

Data Augmentation for images and tabular data

Fabio Massimo D'Amore

Chi sono

- Più di 20 anni di esperienza nel trasporto aereo (IT project manager, Revenue Management per Cargo)
- Ricercatore in ambito IA su tematiche health care in collaborazione ISTC/CNR, IRCCS S. Lucia, La Sapienza
- Docente AS-AI



<https://www.linkedin.com/in/fabio-massimo-d-amore-83859b133/>

Fabio M. D'Amore

Agenda

- Welcome and prerequisites
- What is data augmentation
- Why use data augmentation
- Data augmentation pipeline
 - quiz
- Some data augmentation techniques
 - let's code!
- Open challenges end new paradigm
- Q&A



...a break every 50 min. or on request!

Prerequisite

- Basic Machine learning concepts:

Supervised learning is a machine learning paradigm where a model is trained on a labeled dataset, consisting of input-output pairs (x_i, y_i) , to learn a function that maps inputs to outputs.

The goal is to generalize this mapping to predict the output y for new, unseen inputs x .

A **classifier** is a supervised learning model that maps input data x to discrete class labels $y \in \{1, 2, \dots, K\}$.

Its objective is to learn a decision function that accurately assigns the correct class to unseen inputs based on the training data.

Overfitting occurs when a model learns the training data too well (ie, memorize it), including its noise and specific patterns, resulting in poor generalization to unseen data. It is characterized by low training error but high validation or test error, indicating a lack of true predictive power.

Data **bias** is a phenomenon that occurs when the data used to train a machine learning model reflects reality in a distorted way, introducing systematic prejudices that can compromise the fairness and accuracy of the model's predictions.

Prerequisite

- Metrics

Accuracy: Measures **how often the model is correct**, considering **all predictions**, both positive and negative.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

It is useful **when the classes are balanced**. It can be **misleading** in imbalanced datasets, where one class dominates.

Recall: Measures **how well the model identifies actual positive cases**, i.e., **how many true positives it captures out of all real positives**.

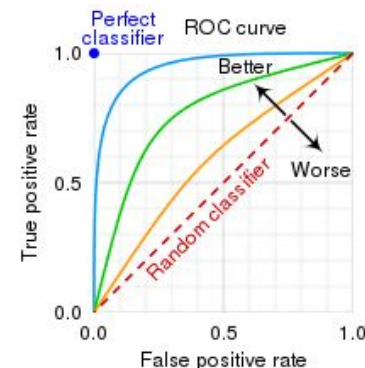
$$\text{Recall} = \frac{TP}{TP + FN}$$

It is useful **when it is important not to miss any positive cases** (e.g., in disease detection or fraud detection). It does **not account for false positives**

AUC (Area Under the Curve) :The **ROC curve** is a graphical plot that shows the **performance of a binary classification model** at various **threshold settings**.

It plots:

- True Positive Rate (TPR) = Recall = $\frac{TP}{TP + FN}$
- False Positive Rate (FPR) = $\frac{FP}{FP + TN}$



What is data augmentation

Wikipedia

Data augmentation is a statistical technique which allows maximum likelihood estimation from incomplete data [...] the technique is widely used in [machine learning](#) to reduce [overfitting](#) when training machine learning models, achieved by training models on several slightly-modified copies of existing data.

https://en.wikipedia.org/wiki/Data_augmentation



ChatGPT

Data augmentation in machine learning involves artificially increasing the size of a dataset by applying transformations such as rotation, scaling, or flipping to the existing data. This technique helps improve model generalization by providing more diverse examples for training without collecting new data. It's commonly used in tasks like image classification and natural language processing.



What is data augmentation

Data augmentation is a technique used to increase the size and diversity of a training dataset, is commonly used in machine learning (especially for deep learning models) to increase model performance.



A chili



Still a chili

image from: <https://valohai.com/blog/data-augmentation/>

Why use data augmentation



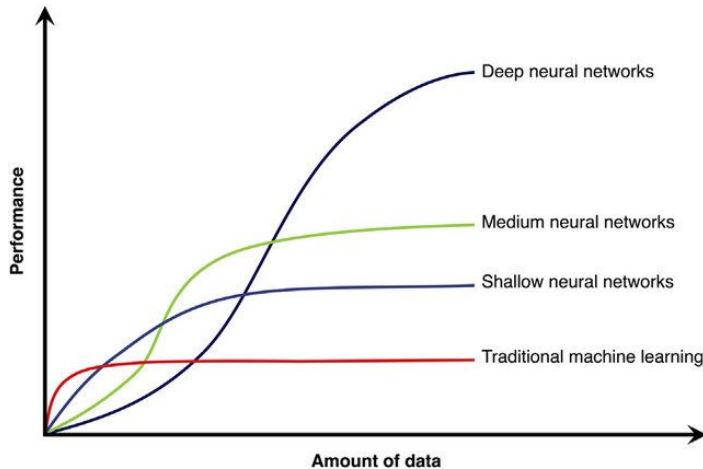
Why use data augmentation

To train high performance supervised models, data needs to be labelled (or segmented) and this is done by hand.



Image segmentation is a **computer vision method** that breaks down an image into meaningful regions by **classifying each pixel** according to what it represents.

Why use data augmentation



<https://www.researchgate.net/publication>



https://mrmen.fandom.com/wiki/Mr._Greedy

Train a good DNN model = spend a lot of money for good training dataset

Why use data augmentation

To increase model performance (accuracy) maintaining costs control.



Why use data augmentation

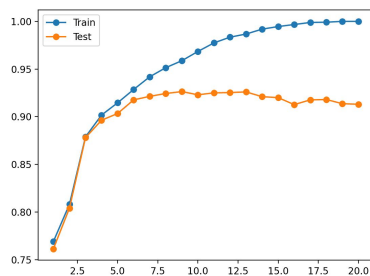
Small datasets

To increase quantity and diversity of data.

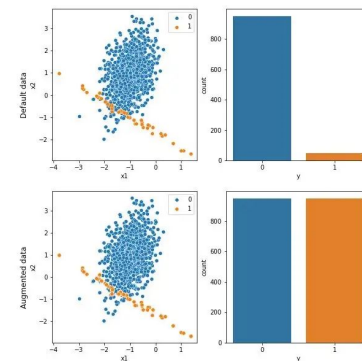
To rescue from over-fitting.

Imbalanced datasets

To oversample minority classes.



Without collecting real new data



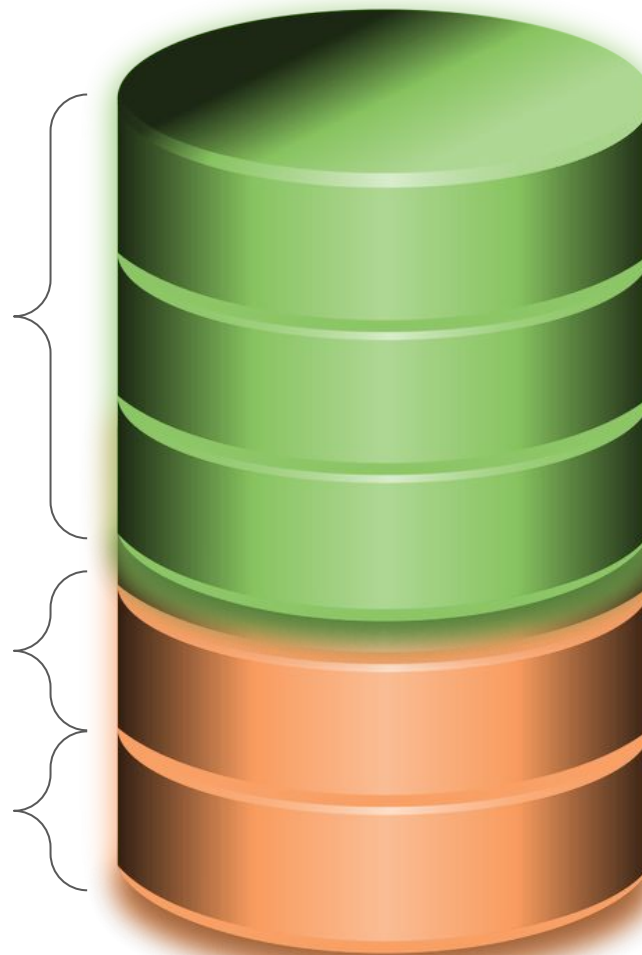
What is data augmentation

Scope

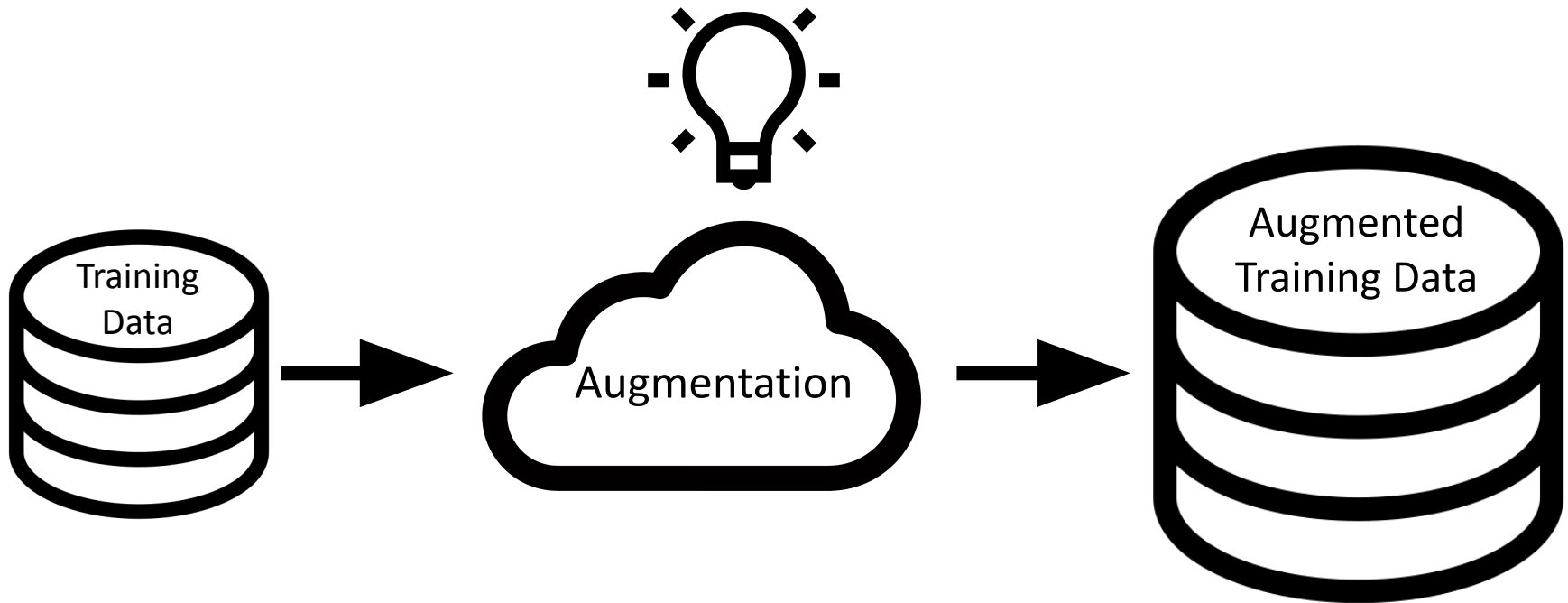
Training set

~~Validation~~

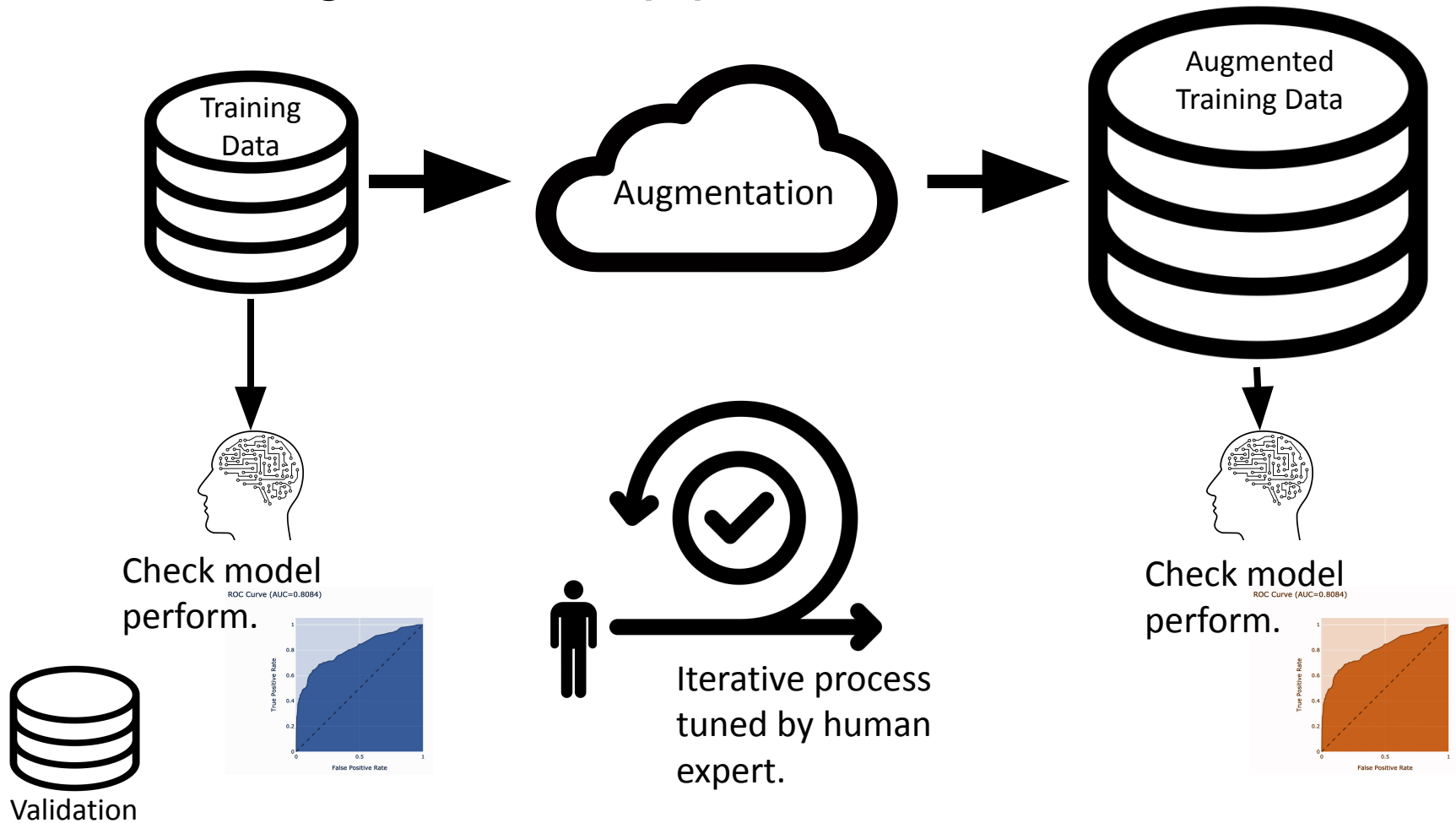
~~Test sets~~



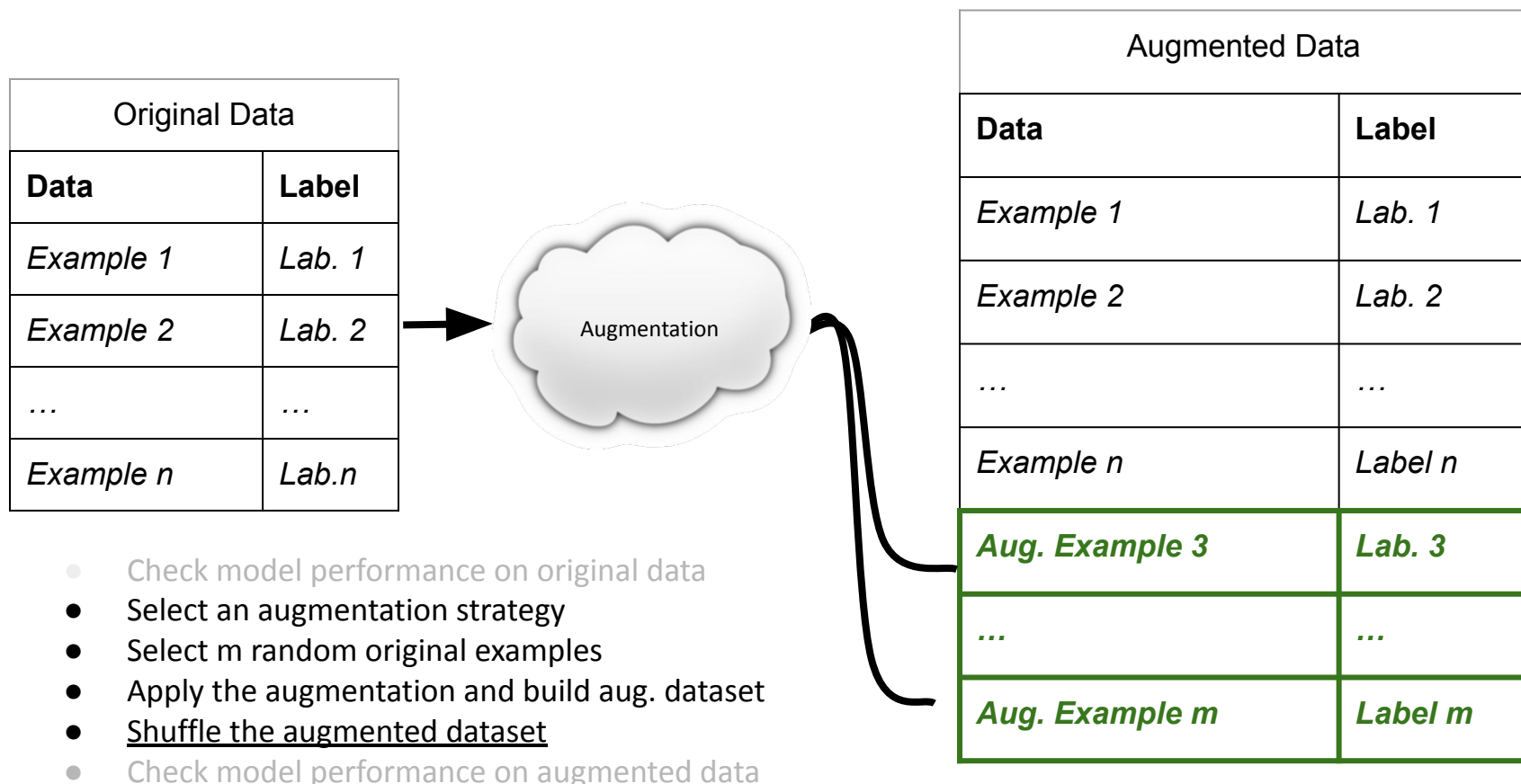
What is data augmentation?



Data augmentation pipeline

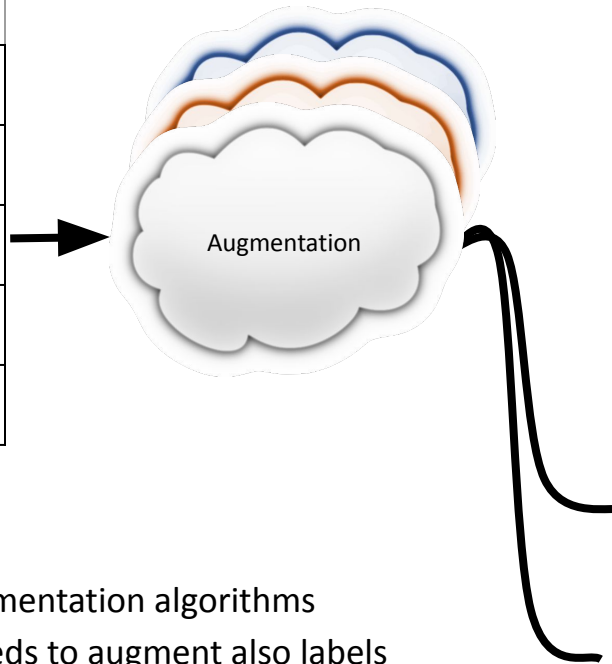


Typical augmentation cycle



Typical augmentation cycle

Original Data	
Data	Label
<i>Example 1</i>	<i>Lab. 1</i>
<i>Example 2</i>	<i>Lab. 2</i>
...	...
<i>Example n</i>	<i>Lab. n</i>

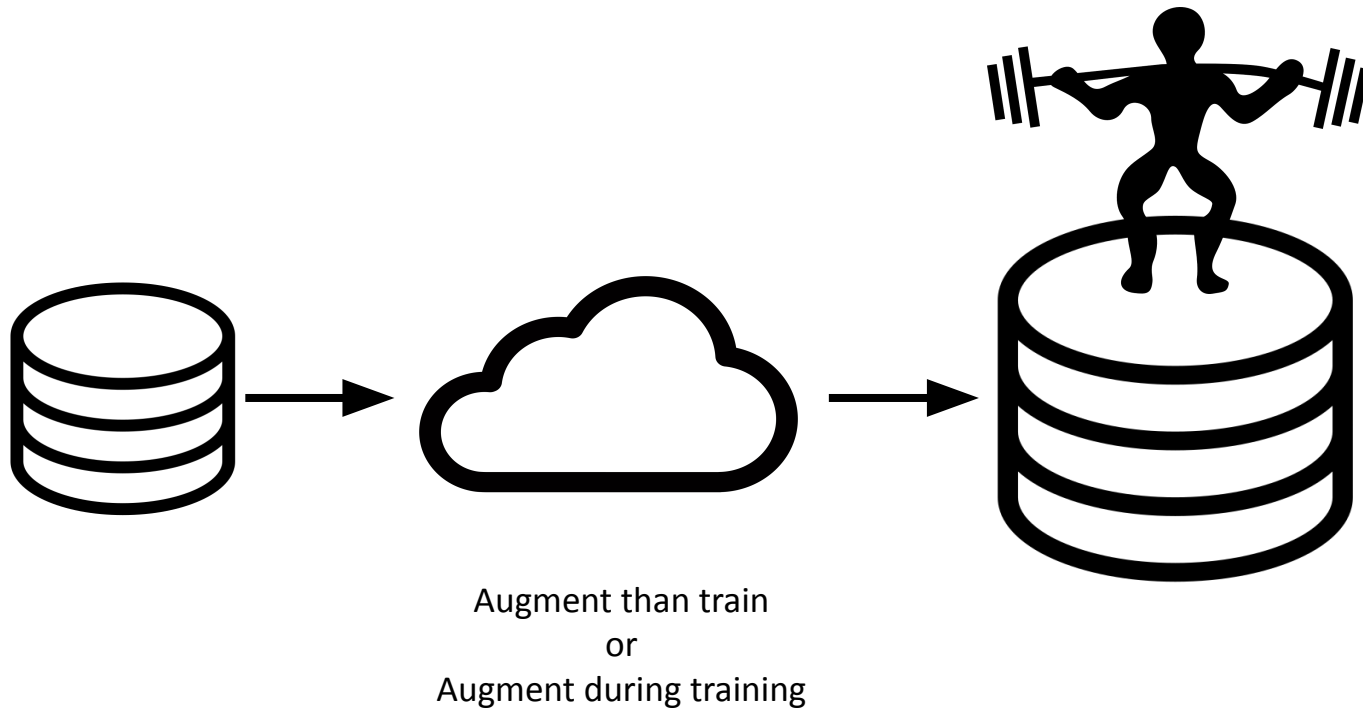


Augmented Data	
Data	Label
<i>Example 1</i>	<i>Lab. 1</i>
<i>Example 2</i>	<i>Lab. 2</i>
...	...
<i>Example n</i>	<i>Label n</i>
Aug. Example 3	A. Lab. 3
...	...
Aug. Example m	A.Label m

Moreover:

- Select random Augmentation algorithms
- Some problems needs to augment also labels

Some aug. technique



https://www.tensorflow.org/tutorials/images/data_augmentation?hl=en

Data Augmentation pros

Enhanced generalization: enhances a model's ability to generalize by exposing it to a broader spectrum of data variations during training, thereby increasing its robustness to diverse scenarios.

Mitigated overfitting: mitigates overfitting, a common machine learning challenge where a model loses the ability to generalize, by diversifying the dataset. This diversity helps prevent the model from memorizing specific instances and improves its ability to generalize to unseen data.

Improved accuracy: Data augmentation often results in enhanced accuracy on both training and test datasets. By training on a larger and more diverse dataset, the model can extract more robust features, leading to better performance.

Cost-effectiveness: Data augmentation offers a cost-effective approach to expand the size and diversity of training datasets, particularly beneficial for tasks where acquiring new data is resource-intensive or time-consuming.

Data Augmentation drawbacks

Elevated computational overhead: Data augmentation often escalates the computational burden of model training, particularly noticeable with deep learning models, as they require training on an expanded image dataset.

Risk of introducing noise: Careless application of data augmentation can introduce unwanted noise into the training data, potentially compromising model performance on the test set.

Restricted effectiveness in specific tasks: Data augmentation might exhibit limited effectiveness in tasks characterized by inherently diverse data, rendering its application less impactful in such scenarios.

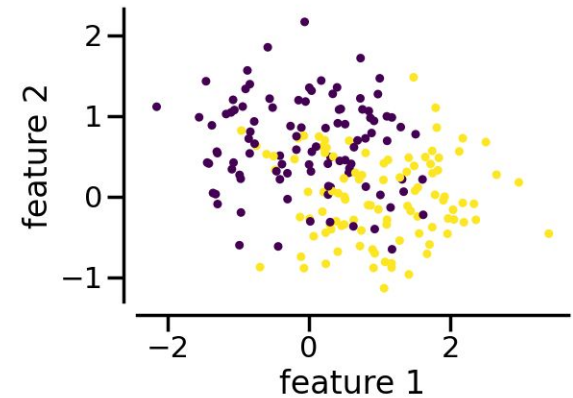
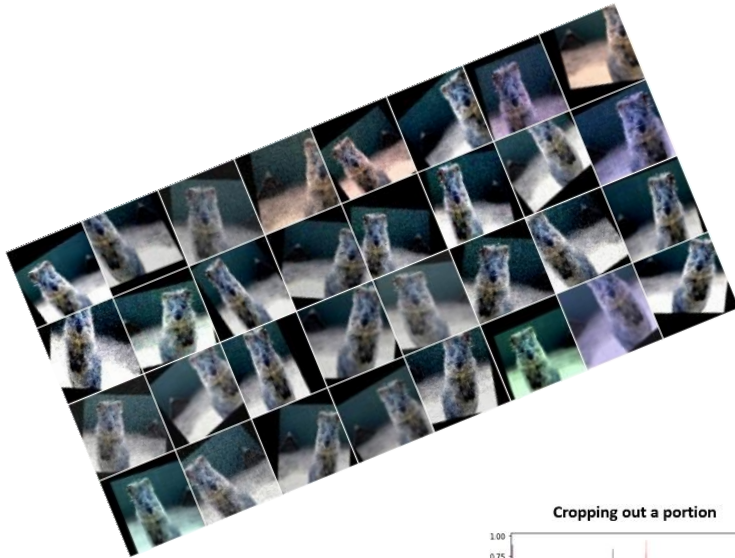
Data Augmentation risk

Fairness and Bias: Data augmentation techniques should be applied in a manner that does not perpetuate or exacerbate biases present in the training data. For instance, if the original dataset is biased towards certain demographics, applying augmentation techniques that maintain or amplify these biases could lead to unfair outcomes. It's crucial to assess and address biases throughout the data augmentation process to ensure fairness in model predictions and decisions.

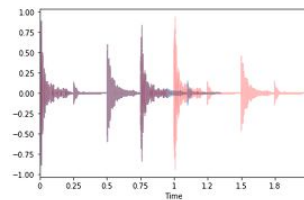
Quiz

QUESTIONS				
1-	A	B	C	D
2-	A	B	C	D
3-	A	B	C	D
4-	A	B	C	D
5-	A	B	C	D
6-	A	B	C	D

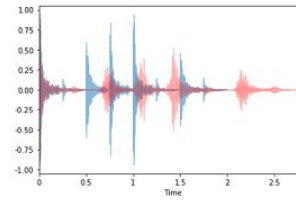
Data Augmentation techniques



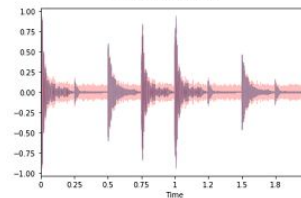
Cropping out a portion



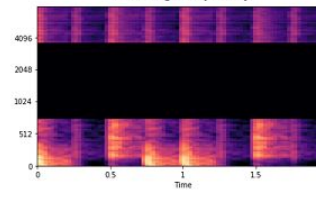
Changing Speed



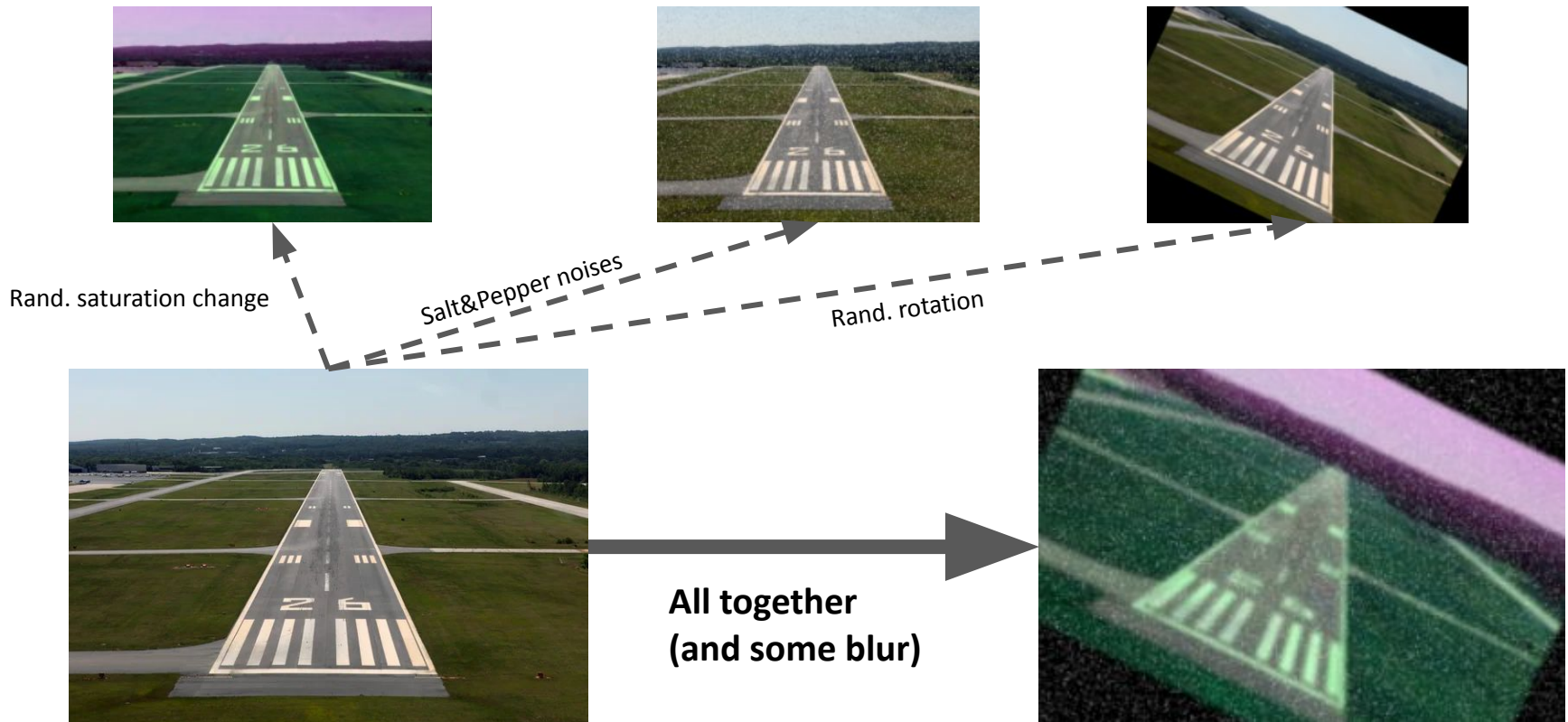
Injecting Noise



Masking Frequency



Some aug. technique for images



Some aug. technique for images

```
# Install libraries for augmenting and displaying images
!pip install imgaug

# For augmenting the images
import imgaug.augmenters as iaa

# Define image augmentation pipeline
seq = iaa.Sequential([
    iaa.Crop(px=(10, 30), keep_size=False),          # crop by 10-30px, original size
    iaa.Affine(rotate=(-25, 25)),                    # rotate -25 to 25 degrees
    iaa.Dropout(p=(0, 0.1)),                          # drop % of all pix. (convert them to black)
    iaa.GaussianBlur(sigma=(0, 4.0)),                # blur using gaussian kernel with sigma of 0-4
    iaa.AddToHueAndSaturation((-100, 100), per_channel=True), # add or sub. from (for hue, sat., lightness)
    iaa.Salt(0.08),
    iaa.Pepper(0.08),
    iaa.Fliplr(0.3), #flip left-right
])
```

This is the aug.
sequence

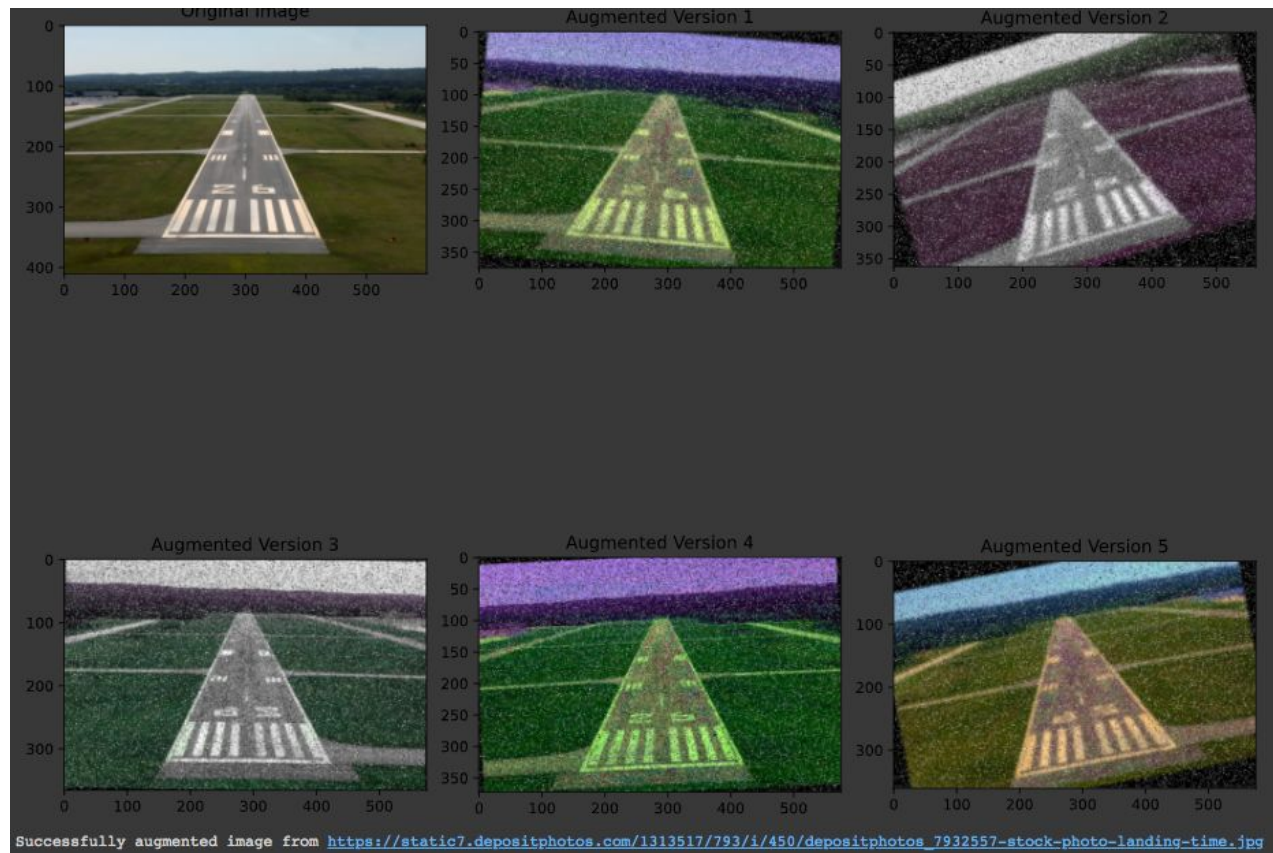
Some aug. technique for images

```
# Insert your URL here
url = "https://static7.depositphotos.com/1313517/793/i/450/depositphotos_7932557-stock-photo-landing-time.jpg"
# Augment train images
try:
    # Read in image from URL
    img = url to image(url)
    # Augment image using settings defined above in seq
    augimgs = [seq.augment(image=img) for x in range(6)]
    # Display augmented image
    nrows, ncols = 2, 3
    fig, ax = plt.subplots(nrows=nrows, ncols=ncols, figsize=(12,12))
    ax[0][0].imshow(img)
    ax[0][0].set title("Original Image")
    # plot simple raster image on each sub-plot
    for i, axi in list(enumerate(ax.flat))[1:]:
        axi.imshow(augimgs[i])
        axi.set_title("Augmented Version "+str(i))
    plt.tight_layout(True)
    plt.show()
    # Display message to track augmentation process by image
    print('Successfully augmented image from {}'.format(url))
except:
    print('Error: check if web address for image from {} is valid'.format(url))
```



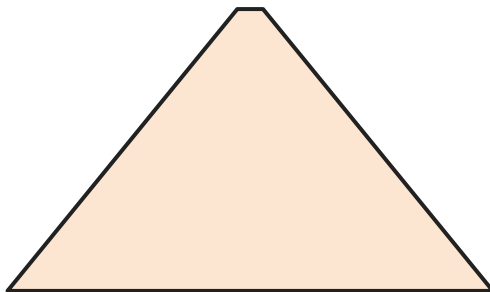
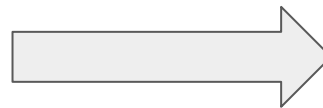
that's all...

Some aug. technique for images

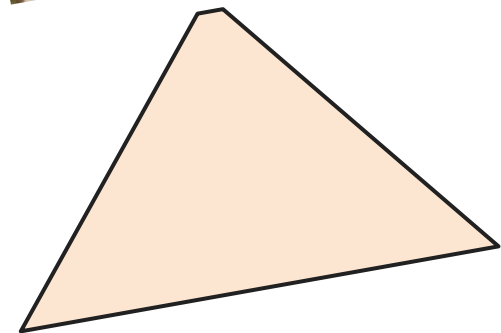


Some aug. technique for images

Augment also data labels (segmentation)



Augmentation
strategy: rotate
left/right of random
deg.



Some aug. technique for images

Image



Image and mask



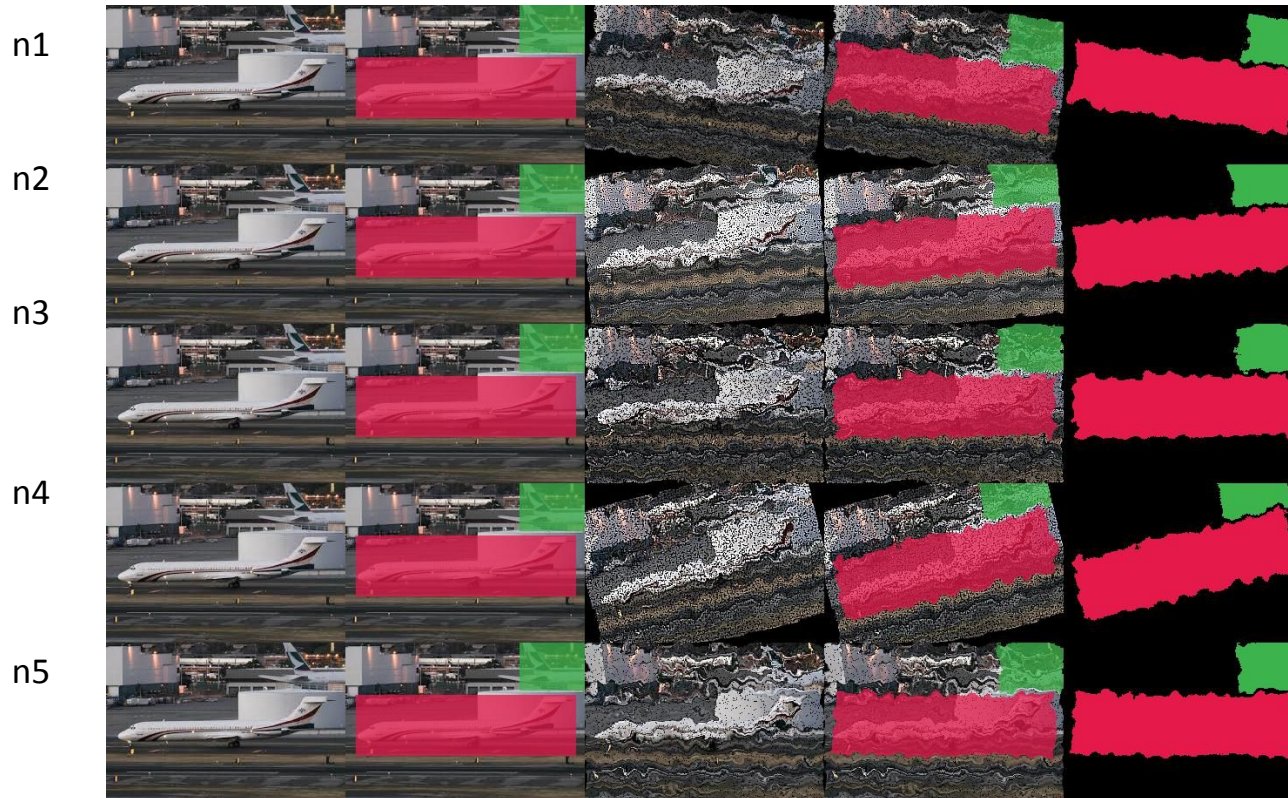
Some aug. technique for images

```
# Augment train images and bounding boxes
url = "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR97tkpCz3QNbPsZqRyXfrC7-k2ICDpGkJ-NQ&usqp=CAU"
# Read in image from URL
image = url to image(url)
# Define an example segmentation map
# Here, we arbitrarily place some squares on the image.
# Class 0 is our intended background class.
segmap = np.zeros((image.shape[0], image.shape[1], 1), dtype=np.int32)
segmap[60:130, 12:265, 0] = 1
segmap[0:55, 200:275, 0] = 2
segmap = SegmentationMapsOnImage(segmap, shape=image.shape)
# Define our augmentation pipeline.
seq = iaa.Sequential([
    iaa.Dropout([0.05, 0.2]),           # drop 5% or 20% of all pixels
    iaa.Sharpen((0.0, 1.0)),           # sharpen the image
    iaa.Affine(rotate=(-45, 45)),       # rotate by -45 to 45 degrees (affects segmaps)
    iaa.ElasticTransformation(alpha=50, sigma=5) # apply water effect (affects segmaps)
], random_order=True)
# Augment images and segmaps.
images_aug = []
segmaps_aug = []
for i in range(5):
    images_aug_i, segmaps_aug_i = seq(image=image, segmentation_maps=segmap)
    images_aug.append(images_aug_i)
    segmaps_aug.append(segmaps_aug_i)
```

here we def.
the bounding
boxes...

...here the aug. of
images and masks

Some aug. technique for images



Some aug. technique for images



...time to code

Some aug. technique for tabular data

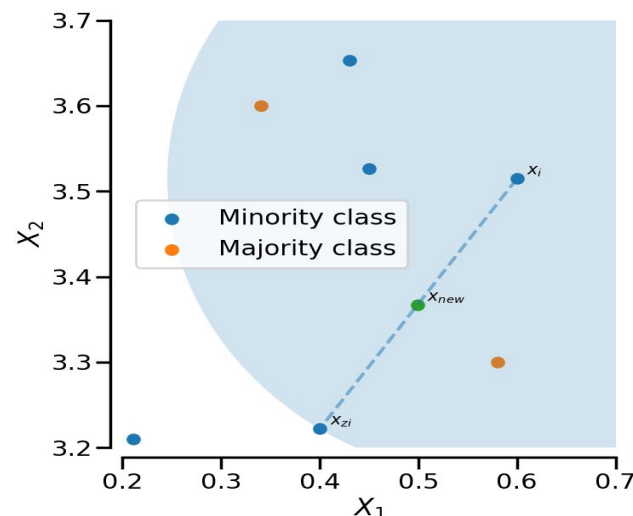
The strategies to generate new samples:

- [RandomOverSampler](#) over-sampling by duplicating some of the original samples of the minority class.
- [SMOTE](#) uses k-nearest algorithm to select random samples from minority class and generate the new example as convex combination of them.
- [ADASYN](#) focuses on generating synthetic samples next to the original samples which are wrongly classified using a k-Nearest Neighbors classifier.

Some aug. technique for tabular data

Recup

- Imbalanced tabular dataset
- Use of Synthetic Minority Oversampling Technique, or SMOTE
- Pipeline for augmentation



Considering a sample x_i , a new sample x_{new} will be generated at a random distance from one of its k nearest-neighbors (blue circle)

Use case: predictive maintenance

Data scenario:

- 99% normal operations
- 1% failures

Problem:

train a model that predict failures

Constrain:

model is fixed (decision tree classifier), work on data !



Some aug. technique for tabular data

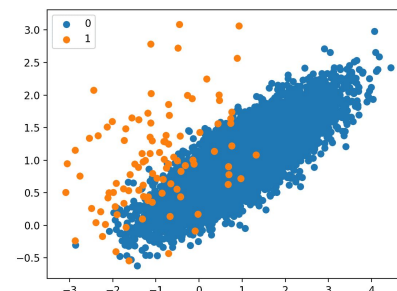
```
# Generate and plot a synthetic imbalanced classification dataset
from collections import Counter
from sklearn.datasets import make_classification
from matplotlib import pyplot
from numpy import where

# define dataset
X, y = make_classification(n_samples=10000, n_features=2, n_redundant=0,
                          n_clusters_per_class=1, weights=[0.99], flip_y=0, random_state=1)

# summarize class distribution
counter = Counter(y)
print(counter)

# scatter plot of examples by class label
for label, _ in counter.items():
    row_ix = where(y == label)[0]
    pyplot.scatter(X[row_ix, 0], X[row_ix, 1], label=str(label))
pyplot.legend()
pyplot.show()
```

99% of
samples are in
class 0



```
Counter({0: 9900, 1: 100})
```

Some aug. technique for tabular data

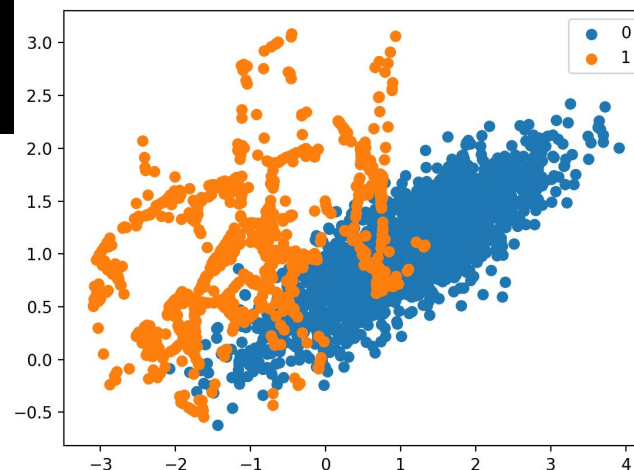
```
from imblearn.pipeline import Pipeline
...
# Oversampling to 10% of largest class
# Undersampling to 50%
over = SMOTE(sampling_strategy=0.1)
under =
RandomUnderSampler(sampling_strategy=0.5)
# Build the pipeline
steps = [('o', over), ('u', under)]
pipeline = Pipeline(steps=steps)
...
# Transform the dataset
X, y = pipeline.fit_resample(X, y)
```

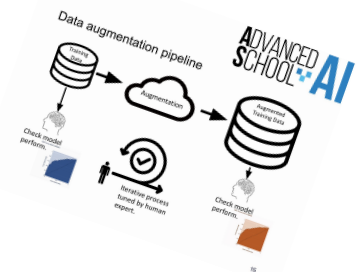
oversample to 10% of
majority
 $10\% \text{ of } 9900 = 990$

undersample majority
to final ratio of 50%
 $990/1980 = 0.5$

Before
`Counter({0: 9900, 1: 100})`

Now
`Counter({0: 1980, 1: 990})`





Some aug. technique for tabular data

```
# define model
model = DecisionTreeClassifier()

# evaluate pipeline
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(model, X, y, scoring='roc_auc', cv=cv, n_jobs=-1)
```

Before D.A.

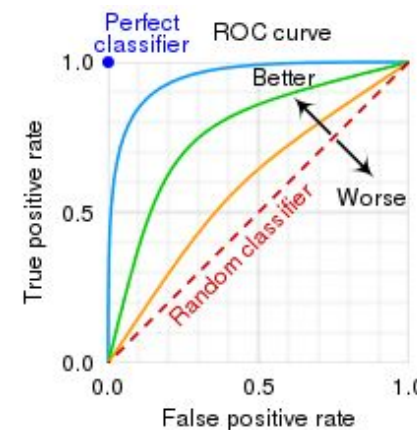
Mean
ROC AUC: 0.773

```
# define pipeline
model = DecisionTreeClassifier()
over = SMOTE(sampling_strategy=0.1)
under = RandomUnderSampler(sampling_strategy=0.5)
steps = [('over', over), ('under', under), ('model', model)]
pipeline = Pipeline(steps=steps)

# evaluate pipeline
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
scores = cross_val_score(pipeline, X, y, scoring='roc_auc', cv=cv, n_jobs=-1)
```

After D.A.

Mean
ROC AUC: 0.843



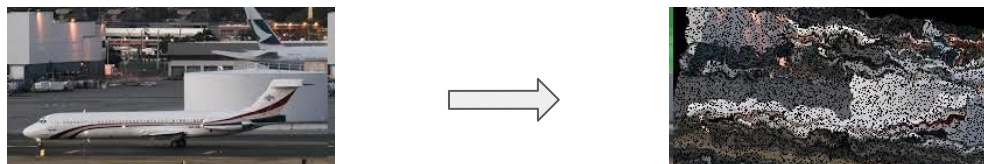
Some augmentation technique for tabular data



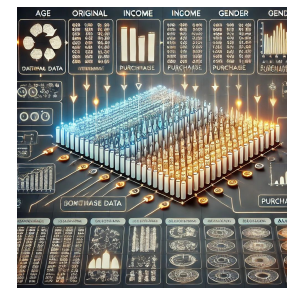
...time to code

Augmentation vs Synthetic data

- **Augmented data:** is driven from original data with some minor changes. In the case of image augmentation, we make geometric and color space transformations (flipping, resizing, cropping, brightness, contrast) to increase the size and diversity of the training set.

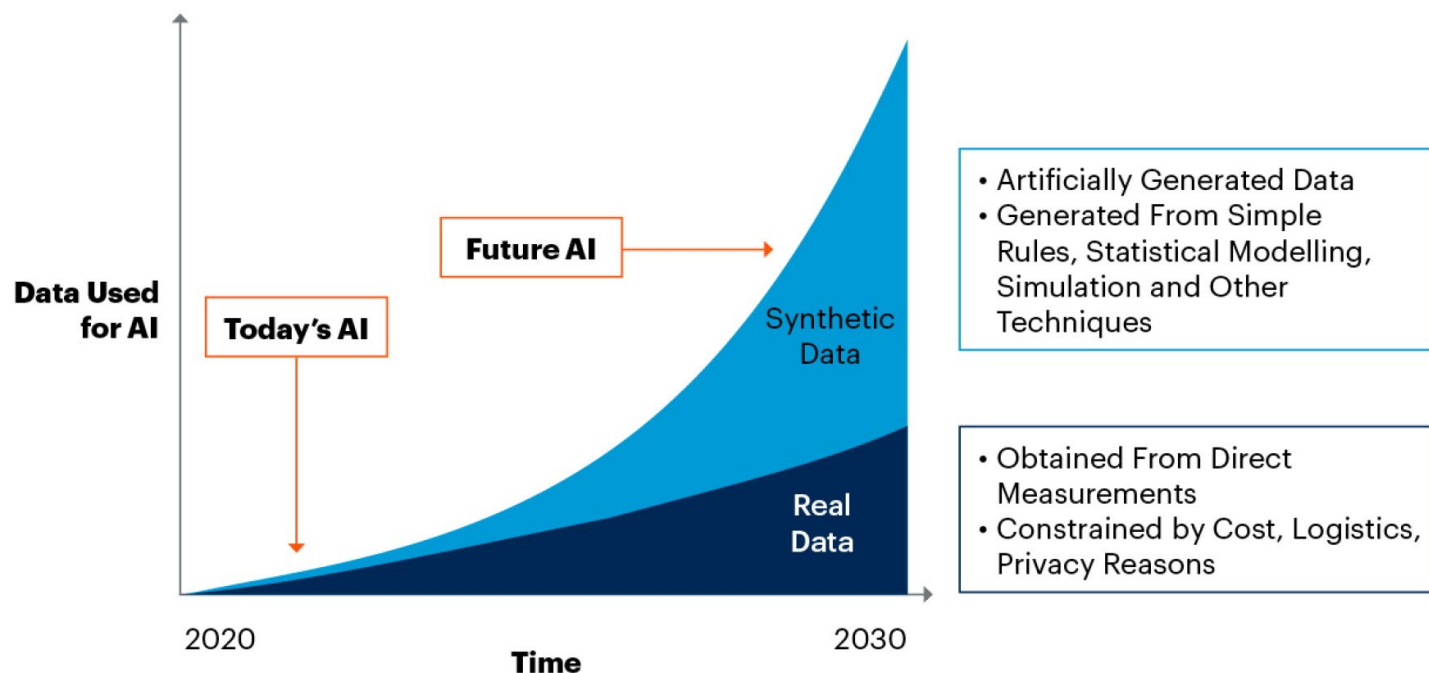


- **Synthetic data:** partly or completely artificial. It often uses DNNs (Deep Neural Networks) and GANs (Generative Adversarial Networks) to generate synthetic data.



Use of synthetic data

By 2030, Synthetic Data Will Completely Overshadow Real Data in AI Models



Source: Gartner
750175_C

Synthetic tabular data



...time to code

Nvidia Omniverse Replicator



LLM-based Data Augmentation

Comprehensive Survey on LLM-based Data Augmentation

A survey by Ding et al. (2024) examines how LLMs are transforming data augmentation strategies. It categorizes techniques into simple augmentation, prompt-based augmentation, retrieval-based augmentation, and hybrid methods. The paper also discusses challenges such as ensuring semantic fidelity and controlling the diversity of generated data.

Data Perspectives for LLM-Based Data Augmentation

- **Data Generation**
Leverage the few-shot capabilities of LLMs to generate large-scale synthetic datasets—particularly valuable in specialized or private domains and in tasks with large label spaces (e.g., medical dialogue summarization, intent-driven dialogues, data for information retrieval, chain-of-thought reasoning).
- **Data Labeling**
Use LLMs to annotate unlabeled datasets with accuracy comparable to or exceeding that of human annotators—especially effective in cross-lingual and multimodal scenarios.
- **Data Reformulation**
Transform existing data to create meaningful variations (e.g., counterfactual generation, paraphrasing) to enhance model robustness and generalization.
- **Co-Annotation**
Combine human and LLM efforts to reduce annotation costs and improve quality—often driven by LLM uncertainty or guided through human feedback.

LLM-Data Reformulation example

Reclamo in Input: "Cliente insoddisfatto per ritardo consegna"

Task:

"Genera 5 varianti realistiche di questo reclamo mantenendo il significato ma cambiando tono e contesto. Modifica il sentiment indicandolo fra parentesi."

1. "Buongiorno, segnalo che la consegna del mio ordine ha subito un ritardo significativo rispetto a quanto previsto. Questa situazione mi ha causato diversi disagi." *(negativo)*
2. "Salve, capisco che possano esserci imprevisti, ma il ritardo nella consegna è stato piuttosto lungo. Vorrei avere maggiori informazioni sullo stato dell'ordine." *(neutro)*
3. "Purtroppo devo esprimere la mia insoddisfazione: il pacco è arrivato molto oltre i tempi indicati e questo ha compromesso i miei piani." *(negativo)*
4. "La consegna è arrivata in ritardo, ma apprezzo comunque l'assistenza ricevuta. Spero che in futuro il servizio possa essere più puntuale." *(positivo)*
5. "Sono rimasto un po' deluso dal ritardo nella spedizione: mi aspettavo tempi più rapidi, anche se il prodotto è conforme alle aspettative." *(misto)*



LLM-based Data Augmentation

Key Challenges

- **Data Contamination**
Risk of augmented training data unintentionally overlapping with evaluation sets, compromising model assessment integrity.
- **Controllable Data Augmentation**
Difficulty in ensuring quality and diversity of synthetic data while maintaining control over specific attributes—risk of model collapse.
- **Culturally-Aware (Multilingual) Augmentation**
Incorporating cultural nuances and social norms in synthetic data generation for multilingual systems.
- **Multimodal Data Augmentation**
Aligning and integrating multiple modalities (text, images, audio, video, graphs) in a coherent and consistent way.
- **Privacy**
Protecting sensitive information—especially in regulated domains like healthcare (e.g., HIPAA compliance)—during synthetic data generation.

Discrimination Prevention and Bias Disambiguation

Identify a protected attribute

Select a sensitive variable (e.g., gender, race) that may be subject to discrimination.

Construct an “ideal world” dataset via counterfactual augmentation

For each real example, generate a synthetic data point with the same features and label but with the protected attribute value inverted, to simulate a fair world.

Rank and incrementally add synthetic points

Order the synthetic points according to their similarity to the real data (e.g., using k-means clustering) and progressively add them to create increasingly balanced datasets.

Evaluate with ML models and fairness metrics (SPD, AOD)

Train machine learning models on the augmented datasets and measure the impact on accuracy, Statistical Parity Difference (SPD), and Average Odds Difference (AOD) to reduce bias and identify its source (sampling or prejudice).

Hire or not

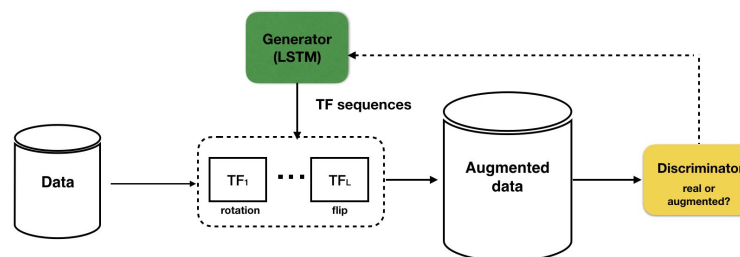
Occupation	Education	Gender	Decision
Clinician	College	Male	1
Clinician	College	Female	0
Clinician	High school	Male	1
Nurse	High school	Female	1
Nurse	College	Female	1
Nurse	College	Male	0
Nurse	High school	Male	0
Nurse	PhD	Female	1
Scientist	PhD	Male	1
Scientist	PhD	Male	1

Occupation	Education	Gender	Decision
Clinician	College	Female	1
Clinician	College	Male	0
Clinician	High school	Female	1
Nurse	High school	Male	1
Nurse	College	Male	1
Nurse	College	Female	0
Nurse	High school	Female	0
Nurse	PhD	Male	1
Scientist	PhD	Female	1
Scientist	PhD	Female	1

Open challenges

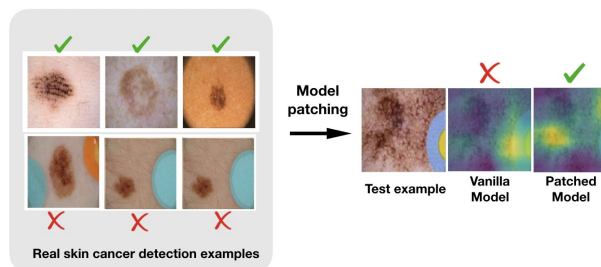
- Tuning: from manual to automated

A TF sequence generator is trained adversarially to produce augmented images that are realistic compared to training data.



- From coarse-grained to fine-grained model quality assurance

A standard model trained on a skin cancer dataset exhibits a subgroup performance gap between images of malignant cancers with and without colored bandages.



- From practical to theoretical understanding

Tuning: from manual to automated

- AutoAugment

The optimization problem is formulated as a Reinforcement Learning task, where an agent is trained to discover the best data augmentation policy.

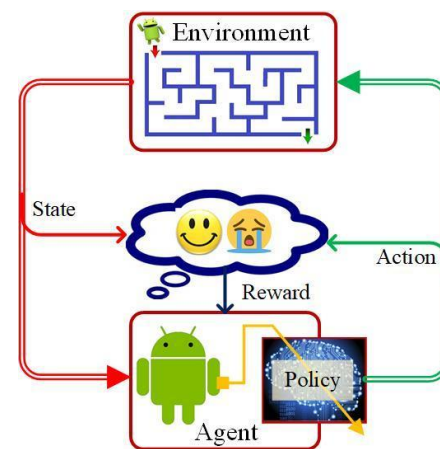
Action: selecting a single augmentation operation (type, probability, magnitude) at each step

Policy (trajectory): a sequence of actions forming a full augmentation policy

Environment: the child model training process

Reward: validation accuracy after training with the full policy

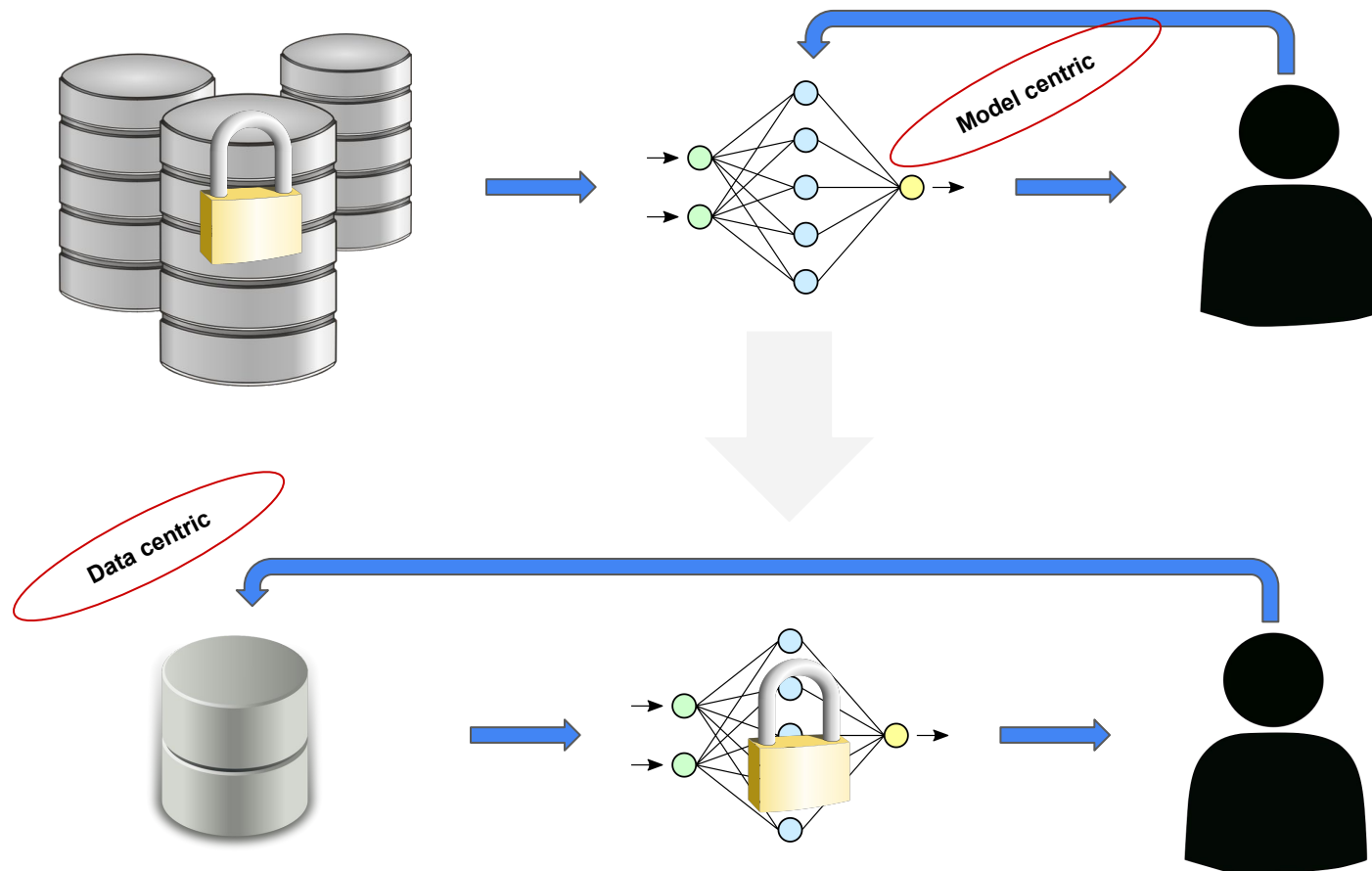
Controller: an RNN that autoregressively generates the sequence of actions.








- Fast AutoAugment

AutoAugment is computationally expensive, as it requires training many child models to evaluate candidate policies; to address this, subsequent methods such as **Fast AutoAugment** reformulate the search for optimal augmentation policies **using more efficient optimization strategies**, including density matching and Bayesian optimization with surrogate models (typically Gaussian Process), significantly reducing the computational cost.

New paradigm - Small Data



When does Data Augmentation really work?

Scenario	Augment	Recommended Technique
Small dataset		Traditional data augmentation (rotations, flips, noise) or synthetic data generated by LLMs
Imbalanced dataset		SMOTE, GANs, or other synthetic oversampling methods
Noisy dataset		Avoid blind augmentation; consider noise-aware methods
High-stakes task (medical, finance)		Controlled augmentation with careful validation
Already large dataset		Limited benefit; augmentation usually unnecessary

Is data augmentation a magic wand?

...no but it helps...



To take away...

- To train a good ML model needs good data in terms of quantity and variation
- To collect and label good data is extremely expensive
- Data augmentation is an iterative process that help to increase data quantity and variation starting from the original data.



To take away: benefits

- Reduce the operational cost of labeling and cleaning the dataset.
- Prevent models from overfitting.
- Improve the model accuracy.



To take away: limitations

- Usually the biases in the original dataset persist in the augmented data.
- Quality assurance for data augmentation is expensive.
- Finding an effective data augmentation approach can be challenging and introduce hyperparameter to tune.



Resources 1/3

Artificial Intelligence and Machine Learning Advancing in Aviation

link: <https://interactive.aviationtoday.com/avionicsmagazine/april-may-2021/artificial-intelligence-and-machine-learning-advancing-in-aviation/>

Contactless airport boarding

link: <https://www.airport-technology.com/analysis/contactless-airport-boarding-biometric-technology-with-sita/>

Deep Learning-Based Approach for Civil Aircraft Hazard Identification and Prediction

link: <https://www.readcube.com/articles/10.1109%2Faccess.2020.2997371>

Automating Data Augmentation: Practice, Theory and New Direction

link: <https://ai.stanford.edu/blog/data-augmentation/>

A Complete Guide to Data Augmentation

link: <https://www.datacamp.com/tutorial/complete-guide-data-augmentation>

imgaug

link: <https://imgaug.readthedocs.io/en/latest/index.html>

SMOTE for Imbalanced Classification with Python

link: <https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

SMOTE: Synthetic Minority Over-sampling Technique

link: <https://arxiv.org/abs/1106.1813>

imbalanced-learn

<https://imbalanced-learn.org/>

Receiver operating characteristic

link: https://en.wikipedia.org/wiki/Receiver_operating_characteristic

Resources 2/3

What Is Synthetic Data?

link: <https://blogs.nvidia.com/blog/2021/06/08/what-is-synthetic-data/>

sdv

link: <https://docs.sdv.dev/sdv>

Real-life Examples of Discriminating Artificial Intelligence

link: <https://towardsdatascience.com/real-life-examples-of-discriminating-artificial-intelligence-cae395a90070>

Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects

link: <https://arxiv.org/abs/1809.10790>

Learning to Compose Domain-Specific Transformations for Data Augmentation

link: <https://arxiv.org/abs/1709.01643>

AutoAugment: Learning Augmentation Policies from Data

link: <https://arxiv.org/abs/1805.09501>

Data Augmentation for Discrimination Prevention and Bias Disambiguation

link: http://krvarshney.github.io/pubs/SharmaZRBMV_aies2020.pdf

Resources 3/3

Model Patching: Closing the Subgroup Performance Gap with Data Augmentation

link: <https://arxiv.org/pdf/2008.06775.pdf>

Real-life Examples of Discriminating Artificial Intelligence

link: <https://towardsdatascience.com/real-life-examples-of-discriminating-artificial-intelligence-cae395a90070>

Data Augmentation for Discrimination Prevention and Bias Disambiguation

link: http://krvarshney.github.io/pubs/SharmaZRBMV_aies2020.pdf

Data Augmentation using Large Language Models: Data Perspectives, Learning Paradigms and Challenges

link: <https://arxiv.org/pdf/2403.02990>

Data Augmentation with Large Language Models: A Scaling Law-Guided Approach

link: <https://dl.acm.org/doi/epdf/10.1145/3787100>

Fast AutoAugment

link: <https://arxiv.org/abs/1905.00397>

Data-centric AI: Perspectives and Challenges

link: <https://arxiv.org/pdf/2301.04819.pdf>

Q&A



...thanks for your attention

Fabio M. D'Amore

f_d_amore@hotmail.com