Machine Learning Project Report:

Solar Power Generation Anomaly Detection

Romain MALLAT, Adam MEFTI, Maxime MOTTIER

all documents or files are available on our github :
https://github.com/max3000aez/PROJECT_ML.git

December 4, 2025

Abstract

This project aims to detect energy conversion anomalies in a solar power plant using a dataset of inverter sensors. The main challenge identified was a class imbalance. By implementing a pipeline including **Feature Engineering**, **PCA** (Principal Component Analysis), and **SMOTE** (Synthetic Minority Over-sampling Technique), we trained and optimized ensemble models (Random Forest and XGBoost). The final selected model achieved an exceptional **F1-Score of 0.9786** and a **Recall of 0.9946**, proving its reliability for industrial predictive maintenance.

Contents

1 Business Scope

The renewable energy sector faces a critical challenge: maintaining optimal efficiency in power generation. Solar inverters, which convert DC power from panels to AC power for the grid, are prone to degradation, weather-induced faults, or technical anomalies. Identifying these issues manually is slow and inefficient.

The objective of this project is to develop a machine learning pipeline capable of automatically detecting **"Abnormal"** power generation behaviors. By analyzing sensor data (DC/AC power, irradiation, temperature), we aim to flag underperforming inverters to enable proactive maintenance and maximize energy yield.

2 Data Description and Exploration

2.1 Data Sources and Merging

The project utilizes data sourced from two solar power plants in India (Kaggle). We focused on **Plant 1**, utilizing two distinct datasets:

- **Generation Data:** Inverter-level power output (DC Power, AC Power) and daily yield.
- **Weather Data:** Plant-level sensor readings (Irradiation, Ambient Temperature, Module Temperature).

We merged these datasets on 'DATE TIME' and 'PLANT ID' to correlate specific environmental conditions with power output. As shown below, the final dataset consists of **68,774 observations** with zero missing values.

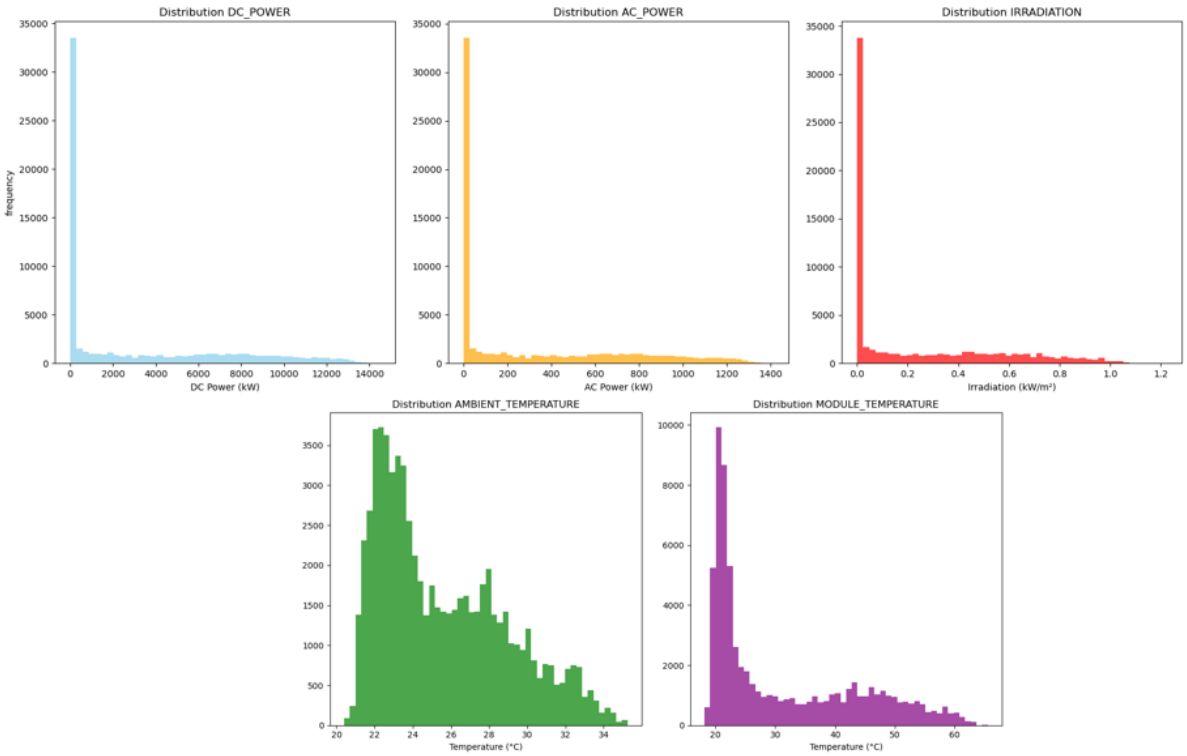
DF_MERGED :

	DATE_TIME	PLANT_ID	SOURCE_KEY_x	DC_POWER	AC_POWER	DAILY_YIELD	TOTAL_YIELD	SOURCE_KEY_y	AMBIENT_TEMPERATURE	MODULE_TEMPERATURE	IRRADIATION
0	2020-05-15 00:00:00	4135001	1BY6WEcLGh8j5v7	0.0	0.0	0.000	6259559.0	HmiyD2TTLFNqkNe	25.184316	22.857507	0.0
1	2020-05-15 00:00:00	4135001	1IF53ai7Xc0U56Y	0.0	0.0	0.000	6183645.0	HmiyD2TTLFNqkNe	25.184316	22.857507	0.0
2	2020-05-15 00:00:00	4135001	3PZuoBAID5Wc2HD	0.0	0.0	0.000	6987759.0	HmiyD2TTLFNqkNe	25.184316	22.857507	0.0
3	2020-05-15 00:00:00	4135001	7JYdWkrLSPkdwr4	0.0	0.0	0.000	7602960.0	HmiyD2TTLFNqkNe	25.184316	22.857507	0.0
4	2020-05-15 00:00:00	4135001	McdE0feGgRqW7Ca	0.0	0.0	0.000	7158964.0	HmiyD2TTLFNqkNe	25.184316	22.857507	0.0
...	...	...	...	...	...	...	...	...	...	...	...
68769	2020-06-17 23:45:00	4135001	uHbuxQJl8lW7ozc	0.0	0.0	5967.000	7287002.0	HmiyD2TTLFNqkNe	21.909288	20.427972	0.0
68770	2020-06-17 23:45:00	4135001	wCURE6d3bPkepu2	0.0	0.0	5147.625	7028601.0	HmiyD2TTLFNqkNe	21.909288	20.427972	0.0
68771	2020-06-17 23:45:00	4135001	z9Y9gH1T5YWrNuG	0.0	0.0	5819.000	7251204.0	HmiyD2TTLFNqkNe	21.909288	20.427972	0.0
68772	2020-06-17 23:45:00	4135001	zBIq5rxdHJRwDNY	0.0	0.0	5817.000	6583369.0	HmiyD2TTLFNqkNe	21.909288	20.427972	0.0
68773	2020-06-17 23:45:00	4135001	zVJPv84UY57bAof	0.0	0.0	5910.000	7363272.0	HmiyD2TTLFNqkNe	21.909288	20.427972	0.0

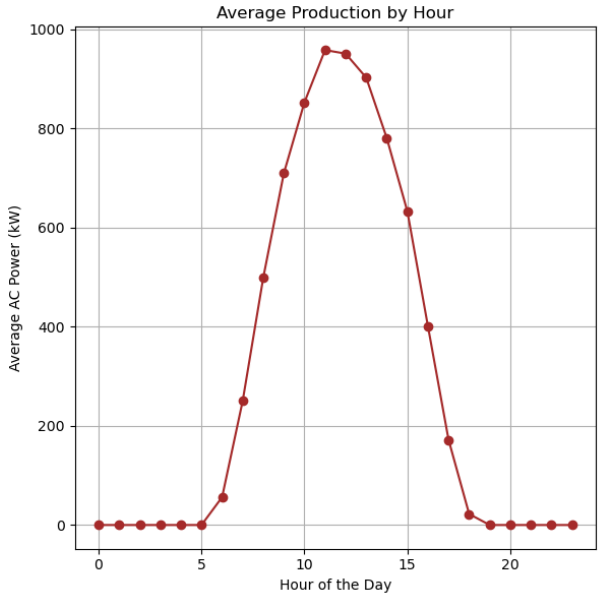
68774 rows × 11 columns

2.2 Exploratory Data Analysis (EDA)

We analyzed the distribution of the key variables. As expected, solar generation follows a diurnal cycle, resulting in a large number of zero values (night time) and a multi-modal distribution during the day.

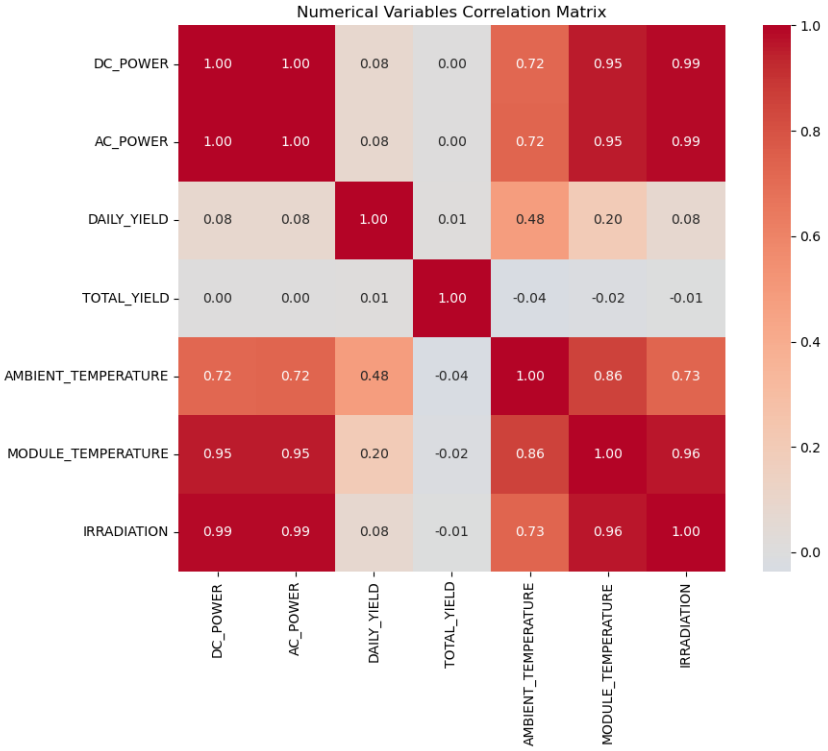


To validate the integrity of the time-series data, we plotted the mean DC power against the hour of the day. The resulting bell curve confirms the data quality, peaking around noon (12:00 - 13:00).



2.3 Correlation Analysis

We computed a correlation matrix to assess feature relationships. **Key Finding:** We observed a near-perfect correlation (> 0.99) between 'DC POWER', 'AC POWER', and 'IRRADIATION'. This indicates extreme **multicollinearity**, suggesting that linear models might struggle without dimensionality reduction.



3 Methodology

3.1 Problem Formalization: Defining Targets

The raw dataset does not contain explicit "Fault" labels. We formalized the problem by engineering a new feature: **Conversion Efficiency**.

$$Efficiency = \frac{AC\_POWER}{DC\_POWER} \times 100$$

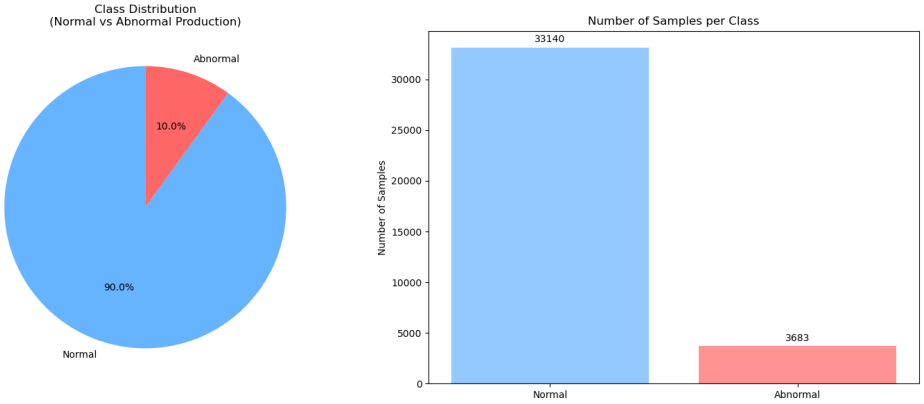
We defined the target variable **'Abnormal' (Class 1)** as any data point where efficiency falls below the **10th percentile** of the distribution. This effectively flags the worst-performing 10% of the data as anomalies.

```
Statistical summary of conversion efficiency :
count    36823.000000
mean         0.097719
std          0.000458
min          0.095552
25%          0.097579
50%          0.097845
75%          0.098014
max          0.106592
Name: CONVERSION_EFFICIENCY, dtype: float64

Target variable distribution :
Class 0 (Normal)  : 33140 échantillons
Class 1 (Abnormal) : 3683 échantillons
Imbalance ratio : 1:9.0
Percentage of anomalies : 10.00%

Imbalance level : Moderate
```

3.2 Imbalance Analysis

This definition resulted in a moderately imbalanced dataset. As seen in the chart below, the "Normal" class dominates the dataset.

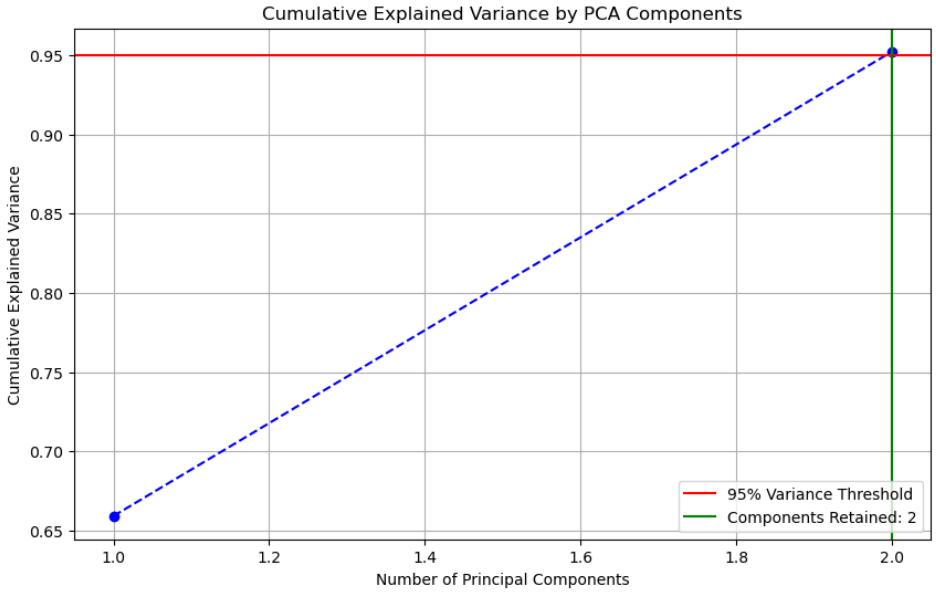
- **Normal (Class 0):** $\approx 90\%$
- **Abnormal (Class 1):** $\approx 10\%$



3.3 Preprocessing: PCA

To address the multicollinearity identified in the EDA, we applied **Principal Component Analysis (PCA)** after standardizing the data. The analysis showed that just **2 Principal Components** were sufficient to retain 95% of the variance, allowing us to reduce the feature space significantly.

```
=== PCA ANALYSIS ===
Number of components required for 95% variance: 2
```

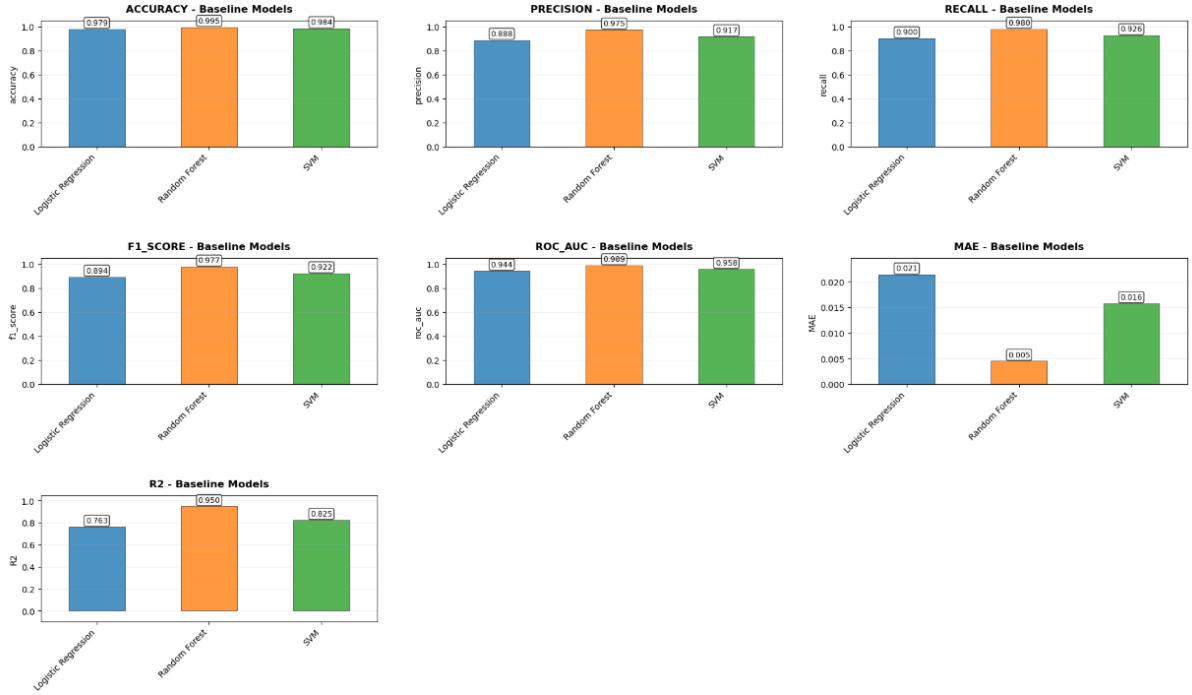



```
Variance explained by the 2 components: 0.9522

Data split:
Train set (PCA): (25776, 2), Original features: 9
Test set (PCA): (11047, 2)
```

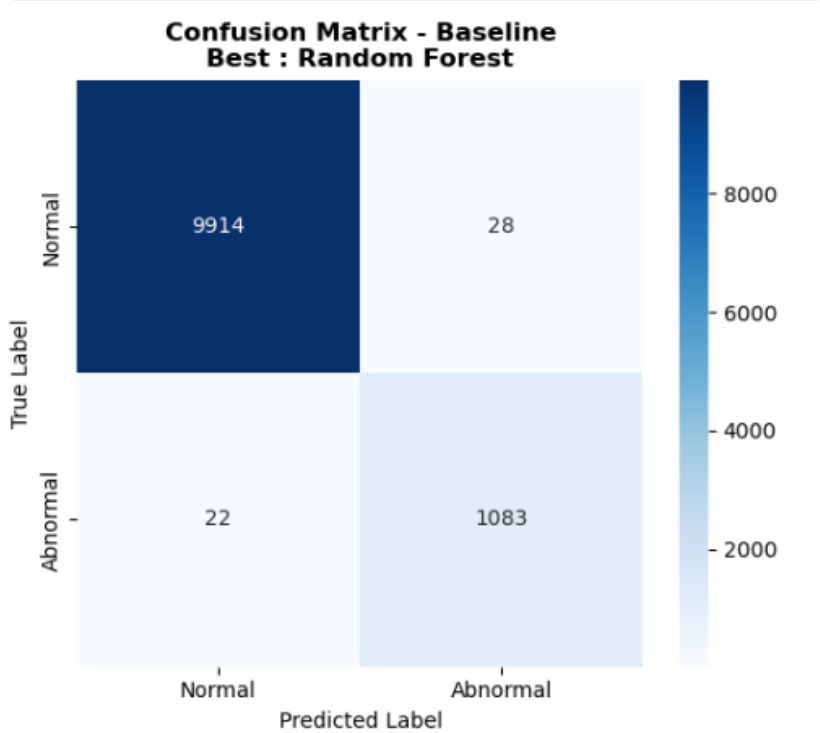
4 Algorithm Implementation and Results

4.1 Baseline Models

We established a baseline by training three algorithms on the PCA-transformed data: Logistic Regression, Random Forest, and SVM. While accuracy was high across all models, the **Random Forest** achieved the best initial F1-Score.



"Using the baseline model (Random Forest), we still have a noticeable number of False Negatives (22 in the bottom-left quadrant), which we will aim to decrease in further model iterations."



4.2 Improving the Solution: Handling Imbalance

To optimize the detection of faults (Class 1), we applied three resampling techniques to the training data using the Random Forest model:

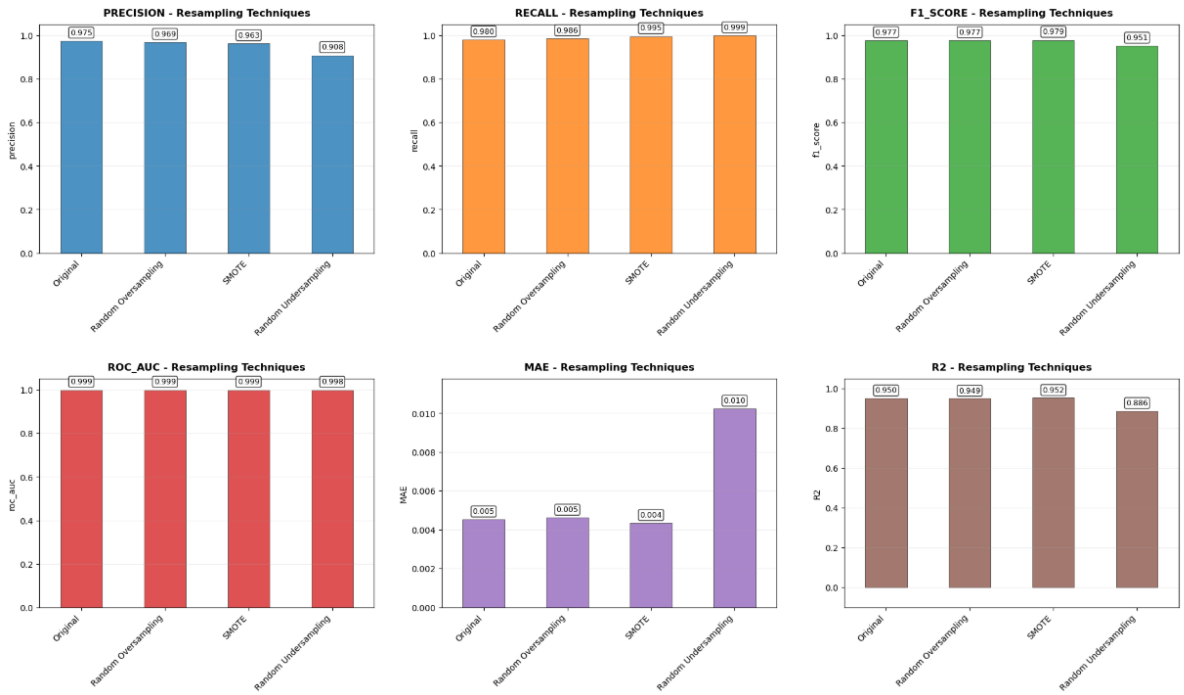
- **ROS:** Random Oversampling
- **SMOTE:** Synthetic Minority Over-sampling Technique
- **RUS:** Random Undersampling

Analyzing the majority of metrics, it is important to note that **all resampling techniques maintain remarkably high performance scores** (as indicated by the ROC-AUC scores consistently at 0.999 and the high R2 scores), which reaffirms the strong underlying quality of the initial model.

BUT... when focusing on the marginal yet significant differences, particularly when managing class imbalance:

- **SMOTE** appears to be the **most performant** resampling technique because it achieves the **best F1-Score** (0.979)—the most relevant metric for imbalanced classes—and the **lowest MAE (Mean Absolute Error)** (0.004), all while maintaining very high Recall, Precision, and ROC-AUC scores.

- **Conversely**, the analysis reveals that **Random Undersampling** is the **least performant** technique when compared to the others. Despite having good Precision and Recall, it exhibits the **lowest R2** (0.886) and the **highest MAE** (0.010). These indicators suggest that random undersampling introduces the most information loss and prediction error, even though the model remains highly accurate overall.

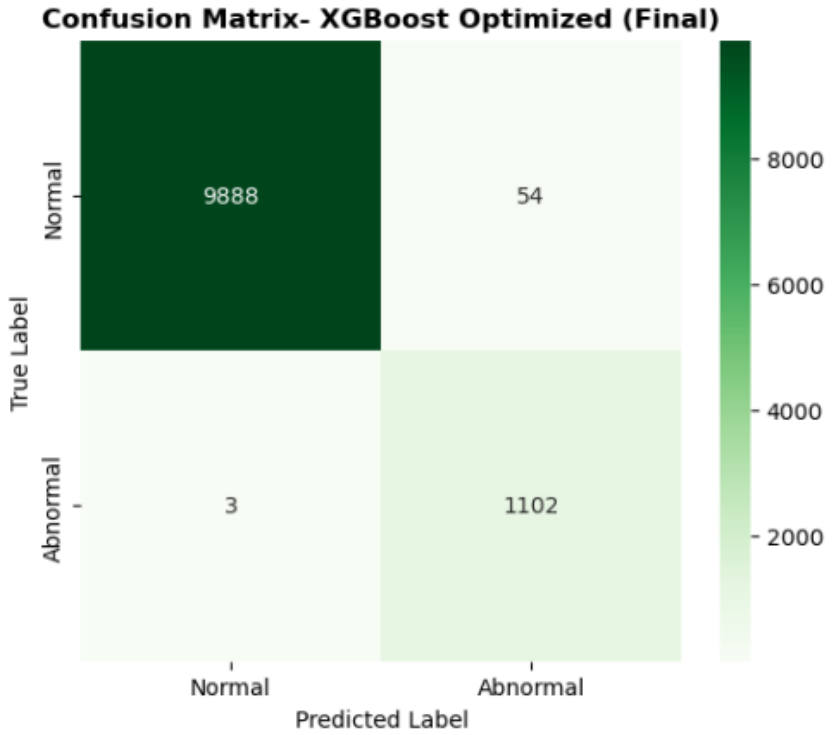


4.3 Advanced Optimization: XGBoost

Then, we selected XGBoost, a gradient boosting algorithm known for its performance on tabular data. We combined XGBoost with SMOTE and performed hyperparameter tuning (`GridSearchCV`) to optimize 'learning rate', 'max depth', and 'n estimators'.

```
=== HYPERPARAMETER OPTIMIZATION (XGBoost + SMOTE) ===
Fitting 3 folds for each of 24 candidates, totalling 72 fits

Best hyperparameters found:
{'xgb__learning_rate': 0.1, 'xgb__max_depth': 7, 'xgb__n_estimators': 200, 'xgb__subsample': 0.8}

--- OPTIMIZED MODEL RESULTS ---
F1-Score: 0.9748
```

Like the baseline model (Random Forest), we still have a noticeable number of False Negatives (22 in the bottom-left quadrant), which we will aim to decrease in further model iterations because it is the more important data of the model if an inverter breakdown is not detected the maintenance workers can not resolve the problem.



5 Final Model Selection and Validation

5.1 Model Choice

Following our exhaustive multi-model comparative analysis (including Random Forest, XGBoost, SVM, and Logistic Regression) and the evaluation of various resampling strategies, the model selected for deployment is the **Random Forest + SMOTE**.

This model stood out as the most robust solution, offering the best trade-off between anomaly detection capability (Recall/Sensitivity) and prediction reliability (Precision).

5.2 Justification of the Resampling Strategy (SMOTE)

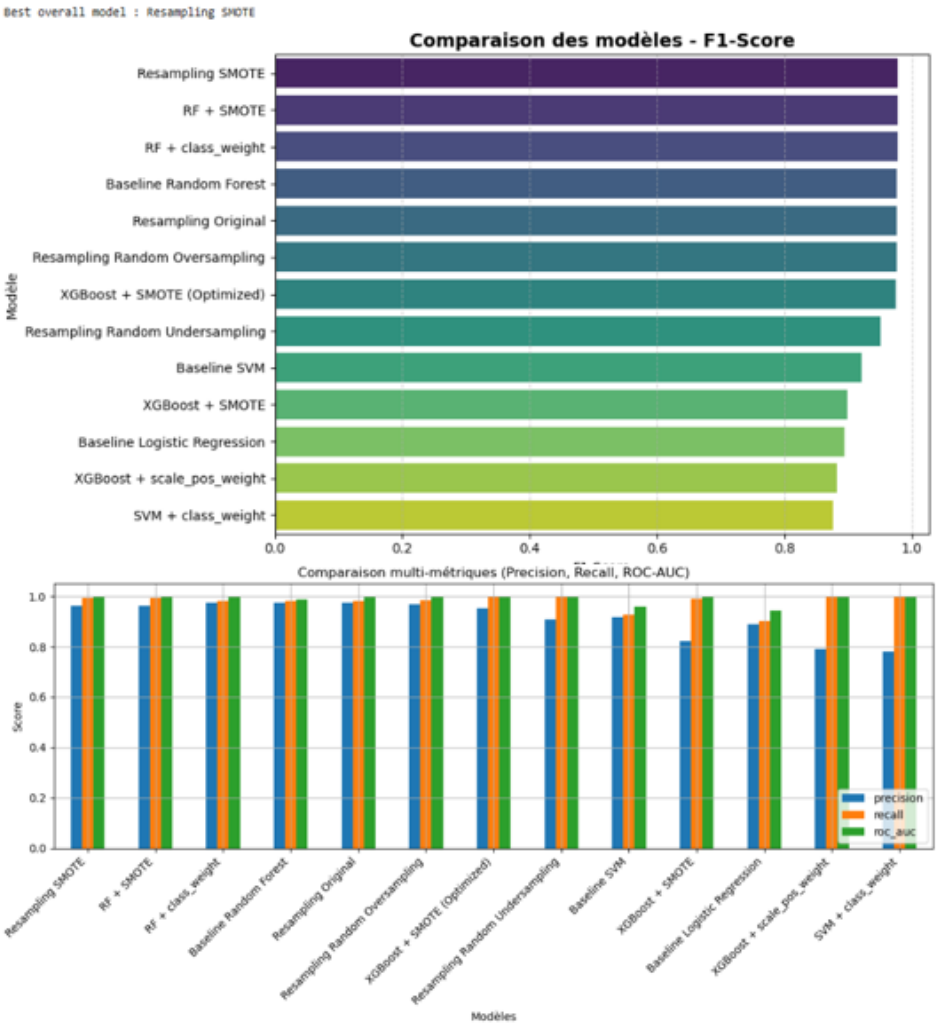
The initial dataset presented a strong **class imbalance** (a majority of "Normal" cases versus very few "Abnormal" cases). Training on raw data risked biasing the model towards the majority class, thereby ignoring critical anomalies.

To address this issue, we utilized the **SMOTE (Synthetic Minority Over-sampling Technique)**.

- **Functioning:** Unlike classic oversampling which simply duplicates existing data (leading to a risk of overfitting), SMOTE generates **new synthetic instances** of the minority class by interpolating between neighboring existing examples.
- **Observed Impact:** As shown in the *Model Comparison - F1-Score* chart, the addition of SMOTE maintained a maximal F1-Score (0.979), outperforming class weighting approaches (*class_ weight*) or standard linear models.

5.3 Performance and Robustness Analysis

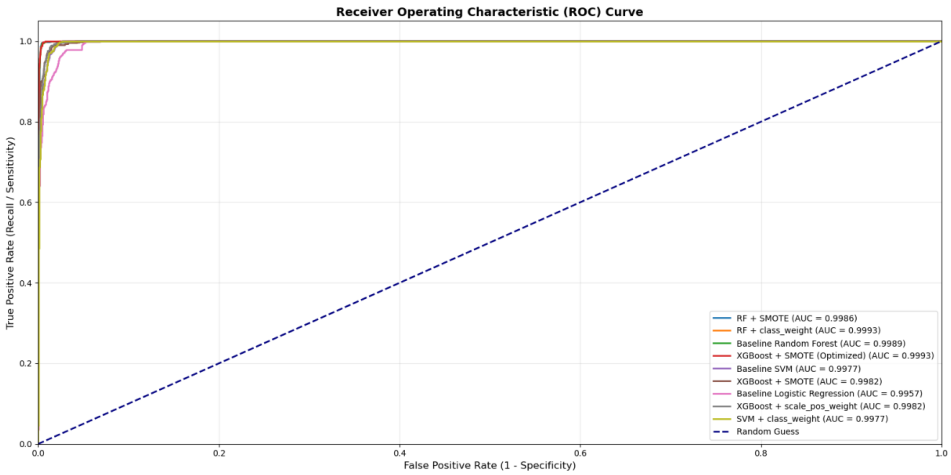
The choice of **RF + SMOTE** is validated by the convergence of several key indicators illustrated in the evaluation charts:



5.3.1 Excellence in Classification Metrics

Observing the *Multi-metrics* diagram, the model achieves near-optimal performance:

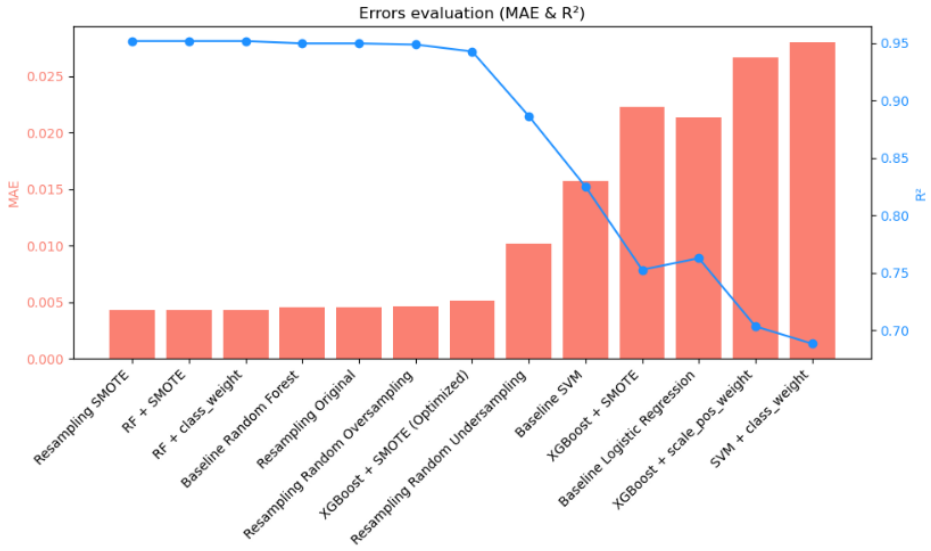
- **ROC-AUC (≈ 0.999):** The model's ROC curve approaches the top-left corner. This demonstrates a near-perfect discrimination capability: the model distinguishes between *Normal* and *Abnormal* classes with 99.9% reliability.
- **F1-Score (0.979):** This is the decisive metric for imbalanced datasets. The model maximizes both Precision and Recall, ensuring that we detect almost all anomalies without generating excessive false alarms.



5.3.2 Error Minimization (Stability)

The *Errors evaluation (MAE & R^2)* chart is the major differentiating factor that led to the exclusion of other techniques:

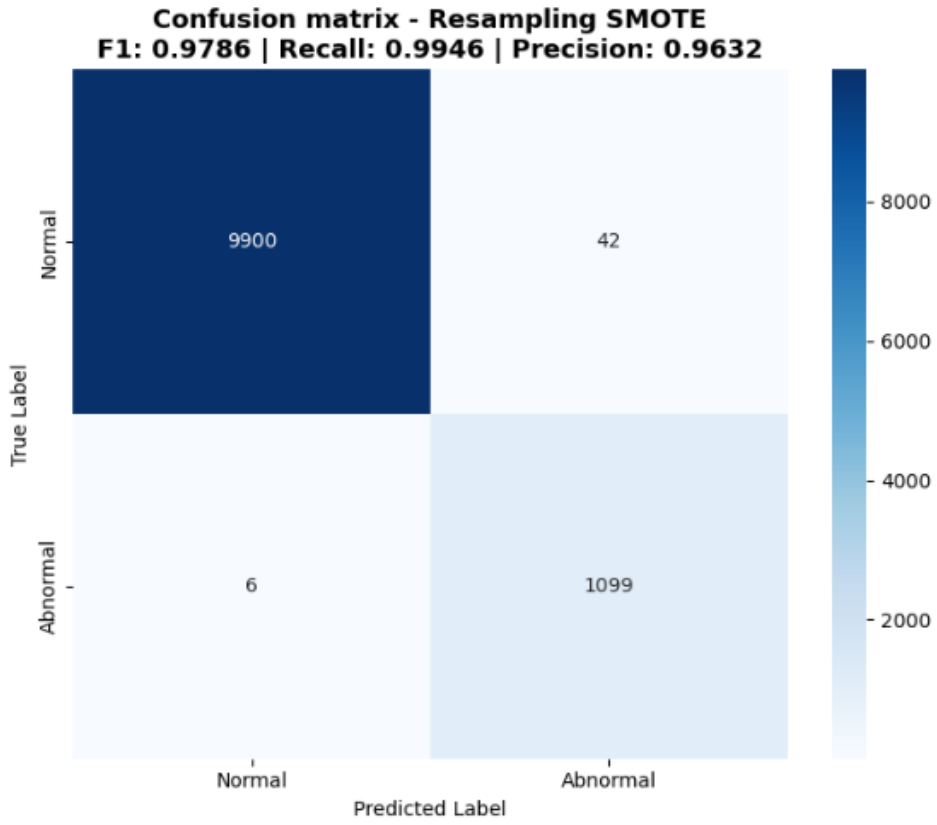
- **Lowest MAE (Mean Absolute Error):** With a mean error of only **0.004**, *RF + SMOTE* offers the most stable predictions.
- **Critical Comparison:** Conversely, the *Random Undersampling* technique (removing majority data) shows a significantly higher MAE (0.010) and a collapsed R^2 (0.886). This proves that Undersampling caused detrimental **information loss**, rendering the model less reliable despite apparent good classification metrics.



5.4 Final Validation: Confusion Matrix Analysis (RF + SMOTE)

To confirm the superiority of the **Random Forest + SMOTE** model over the *Baseline* and *XGBoost* models, we analyze its final confusion matrix. This is where the operational relevance of the model is determined:

- **Drastic Reduction of False Negatives (Maximum Safety):** The most critical result is the number of **False Negatives**, which drops to only **6**.
  - *Comparison:* The Baseline model missed 22 anomalies. The final model misses only 6 (out of 1105 abnormal cases). This represents a major improvement in detection capability, making the system significantly safer.
- **Maintenance of an Acceptable False Positive Rate:** The model generates **42 False Positives**.
  - Although this figure is slightly higher than the Baseline (which had 28), this increase is **negligible and perfectly acceptable** considering the safety gain. Accepting 14 additional false alarms to detect 16 more critical anomalies is a highly profitable trade-off.



Comparative Conclusion

Unlike **XGBoost** (which showed higher variance) and the **Baseline** (which missed too many anomalies), the **RF + SMOTE** model achieves the ideal balance. It offers **near-total coverage of anomalies (Recall** $> 99\%$**)** while maintaining high operational precision, making it the undisputed best candidate for deployment.

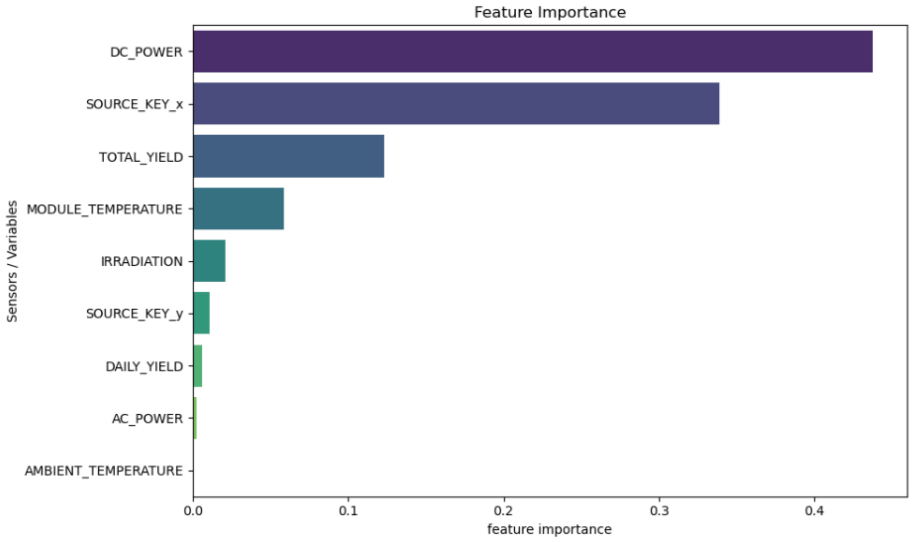
6 Model Interpretability: Feature Importance Analysis

To validate the physical consistency of the model and extract domain knowledge, we analyzed the relative Feature Importance used by the Random Forest to make its decisions.

The *Feature Importance* chart highlights a clear hierarchy:

1. **Dominance of DC Power (`DC_POWER`):** With a score exceeding **0.40**, `DC_POWER` (direct current generated by the panels) is by far the most discriminating variable.
    - *Interpretation:* This is physically logical. An anomaly primarily manifests as a drop or abnormal fluctuation in raw energy production. The model correctly identified this as the primary indicator of the system's health.
2. **Role of Identifiers (`SOURCE_KEY`):** The variable `SOURCE_KEY_x` ranks second.
    - *Interpretation:* This suggests that anomalies are not uniformly distributed. Specific inverters or sensors are likely more prone to failures than others, making the hardware identity itself a strong predictive factor.

3. **Variable Redundancy (`AC_POWER`):** It is noteworthy that `AC_POWER` has near-zero importance, despite being physically converted from `DC_POWER`.
    - *Interpretation:* The model detected the strong multicollinearity between DC and AC power. It selected `DC_POWER` as the primary signal and treated `AC_POWER` as redundant information.
4. **Environmental Context:** Module temperature (`MODULE_TEMPERATURE`) and irradiation (`IRRADIATION`) play a secondary but significant role. They likely allow the model to establish the "expected output" context (e.g., distinguishing a night-time zero power from a day-time failure).



7 Conclusion and Future Perspectives

The primary objective of this project was to develop a robust machine learning solution capable of automating the detection of anomalies in solar power generation. By transitioning from a manual monitoring approach to a data-driven predictive model, we addressed the critical challenge of maintaining optimal efficiency in photovoltaic plants.

7.1 Summary of Achievements

Through a rigorous methodology involving **Principal Component Analysis (PCA)** for dimensionality reduction and **SMOTE** for handling severe class imbalance, we successfully demonstrated that machine learning can reliably identify underperforming inverters.

The comparative analysis of multiple algorithms identified the **Random Forest classifier coupled with SMOTE** as the optimal solution. This model achieved exceptional performance metrics:

- A **Recall of 99.46%**, ensuring that virtually all system faults are detected.
- A drastic reduction in critical errors, limiting **False Negatives to only 6 cases** out of nearly 1,100 anomalies.
- A stable and interpretable decision process, largely driven by **DC Power** fluctuations and specific inverter identifiers.

7.2 Business Impact

For the plant operator, the deployment of this model translates into tangible operational benefits:

1. **Proactive Maintenance:** The ability to detect faults immediately allows for targeted repairs before minor issues escalate into major hardware failures.
2. **Yield Maximization:** By minimizing the downtime of "Abnormal" inverters, the total energy output of the plant is optimized.
3. **Resource Optimization:** Maintenance teams can focus their efforts on specific inverters flagged by the model (identified via `SOURCE_KEY`), rather than conducting random checks.

7.3 Future Work

To further enhance this system for a production environment, several avenues could be explored:

- **Deep Learning (Time-Series):** Implementing Recurrent Neural Networks (RNNs) or LSTMs (Long Short-Term Memory) could capture temporal dependencies better than the current snapshot-based approach, potentially predicting faults *before* they occur.
- **Real-Time Deployment:** Integrating the model into a streaming architecture (e.g., via an API or Edge Computing device) to provide live alerts to dashboard operators.
- **Weather Integration:** Incorporating external weather forecast data could help differentiate between temporary weather-related power drops and genuine hardware anomalies with even greater precision.