpg	0
10	train/BabyPants/BabyPants_1063.jpg	0
11	train/BabyPants/BabyPants_107.jpg	0
12	train/BabyPants/BabyPants_1070.jpg	0
13	train/BabyPants/BabyPants_1071.jpg	0
14	train/BabyPants/BabyPants_1078.jpg	0
15	train/BabyPants/BabyPants_1079.jpg	0
16	train/BabyPants/BabyPants_1095.jpg	0
17	train/BabyPants/BabyPants_1097.jpg	0

Shopee-IET dataset

```
import pandas as pd

train_data_path = pd.read_csv('./shopee/train.csv')
test_data_path = pd.read_csv('./shopee/test.csv')
```

```
from autogluon.multimodal import MultiModalPredictor

predictor = MultiModalPredictor(label='label')
predictor.fit(train_data=train_data_path)
```

Use AutoMM to Fit Models

```python
from autogluon.multimodal import MultiModalPredictor

predictor = MultiModalPredictor(label='label')
predictor.fit(train_data=train_data_path)
```

```
Epoch 2: 100%
INFO: Epoch 0, global step 2: 'val_accuracy' reached 0.22500 (best 0.22500),
INFO: Epoch 0, global step 5: 'val_accuracy' reached 0.81875 (best 0.81875),
INFO: Epoch 1, global step 7: 'val_accuracy' reached 0.91250 (best 0.91250),
INFO: Epoch 1, global step 10: 'val_accuracy' reached 0.97500 (best 0.97500),
INFO: Epoch 2, global step 12: 'val_accuracy' reached 0.97500 (best 0.97500),
INFO: Epoch 2, global step 15: 'val_accuracy' reached 0.97500 (best 0.97500),
```

https://auto.gluon.ai/stable/api/autogluon.multimodal.MultiModalPredictor.fit.html

Use AutoMM to Fit Models

```
predictor = MultiModalPredictor(label='label')
predictor.fit(train_data=train_data_path,
              hyperparameters={'model.timm_image.checkpoint_name': 'mobilevitv2_150_in22ft1k'})
```

Models

All model architecture families include variants with pretrained weights. There are specific model variants without any weights, it is NOT a bug. Help training new or better weights is always appreciated.

- Aggregating Nested Transformers - https://arxiv.org/abs/2105.12723
- BEiT - https://arxiv.org/abs/2106.08254
- Big Transfer ResNetV2 (BiT) - https://arxiv.org/abs/1912.11370
- Bottleneck Transformers - https://arxiv.org/abs/2101.11605
- CaiT (Class-Attention in Image Transformers) - https://arxiv.org/abs/2103.17239
- CoaT (Co-Scale Conv-Attentional Image Transformers) - https://arxiv.org/abs/2104.06399
- CoAtNet (Convolution and Attention) - https://arxiv.org/abs/2106.04803
- ConvNeXt - https://arxiv.org/abs/2201.03545

https://auto.gluon.ai/new/tutorials/multimodal/advanced_topics/customization.html

Evaluate on Test Dataset

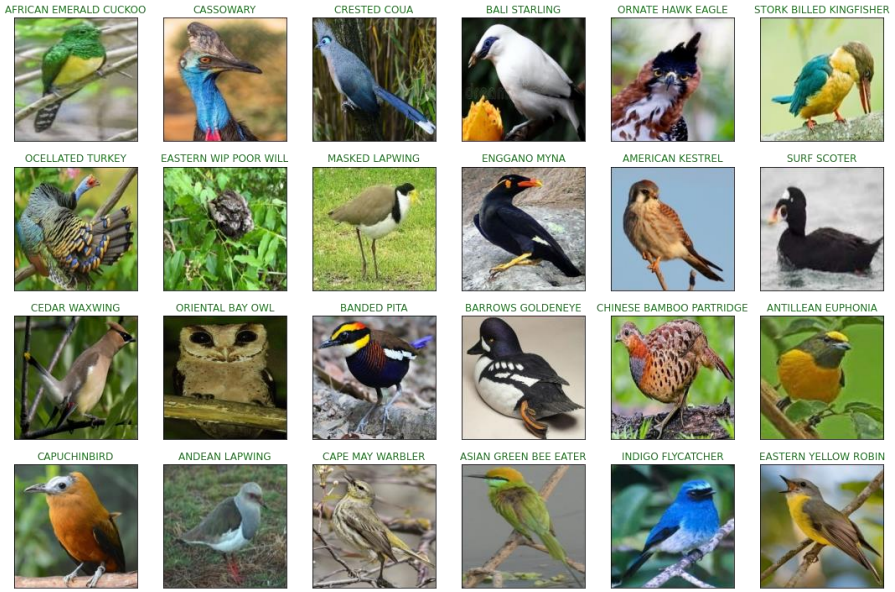
```
scores = predictor.evaluate(test_data_path, metrics=["accuracy"])
print('Top-1 test acc: %.3f' % scores["accuracy"])
```


```
Top-1 test acc: 1.000
```

```
MultiModalPredictor.evaluate(data: DataFrame | dict | list | str, query_data: list | None =
    None, response_data: list | None = None, id_mappings: Dict[str, Dict] | Dict[str, Series]
    | None = None, metrics: str | List[str] | None = None, chunk_size: int | None = 1024,
    similarity_type: str | None = 'cosine', cutoffs: List[int] | None = [1, 5, 10], label: str
    | None = None, return_pred: bool | None = False, realtime: bool | None = False, eval_tool:
    str | None = None)                                                              [source]
```

https://auto.gluon.ai/stable/api/autogluon.multimodal.MultiModalPredictor.evaluate.html

Hands-on experience

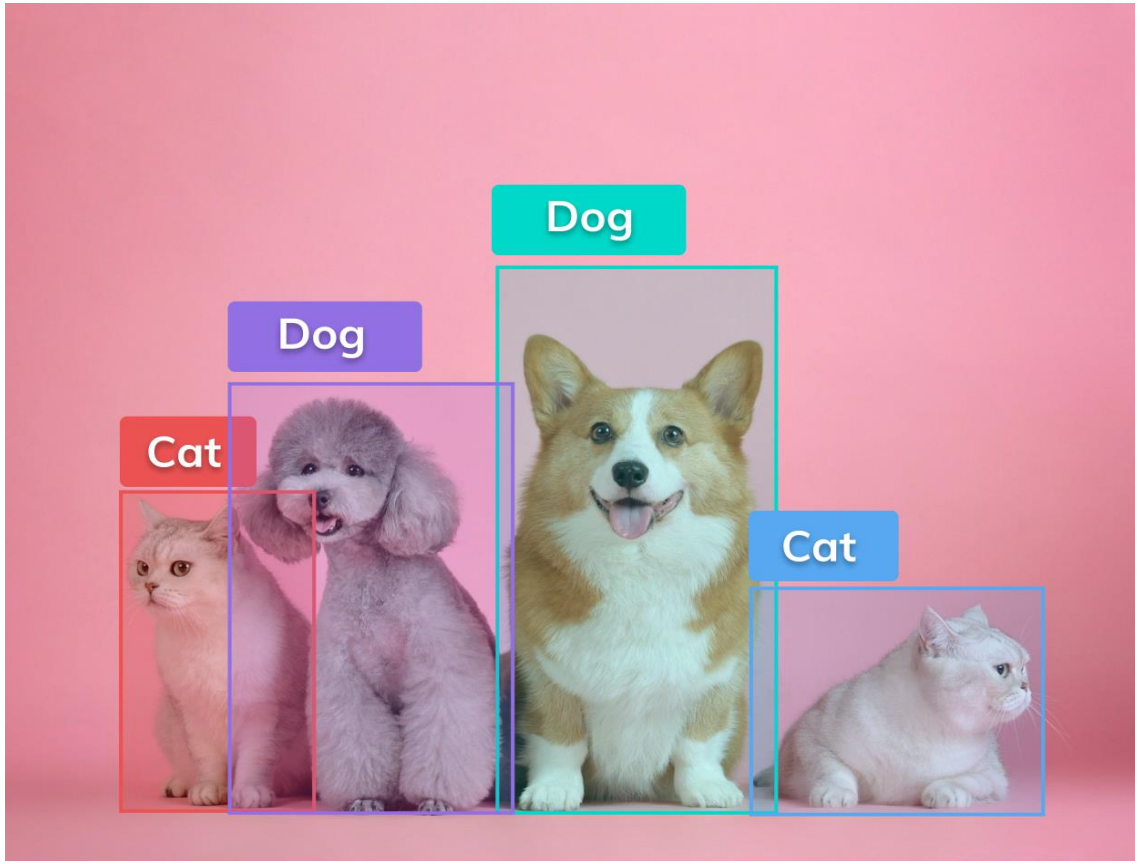


Data set of 525 bird species. 84635 training images, 2625 test images(5 images per species) and 2625 validation images(5 images per species. This is a very high quality dataset where there is only one bird in each image and the bird typically takes up at least 50% of the pixels in the image.

https://www.kaggle.com/datasets/gpiosenka/100-bird-species?select=train

Object detection

Object detection



- **Object Detection** is a computer vision task in which the goal is to detect and locate objects of interest in an image or video.

Annotation

```xml
<annotation verified="yes">
    <folder>Annotation</folder>
    <filename>Sachin.jpg</filename>
    <path>Cricketer-PascalVOC-export/Annotations/Sachin.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>1024</width>
        <height>672</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
    <name>cricketers</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>197.0057899090157</xmin>
        <ymin>121.13067465321564</ymin>
        <xmax>683.4342431761786</xmax>
        <ymax>619.3841424968474</ymax>
    </bndbox>
</object>
</annotation>
```

Pascal VOC

```
{
   "info"              : info,
   "images"            : [image],
   "annotations"       : [annotation],
   "licenses"          : [license],
}

info{
   "year"              : int,
   "version"           : str,
   "description"       : str,
   "contributor"       : str,
   "url"               : str,
   "date_created"      : datetime,
}

image{
   "id"                : int,
   "width"             : int,
   "height"            : int,
   "file_name"         : str,
   "license"           : int,
   "flickr_url"        : str,
   "coco_url"          : str,
   "date_captured"     : datetime,
}

license{
   "id"                : int,
   "name"              : str,
   "url"               : str,
}
```

COCO

Prepare Data

```
data_dir = os.path.join(download_dir, "tiny_motorbike")
train_path = os.path.join(data_dir, "Annotations", "trainval_cocoformat.json")
test_path = os.path.join(data_dir, "Annotations", "test_cocoformat.json")
```

```
        <width>500</width>
        <height>334</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>motorbike</name>
        <pose>Right</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>39</xmin>
            <ymin>24</ymin>
            <xmax>386</xmax>
            <ymax>275</ymax>
        </bndbox>
    </object>
</annotation>
```

Predictor

```
predictor = MultiModalPredictor(
    problem_type="object_detection",
    sample_data_path=train_path,
    presets='medium_quality',
)
```

```
predictor.fit(train_path)  # Fit
```

```
================== System Info ==================
AutoGluon Version:  1.1.0
Python Version:     3.10.14
Operating System:   Linux
Platform Machine:   x86_64
Platform Version:   #191-Ubuntu SMP Fri Feb 2 13:55:07 UTC 2024
CPU Count:          18
```

- presets – Presets regarding model quality, 'best_quality', 'high_quality' (default), and 'medium_quality'.

  medium_quality: YOLOX-large model pretrained on COCO dataset
  high_quality: DINO-Resnet50 model
  best_quality: DINO-SwinL model

Evaluate

```
predictor.evaluate(test_path)
```

```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.419
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.677
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.478
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.288
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.418
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.542
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.235
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.535
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.594
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.515
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.578
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.691
```

Average Precision (AP):

- AP % AP at IoU=.50:.05:.95 **(primary challenge metric)**
- $AP^{IoU=.50}$ % AP at IoU=.50 (PASCAL VOC metric)
- $AP^{IoU=.75}$ % AP at IoU=.75 (strict metric)

AP Across Scales:

- AP^{small} % AP for small objects: area < 32^2
- AP^{medium} % AP for medium objects: 32^2 < area < 96^2
- AP^{large} % AP for large objects: area > 96^2

Average Recall (AR):

- $AR^{max=1}$ % AR given 1 detection per image
- $AR^{max=10}$ % AR given 10 detections per image
- $AR^{max=100}$ % AR given 100 detections per image

AR Across Scales:

- AR^{small} % AR for small objects: area < 32^2
- AR^{medium} % AR for medium objects: 32^2 < area < 96^2
- AR^{large} % AR for large objects: area > 96^2

Inference

- To run inference on the entire test set, perform:

```
pred = predictor.predict(test_path, save_results=True)
```

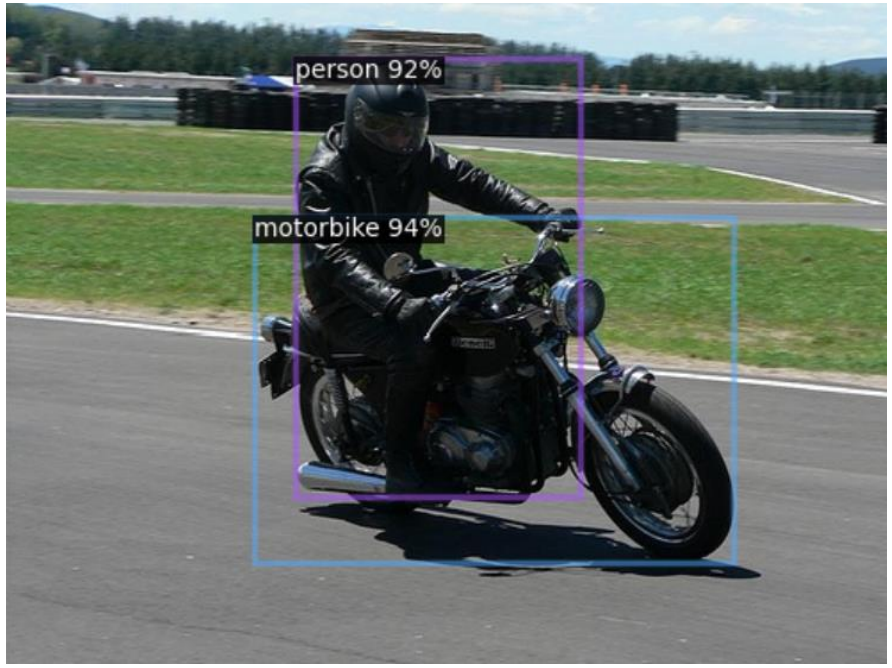
```
Using default root folder: ./pothole/pothole/Annotations/... Specify `root=...` if you feel it is wrong...
loading annotations into memory...
Done (t=0.00s)
creating index...
index created!

image     ./pothole/pothole/Annotations/../JPEGImages/po...
bboxes    [{'class': 'pothole', 'class_id': 0, 'bbox': [...
Name: 0, dtype: object
```

- In image, each row contains the image path.
- In bboxes, each row is a list of dictionaries, each one representing a bounding box:
  {“class”: <predictd_class_name>, “bbox”: [x1, y1, x2, y2], “score”: <confidence_score>}

Visualizing Results

- To visualize the detection bounding boxes, run the following:

```
from autogluon.multimodal.utils import ObjectDetectionVisualizer

conf_threshold = 0.4  # Specify a confidence threshold to filter out unwanted boxes
image_result = pred.iloc[38]

img_path = image_result.image  # Select an image to visualize

visualizer = ObjectDetectionVisualizer(img_path)  # Initialize the Visualizer
out = visualizer.draw_instance_predictions(image_result, conf_threshold=conf_threshold)
visualized = out.get_image()  # Get the visualized image

from PIL import Image
from IPython.display import display
img = Image.fromarray(visualized, 'RGB')
display(img)
```



Weed Detection

The Weed Image Detection Dataset is a curated collection of images aimed at facilitating research and development in the field of automated weed detection and management. It comprises a diverse range of images featuring different types of weeds commonly found in agricultural and natural environments.

SkyFusion: Aerial Object Detection

Introducing the SkyFusion dataset, designed for detecting tiny objects in satellite images. While many datasets focus on larger structures, SkyFusion addresses the gap for smaller classes like cars, trucks, ships, and airplanes.

Semantic Segmentation

Semantic Segmentation

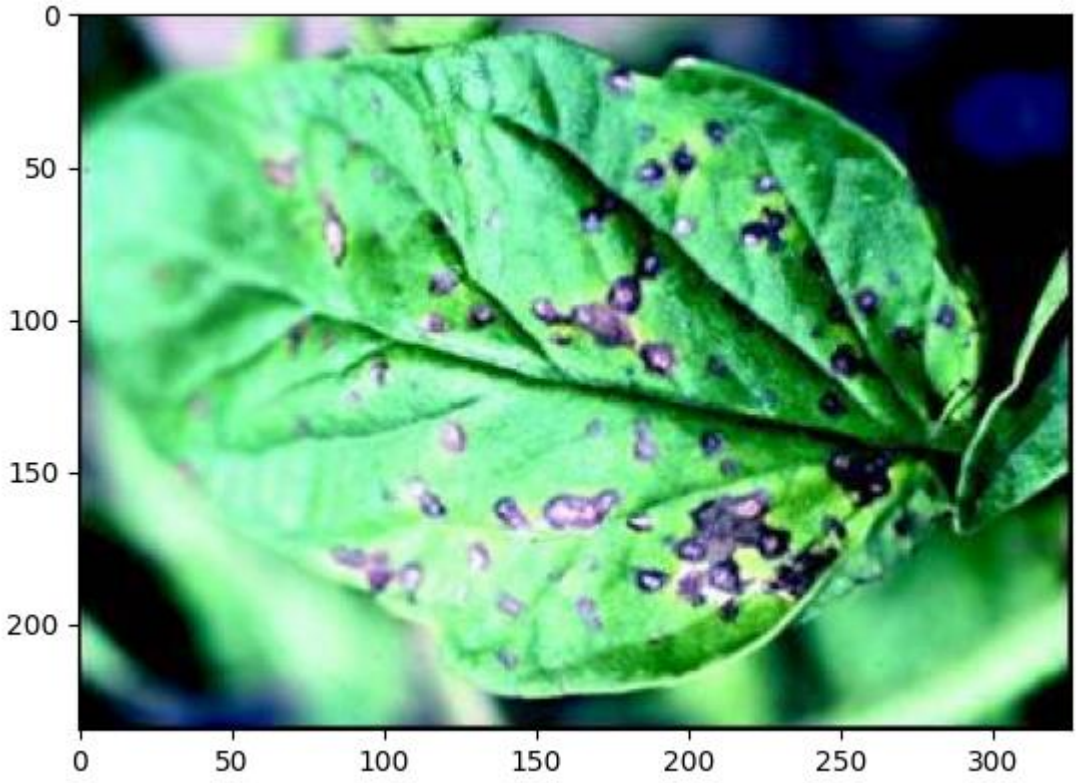
Input

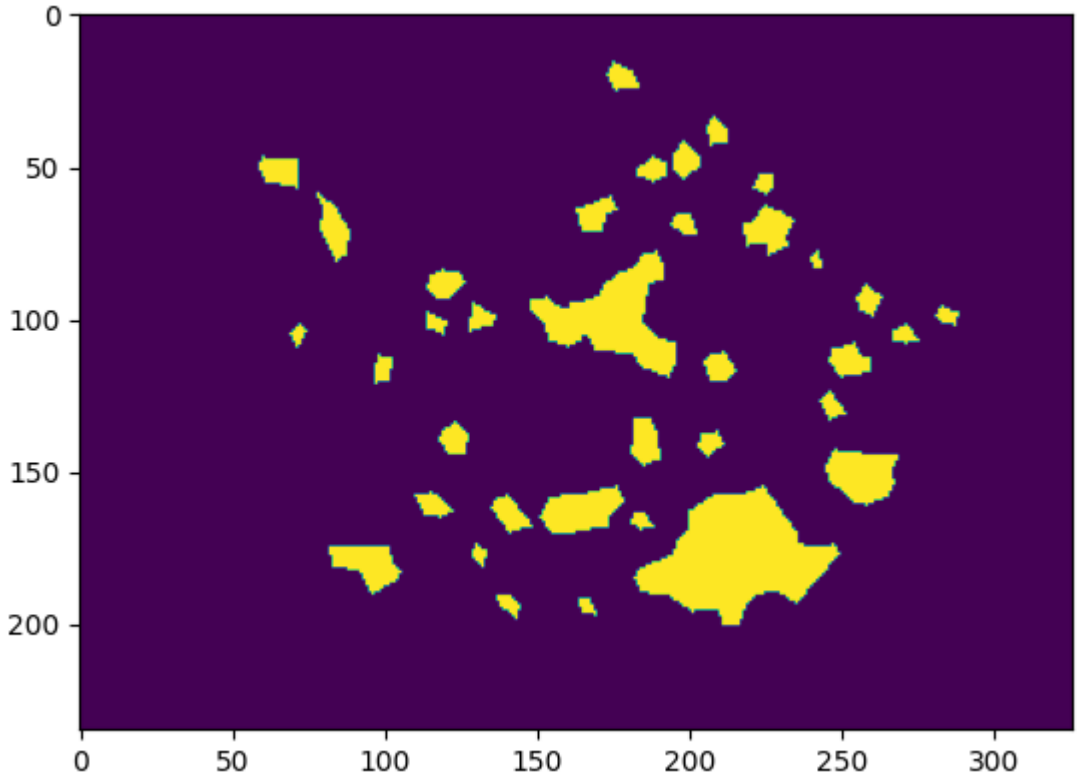
Ground truth

Prediction

- **Semantic Segmentation** is a computer vision task in which the goal is to categorize each pixel in an image into a class or object.

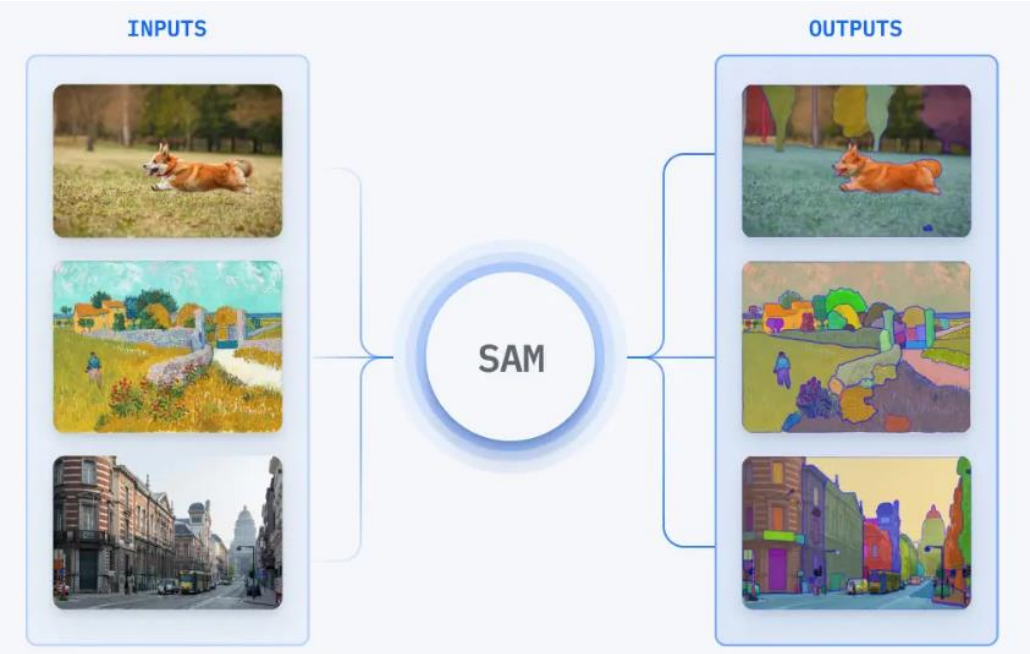
Leaf Disease Segmentation





- The dataset contains a collection of different diseases.
- Disease on leaves is one class and background is another class.

Segment Anything Model on AutoMM



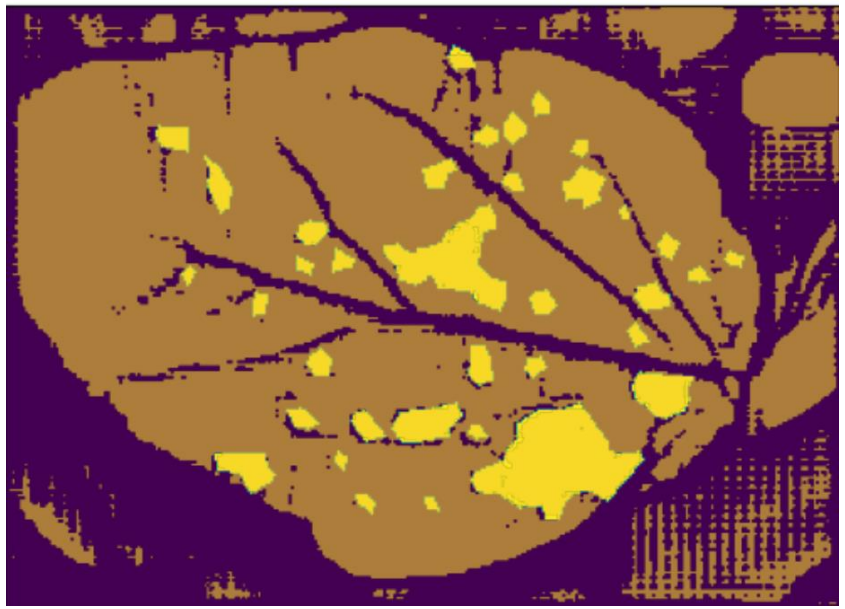
- The **Segment Anything Model** (SAM) is a foundation model pretrained on a vast dataset with 1 billion masks and 11 million images.
- While SAM performs exceptionally well on generic scenes, it encounters challenges when applied to specialized domains like remote sensing, medical imagery, agriculture, and manufacturing.

Prepare Data

```
train_data = pd.read_csv('leaf_disease_segmentation/train.csv', index_col=0)
val_data = pd.read_csv('leaf_disease_segmentation/val.csv', index_col=0)
test_data = pd.read_csv('leaf_disease_segmentation/test.csv', index_col=0)
image_col = 'image'
label_col = 'label'
```

Zero Shot Evaluation

```
from autogluon.multimodal import MultiModalPredictor
predictor_zero_shot = MultiModalPredictor(
    problem_type="semantic_segmentation",
    label=label_col,
     hyperparameters={
            "model.sam.checkpoint_name": "facebook/sam-vit-base",
        },
    num_classes=1, # forground-background segmentation
)
```



```
pred_zero_shot = predictor_zero_shot.predict({'image': [test_data.iloc[0]['image']]})
```

```
scores = predictor_zero_shot.evaluate(test_data, metrics=["iou"])
print(scores)
```

```
{'iou': 0.14012041687965393}
```

https://auto.gluon.ai/stable/tutorials/multimodal/advanced_topics/customization.html

Finetune SAM

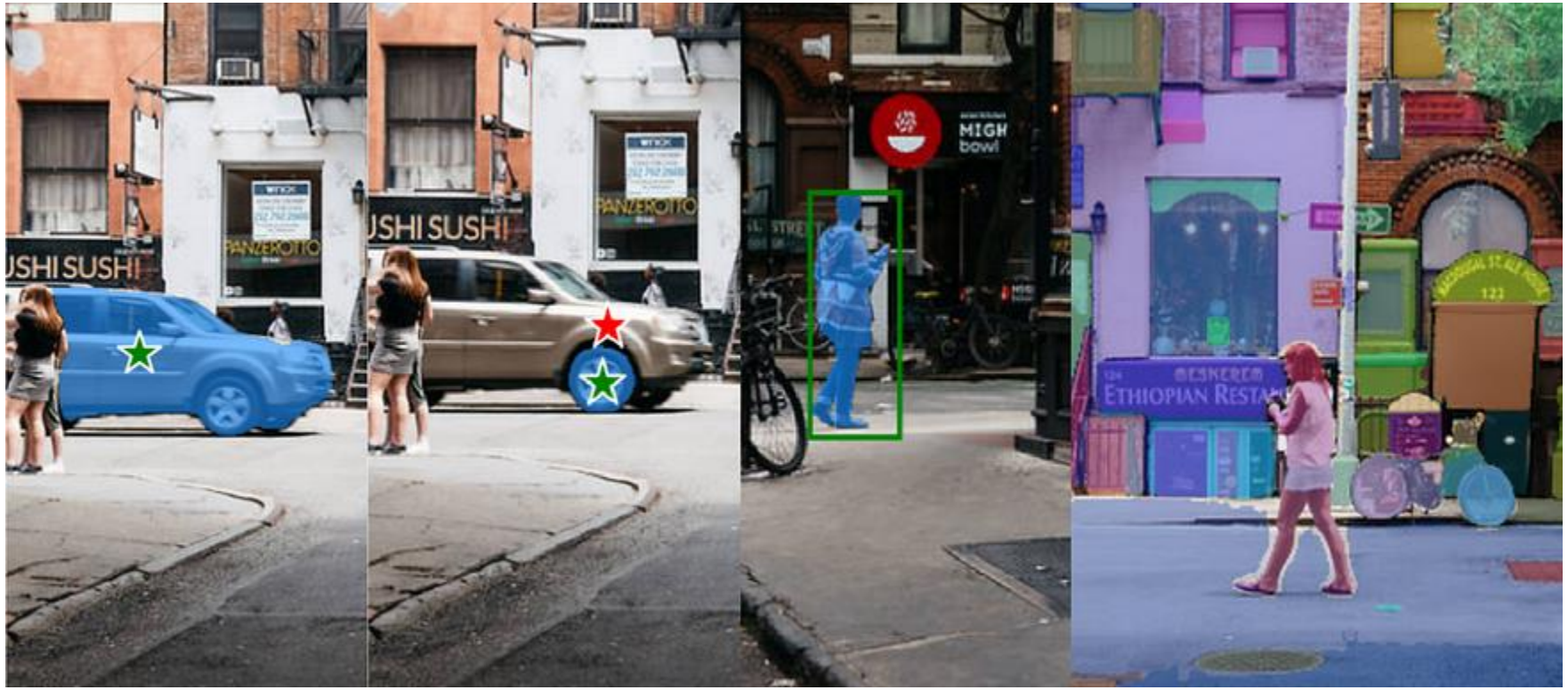
```
from autogluon.multimodal import MultiModalPredictor
import uuid
save_path = f"./tmp/{uuid.uuid4().hex}-automm_semantic_seg"
predictor = MultiModalPredictor(
    problem_type="semantic_segmentation",
    label="label",
     hyperparameters={
            "model.sam.checkpoint_name": "facebook/sam-vit-base",
        },
    path=save_path,
)
predictor.fit(
    train_data=train_data,
    tuning_data=val_data,
)
```

```
scores = predictor.evaluate(test_data, metrics=["iou"])
print(scores)
```

```
{'iou': 0.7180984616279602}
```

https://auto.gluon.ai/stable/tutorials/multimodal/advanced_topics/customization.html

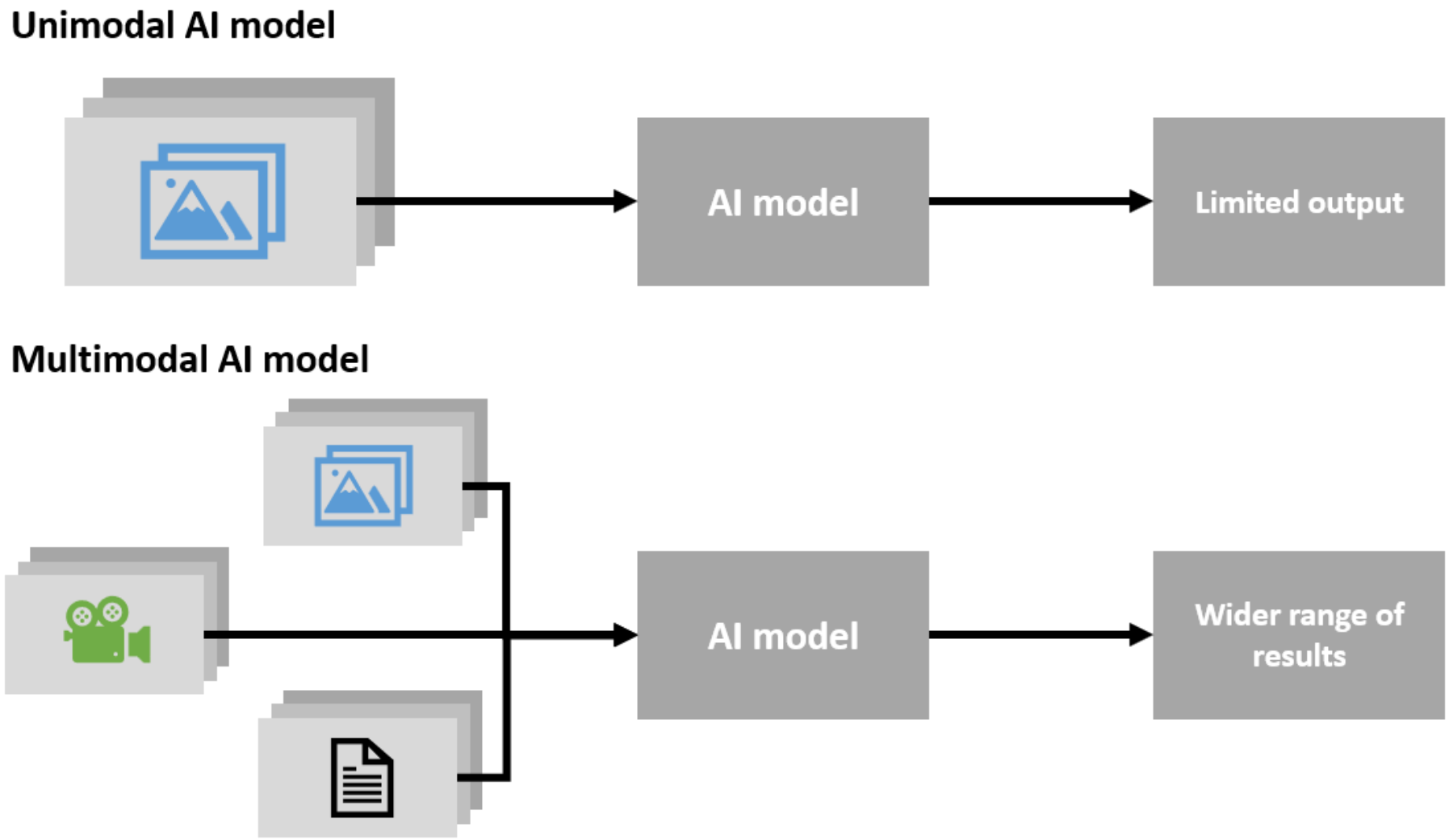
Zero-shot semantic segmentation



Zero-shot segmentation is a technique in computer vision that aims to segment objects in images without having seen any examples of those objects during the training phase.

MultiModal

MultiModal


Unimodal AI model
AI model
Limited output
Multimodal AI model
AI model
Wider range of results

PetFinder dataset

- The PetFinder dataset predicts the speed at which a pet is adopted based on the pet's listing on PetFinder.

Name	Age	Breed1	Breed2	Gender
Coco The Beach	24	265	288	1
Selampit 27	1	307	307	3
Lolita	2	307	0	2
Muffin	1	307	0	2
BROWNIE GIRL	12	307	0	2
Snowie (Female)	3	265	0	2
Molly (momol)	6	300	0	2
Sasha	24	307	0	2
	2	141	307	1
le Abby Needs A Home	3	141	109	2
Pepsi	3	265	0	2
Cara & Kids	2	265	0	3
KEITH	11	307	307	1
Snazzy	8	307	0	2
Kitten 2	2	247	266	2
No Name	1	265	0	2
Abandoned Puppies	12	307	0	3
Fisher	2	307	0	2
Lucy	2	307	0	2

PetFinder dataset

```
import pandas as pd
dataset_path =  './petfinder'
train_data = pd.read_csv(f'{dataset_path}/train.csv', index_col=0)
test_data = pd.read_csv(f'{dataset_path}/test.csv', index_col=0)
label_col = 'AdoptionSpeed'
```

```
Type                                                          2
Name                                                 Yumi Hamasaki
Age                                                           4
Breed1                                                      292
Breed2                                                      265
Gender                                                        2
Color1                                                        1
Color2                                                        5
Color3                                                        7
MaturitySize                                                  2
FurLength                                                     2
Vaccinated                                                    1
Dewormed                                                      3
Sterilized                                                    2
Health                                                        1
Quantity                                                      1
Fee                                                           0
State                                                     41326
RescuerID                          bcc4e1b9557a8b3aaf545ea8e6e86991
VideoAmt                                                      0
Description       I rescued Yumi Hamasaki at a food stall far aw...
PetID                                                  7d7a39d71
PhotoAmt                                                    3.0
AdoptionSpeed                                                 0
Images            /home/work/SEWO/AutoGluon/multimodal/petfinder...
Name: 0, dtype: object
```

I rescued Yumi Hamasaki at a food stall far away in Kelantan. At that time i was on my way back to KL, she was suffer from stomach problem and looking very2 sick.. I send her to vet & get the treatment + vaccinated and right now she's very2 healthy.. About yumi : - love to sleep with ppl - she will keep on meowing if she's hugry - very2 active, always seeking for people to accompany her playing - well trained (poo+pee in her own potty) - easy to bathing - I only feed her with these brands : IAMS, Kittenbites, Pro-formance Reason why i need someone to adopt Yumi: I just married and need to move to a new house where no pets are allowed :(As Yumi is very2 special to me, i will only give her to ppl that i think could take care of her just like i did (especially on her foods things)..

Training and Evaluation

```
from autogluon.multimodal import MultiModalPredictor
predictor = MultiModalPredictor(label=label_col)
predictor.fit(
    train_data=train_data,
)
```

```
  | Name              | Type                | Params
-------------------------------------------------------
0 | model             | MultimodalFusionMLP | 207 M
1 | validation_metric | BinaryAUROC         | 0
2 | loss_func         | CrossEntropyLoss    | 0
-------------------------------------------------------
207 M     Trainable params
0         Non-trainable params
207 M     Total params
828.117   Total estimated model params size (MB)
```

```
scores = predictor.evaluate(test_data, metrics=["roc_auc"])
scores
```

```
{'roc_auc': 0.882}
```

Entity Extraction

Aeva, a Mountain View, California-based lidar company started by two former
[Company] [Location]

Apple engineers and backed by Porsche SE, is merging with special purpose
[Company] [Company]

acquisition company InterPrivate Acquisition Corp., with a post-deal market
[Company]

valuation of $2.1 billion.
[Monetary Value]

text_snippet	Uefa Super Cup : Real Madrid v Manchester United
image	/twitter2017_images/17_06_1818.jpg
entity_annotations	[{"entity_group": "B-MISC", "start": 0, "end":...

Name: 0, dtype: object

Named Entity Recognition, which recognizes the type of each entity in a corpus.
Named entity recognition can be used to find which words in a corpus refer to people, places, organizations, and more.

https://codewithkyrian.github.io/transformers-php/token-classification

감사합니다

Reference

https://www.youtube.com/watch?v=fdfGb2jq-_c
https://auto.gluon.ai/stable/index.html
https://medium.com/@denizgunay/random-forest-af5bde5d7e1e
https://gaussian37.github.io/ml-concept-RandomForest/
https://gist.github.com/tylerwx51/fc8b316337833c877785222d463a45b0
https://arxiv.org/pdf/2003.06505
https://www.automl.org/wp-content/uploads/2023/07/1_ESSAI_General_Intro_to_AutoML-1.pdf

Kepler Exoplanet Search Results

The Kepler Space Observatory is a NASA-build satellite that was launched in 2009. The telescope is dedicated to searching for exoplanets in star systems besides our own, with the ultimate goal of possibly finding other habitable planets besides our own. The original mission ended in 2013 due to mechanical failures, but the telescope has nevertheless been functional since 2014 on a "K2" extended mission.

Kepler Exoplanet Search Results

- kepoi_name: A KOI is a target identified by the Kepler Project that displays at least one transit-like sequence within Kepler time-series photometry that appears to be of astrophysical origin and initially consistent with a planetary transit hypothesis
- kepler_name: [These names] are intended to clearly indicate a class of objects that have been confirmed or validated as planets—a step up from the planet candidate designation.
- koi_disposition: The disposition in the literature towards this exoplanet candidate. One of CANDIDATE, FALSE POSITIVE, NOT DISPOSITIONED or CONFIRMED.
- koi_pdisposition: The disposition Kepler data analysis has towards this exoplanet candidate. One of FALSE POSITIVE, NOT DISPOSITIONED, and CANDIDATE.
- koi_score: A value between 0 and 1 that indicates the confidence in the KOI disposition. For CANDIDATEs, a higher value indicates more confidence in its disposition, while for FALSE POSITIVEs, a higher value indicates less confidence in that disposition.

https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html