L3.2 Utilization of machine learning methods in ionospheric research

Maria Kaselimi

NOA

23 September 2026

What can ML do in ionospheric research?

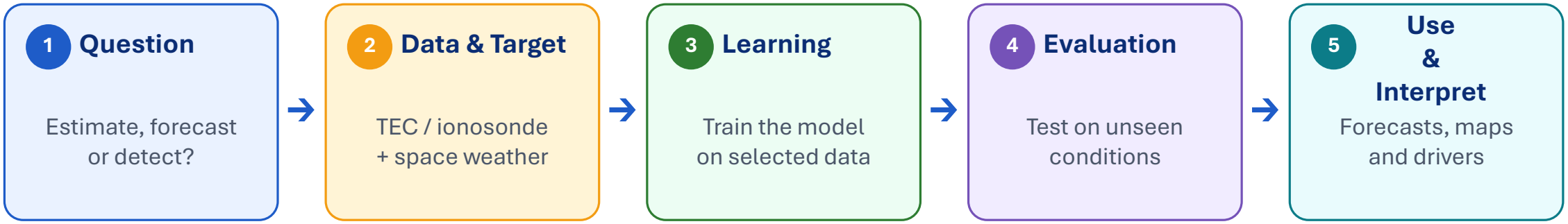
	ML task	Ionospheric question	Example
01	**Estimate** *Regression*	Infer ionospheric parameters from observations and drivers.	**VTEC or foF2**
02	**Forecast** *Time-series prediction*	Predict the next ionospheric state from recent sequences.	**1-24 h TEC**
03	**Reconstruct** *Spatial modeling*	Fill gaps where observations are sparse or missing.	**GIM gap-filling**
04	**Detect** *Classification*	Flag disturbed or anomalous ionospheric conditions.	**storm / scintillation**
05	**Interpret** *Explainable ML*	Identify which physical drivers control variability.	**F10.7, Kp, Dst**

Source: Mao, Hernandez-Pajares & Soja (2025), JGR: Machine Learning and Computation

Designing a Machine Learning System

for Ionospheric Research

A clear workflow: from geophysical question to validated model



Define the output/target clearly

Examples: next-hour VTEC, foF2, or storm-time anomaly label.

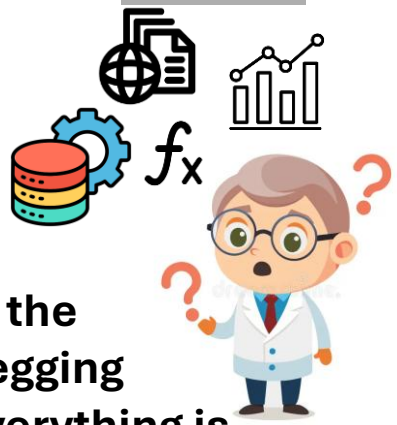
Avoid an easy but misleading test

Evaluate across quiet/storm days, seasons, latitude regions and solar activity.

Critical point Training and test data should represent the conditions where the model is expected to operate.

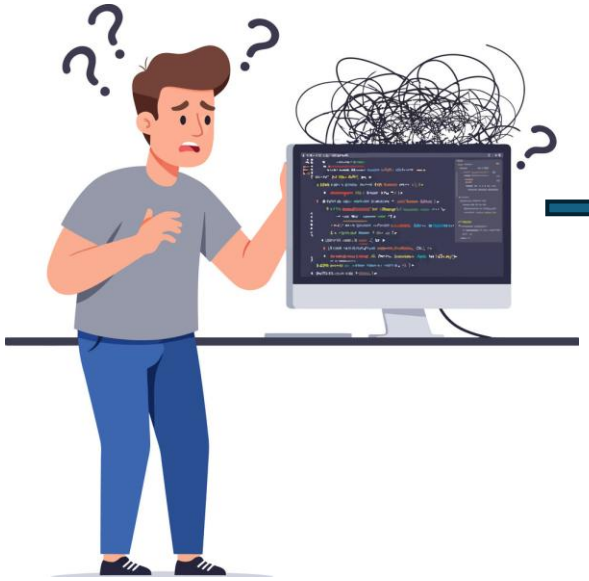
ML Researcher's Life

STEP 1

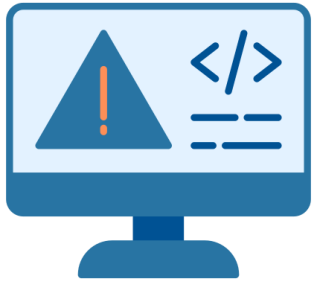


In the begging everything is complicated ...

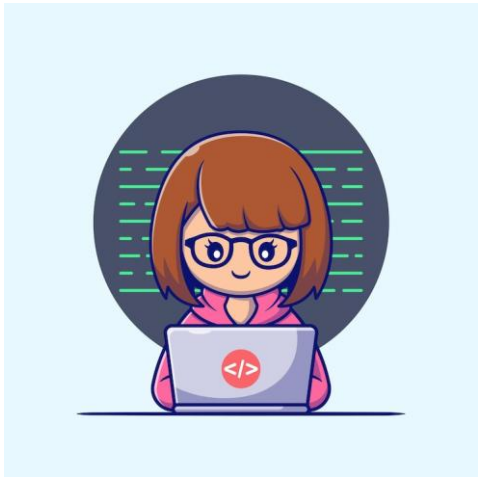
STEP 2



code breaks



STEP 3



code runs without errors



but are the results good enough and useful for my research?

STEP 4











go back

Types of Machine Learning



1. How does the model learn?

Supervised

Learns from examples with known outputs.

Example: VTEC prediction

Unsupervised

Finds structure without predefined labels.

Example: regime clustering

2. How complex should the model be?

Shallow ML

Feature-driven, fast, often easier to interpret.

RF • SVM • XGBoost

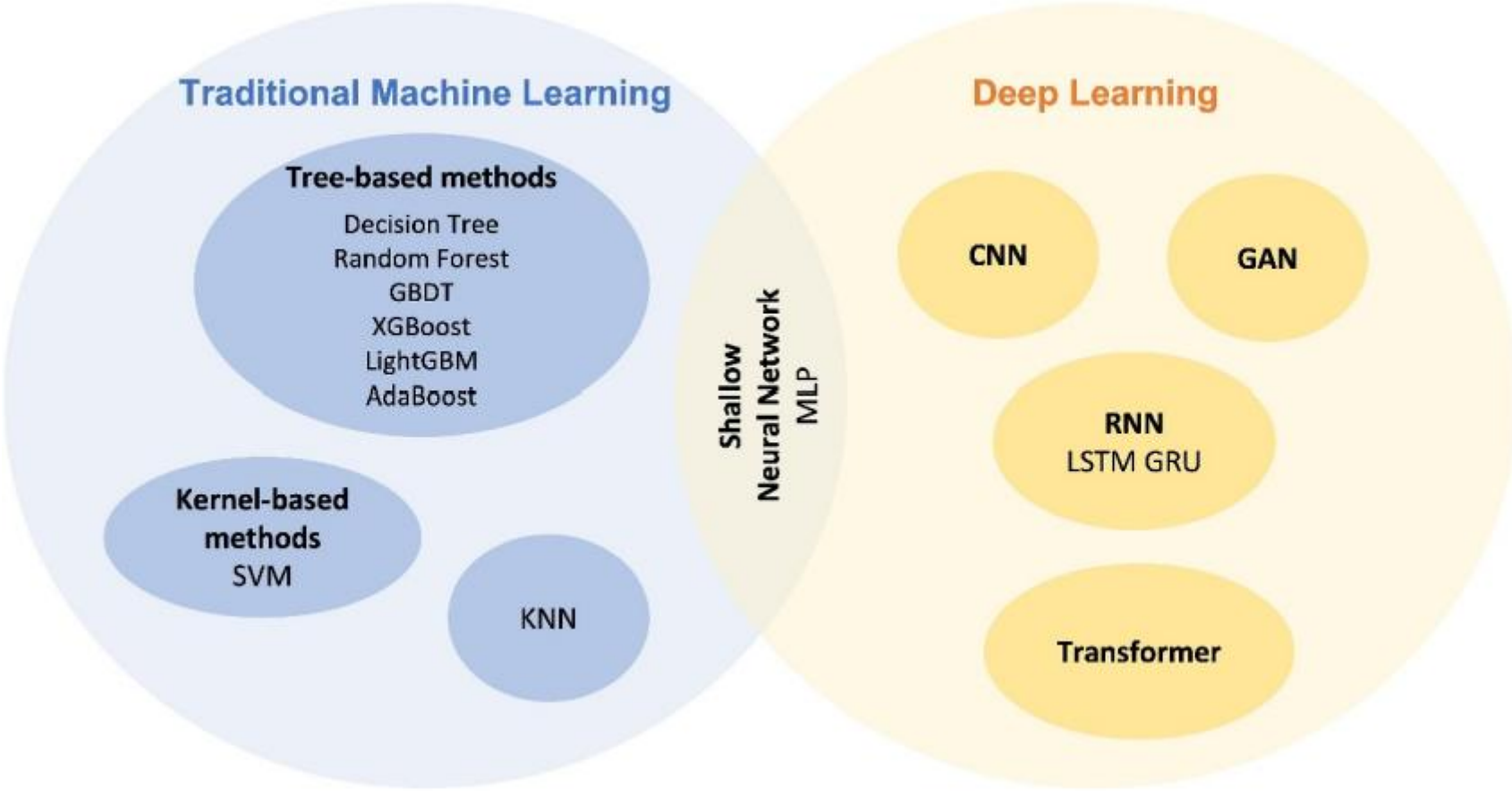
Deep Learning

Learns spatial–temporal representations from data.

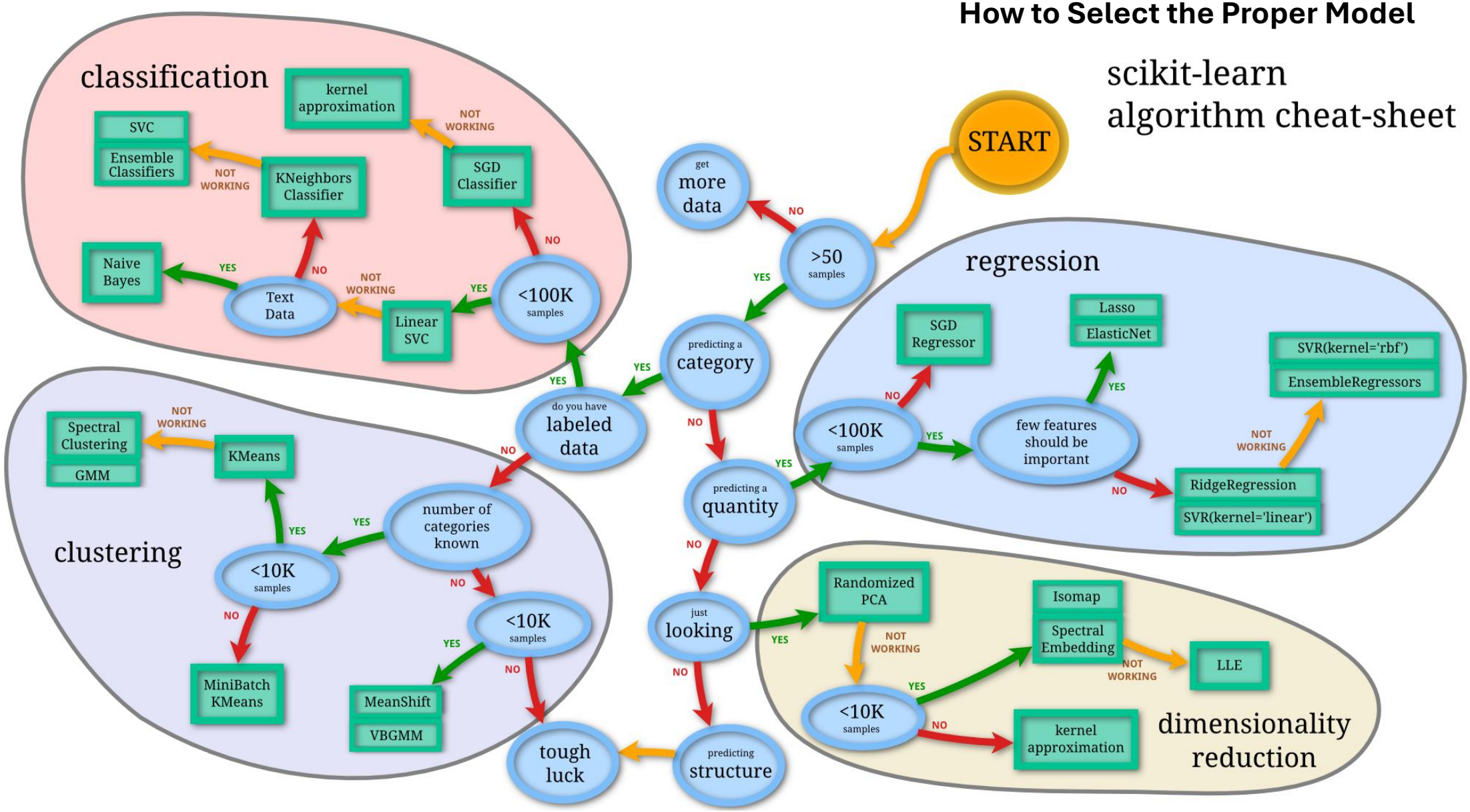
CNN • LSTM • Transformer

Key choice | Match the model to the target, data volume, interpretability needs, and real-time constraints.

Examples of ML Models for Ionospheric Research in Literature



How to Select the Proper Model



Data Processing Before Machine Learning

Basic steps to follow — and the key decisions to make before training

Steps

1 Define & collect data

Process: select observations such as GNSS TEC, ionosonde data, and space-weather indices.

Decision: what is the prediction target and required coverage?

2 Clean & align

Process: handle missing values, outliers, units, and inconsistent sampling.

Decision: interpolate, remove, or flag problematic records?

3 Create features

Process: derive useful variables such as local time, lags, moving averages, or solar drivers.

Decision: which features add information without leakage?

4 Split & evaluate

Process: separate training, validation, and test data before model comparison.

Decision: which split and metric reflect the real task?

Decision checklist

Before training, clarify:

- What is the target variable?
- Which inputs are available in practice?
- Are time and space resolutions aligned?
- How are gaps and outliers handled?
- Does the test set represent unseen conditions?

Good practice: prepare and split the data exactly as the model will receive it in real applications.

Data Types, Volume & Coverage

Before modeling: understand the form, scale, and representativeness of your data.

1 What type of data?

The data form guides the model family and preprocessing.

T — Structured / tabular
e.g., solar and geomagnetic indices

S — Time series
e.g., VTEC or foF2 sequences

M — Spatial grids / maps
e.g., TEC maps or GIM products

L — Labels / events
e.g., quiet vs storm conditions

2 How much — and how representative?

Volume is not only the number of samples. Coverage matters.

 # — Quantity
samples, stations, years

Δ — Resolution
sampling interval and spatial grid

C — Coverage
season, latitude, solar activity

B — Balance
rare events need special care

✓ Modeling decision

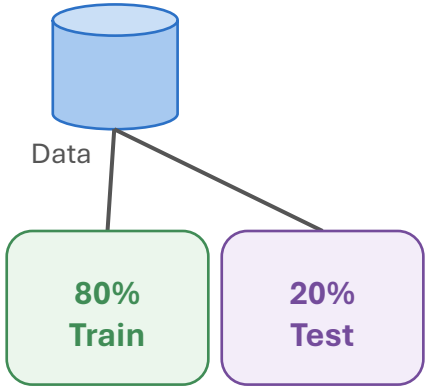
Small structured data → simpler models; long sequences or maps → deep learning; rare events → careful splitting and evaluation.

Model Evaluation

Model evaluation measures how well a machine learning model performs on new, unseen data. The goal is to estimate generalization performance and reduce the risk of overfitting.

Common validation strategies

1. Holdout Method (Train/Test Split)

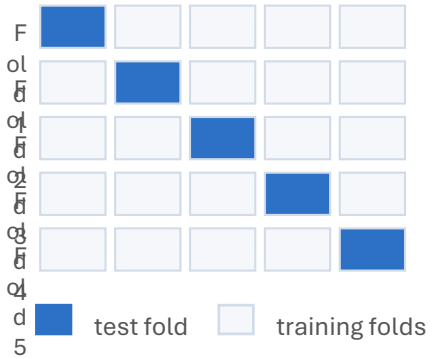


Split the dataset into a training set and a test set (for example 80/20).
Train the model on the training set.
Evaluate performance once on the test set.

Best when:
large datasets and a quick baseline estimate are needed.

Pros: simple and fast | Limitation: result depends on one split

2. K-Fold Cross-Validation



Split the data into K folds.
Use one fold as test data and the remaining K−1 folds as training data.
Repeat K times and average the scores.

Best when:
data are limited and a more stable estimate is needed.

Pros: more reliable estimate | Limitation: more computational cost

Takeaway: choose the validation strategy and metric according to the problem, dataset size, and class balance.

Classification Metrics

Classification metrics quantify how well a model predicts class labels on unseen data. The best metric depends on class imbalance and on the cost of false positives vs false negatives.

Accuracy

Overall fraction of correct predictions

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

Useful when classes are balanced.

Precision

Among predicted positives, how many are correct?

$$Precision = \frac{TP}{TP + FP}$$

Important when false positives are costly.

Recall

Among true positives, how many are found?

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

Important when missed positives are costly.

F1 Score

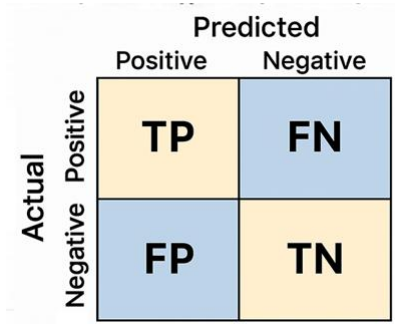
Harmonic mean of precision and recall

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Good summary when classes are imbalanced.

Confusion Matrix

Shows correct and incorrect predictions by class.



TP: correctly predicted positive

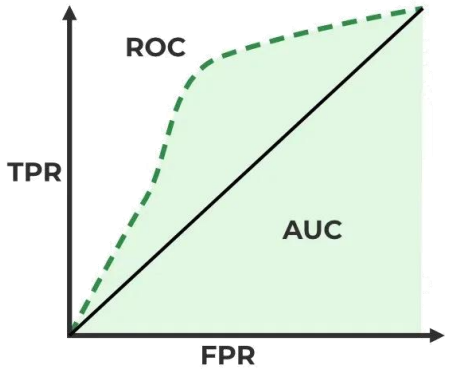
TN: correctly predicted negative

FP: predicted positive but actually negative

FN: predicted negative but actually positive

ROC Curve and AUC

Evaluates the model across different decision thresholds.



ROC plots true positive rate (TPR) vs false positive rate (FPR).
AUC is the area under the ROC curve.
AUC = 0.5: random model | AUC = 1.0: perfect model
Higher AUC means better class separation.

$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

$$\text{True Positive Rate(TPR)} = \frac{TP}{TP + FN}$$

Takeaway: no single metric is sufficient. Choose metrics according to the task, class imbalance, and the cost of FP vs FN.

Regression Metrics

Regression models predict continuous values. Error metrics quantify the difference between predicted and observed values and help compare model performance.

1. Mean Absolute Error (MAE)

Average absolute difference between true and predicted values

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Easy to interpret: in the same units as the target.
Less sensitive to outliers than MSE or RMSE.

2. Mean Squared Error (MSE)

Average squared difference between true and predicted values

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Penalizes large errors more strongly.
Often used as a training loss function.

3. Root Mean Squared Error (RMSE)

Square root of the mean squared error

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Same units as the target variable.
Sensitive to large errors; useful when big errors matter more.

4. R^2 Score (Coefficient of Determination)

Measures how much of the variance is explained by the model

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

y_i: observed value
$\hat{y}_i$: model prediction
$\bar{y}$: mean observed value

$R^2 = 1$: perfect fit; $R^2 = 0$: no improvement over predicting the mean.
Can be negative when the model performs poorly.

Quick interpretation

- **MAE** — easy to interpret
- **MSE** — penalizes large errors
- **RMSE** — same units as target
- **R^2** — explained variance

Takeaway: lower MAE, MSE, and RMSE indicate better predictions, while higher R^2 indicates better explanatory power.

Python



python + colab is a free, cloud-based platform that lets you write and execute Python code directly in your web browser.

Recommended Resources

For the summer school: Utilization of Machine Learning Methods in Ionospheric Research

1. ML foundations

- **Google ML Crash Course** — train/test split, features and overfitting
- **scikit-learn User Guide** — preprocessing, pipelines and evaluation
- **Stanford CS229 notes** — mathematical background for supervised learning

2. Ionospheric and space-weather data

- **IGS / CDDIS IONEX GIMs** — global TEC maps for spatial/temporal modeling
- **NASA OMNIWeb** — F10.7, Dst, Kp, IMF and solar-wind drivers
- **GIRO / DIDBase** — ionosonde parameters such as foF2 and hmF2

3. Tools for hands-on work

- **Jupyter + Python** — interactive analysis and reproducible notebooks
- **pandas / NumPy / xarray** — tabular, array and gridded geophysical data
- **scikit-learn + PyTorch** — baselines first; deep learning when needed

4. Advanced reading and standards

- **Mao et al. (2025) review** — ML-based ionospheric spatial/temporal modeling
- **IRI model** — empirical ionosphere baseline for comparison
- **ESA GSSC: IONEX format** — structure of TEC map files

Suggested path: learn the ML basics → download TEC and space-weather data → build a baseline model → evaluate and compare with advanced models.

How Machine Learning Is Evolving

- **Classical ML**

  Hand-crafted features + relatively simple models

  → RF, SVM, XGBoost

- **Deep Learning**

  Models automatically learn useful feature representations / embeddings from data

  → CNNs, RNNs, Transformers

- **Self-Supervised & Transfer Learning**

  Learn reusable representations from large amounts of unlabeled data

  → less dependence on manually labeled datasets

- **Foundation & Multimodal Models**

  Pretrained models learn rich, transferable embeddings across multiple tasks and data types

  → images, time series, spatial data, text, scientific observations

- **Scientific & Trustworthy AI**

  Combine advanced AI with physics, domain knowledge, uncertainty and interpretability

  → more reliable and scientifically meaningful models

Hands-on Exercise 1 — Classical Forecasting of foF2

The problem

Can we predict foF2 from its recent temporal evolution?

This is the starting point for a forecasting study.

The goal is to build a clear and interpretable baseline.

Data used

foF2 time series

2023–2026 observations

5-minute temporal resolution

Continuous segment after handling short gaps

Researcher decisions

What is the prediction target?

How much past history should be used?

How should missing values be treated?

Why must the split be chronological?

How do AR and ARIMA perform as baselines?

Key message: A good forecast begins with the right problem definition, careful preprocessing, and a strong baseline.

Hands-on Exercise 2 — Deep Learning for Multivariate Forecasting

The problem

Can forecasting improve when we add external drivers (multiple parameters)?

The ionosphere depends on both its recent state and geomagnetic activity.

Now the task becomes a multivariate sequence-learning problem.

Data used

foF2

Geomagnetic indices: AE, AL, AO, AU

Time-of-day encoding

Sliding windows of past observations

Researcher decisions

Which inputs are truly informative?

How long should the input window be?

Should we predict one step or a short sequence?

When is an MLP enough, and when is an LSTM better?

Does higher complexity really improve the forecast?

Key message: Deep learning is useful only when the chosen inputs, model, and evaluation strategy match the scientific question.

Hands-on Exercise 3 — Detecting Active Ionospheric Windows

Research question: Can we detect active ionospheric/geomagnetic windows from recent observations?

Problem setup

- Move from regression to classification
- Each 3-hour window receives a binary label
- Active means ASY-H exceeds a percentile threshold
- The task is to flag active versus quiet windows

Data used

- foF2 time series
- Geomagnetic indices: AE, AL, AO, AU
- ASY-H used to define the activity label
- 5-minute data and 3-hour sliding windows

Researcher decisions

- Which threshold defines “active”?
- How to avoid label and scaling leakage?
- Are the classes balanced?
- Which error matters more: false alarms or missed events?
- Compare MLP and LSTM with precision, recall, F1 and ROC-AUC

Key message: Detection depends on the label definition, class balance, decision threshold and metric — not only on the model.

Hands-on Exercise 4 — Event Forecasting from Pre-Storm Windows

Research question: Can pre-event ionospheric and geomagnetic conditions distinguish storm onset from quiet periods?

Problem setup

- Build an event-based classification problem
- Storm samples use the 3 hours before storm onset
- Quiet samples come from clean periods away from storms
- The model learns storm-window versus quiet-window patterns

Data used

- Storm labels / event times
- foF2 and geomagnetic indices: AU, AE, AL
- 3-hour windows at 5-minute resolution
- Balanced storm and quiet examples for the tutorial

Researcher decisions

- How is a storm event defined?
- How far before onset should the window start?
- How should quiet control periods be selected?
- With few events, use stratified cross-validation
- Compare simple classifiers such as KNN and Random Forest

Key message: Event forecasting is mainly a data-design problem: event definition, control sampling and validation determine what the results mean.

Thank you