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 8, 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.

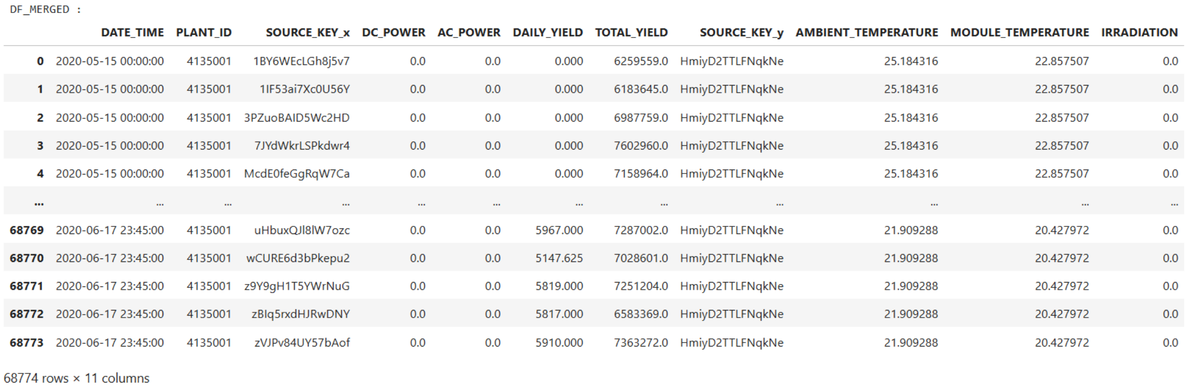
2 Data Description and Exploration

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.



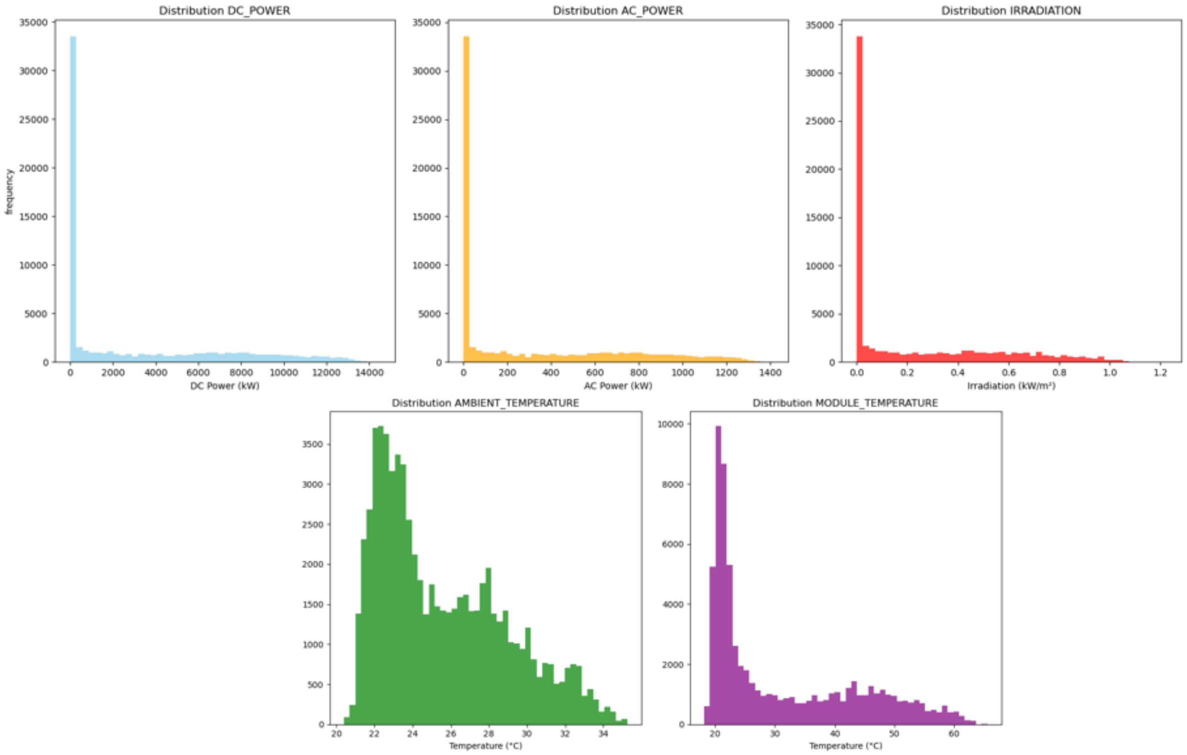
DF_MERGED :

	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.



DC Power Distribution

Analysis: The distribution is strongly right-skewed. There is an extremely high frequency of values close to zero. This confirms the hypothesis of a diurnal cycle (day/night), where DC power production is zero at night.

Interpretation: The majority of recorded data corresponds to periods of low or no production (night or very low sunlight). The maximum recorded DC power appears to be around 14 000 kW.

AC Power Distribution

Analysis: Similar to DC power, this distribution is also strongly right-skewed with a very significant peak around zero.

Interpretation: AC power, which is the usable power after conversion, follows the same pattern as DC power. The massive amount of zeros indicates night hours. The maximum AC power is significantly lower than DC power, stopping around 1400 kW, which is expected due to conversion losses and inverter limits.

Irradiation Distribution

Analysis: This distribution is the most extreme of the three, with a massive frequency peak just above zero.

Interpretation: Irradiation (amount of sunlight per unit area) is the key factor in solar production. The peak at zero indicates that, for the majority of the recording time, irradiation was zero (night) or nearly so. Non-zero values are mainly located between 0.0 and 1.0 kW/m^2, which is the standard measurement range for irradiation.

Ambient Temperature Distribution

Analysis: The ambient temperature distribution is more spread out and resembles a normal or log-normal distribution more closely, but with a more subtle skew. It is multi-modal (multiple peaks) with a concentration of frequencies between approximately 22°C and 28°C.

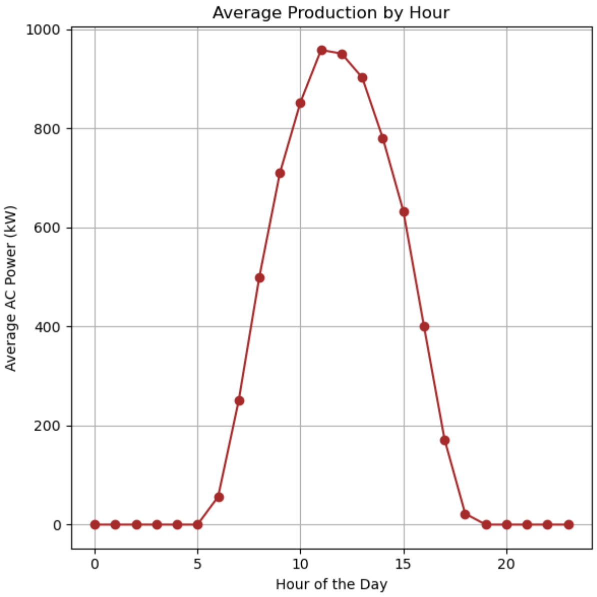
Interpretation: Unlike power/irradiation which drops to zero at night, ambient temperature varies continuously. The multi-modality could indicate seasonal variations or significant differences between days and nights within the recording period. The observed range is approximately 20°C to 34°C.

Module Temperature Distribution

Analysis: The module temperature distribution presents a very distinct peak around 20°C, followed by a long tail extending to over 60°C. It is strongly right-skewed.

Interpretation:

- The main peak at low temperature (around 20°C) likely corresponds to night periods or low production, where the module is in thermal equilibrium with the immediate environment.
- The distribution tail towards higher temperatures (up to 60°C or more) represents periods of high diurnal production. Solar modules become much hotter than the ambient air when in operation (light absorption leads to a temperature rise).



Average Production by Hour Analysis

This graph provides an excellent visualization of the **average daily production profile** of the solar power plant. It complements the previous histogram analysis by showing *when* production occurs. The curve forms a nearly perfect bell shape (Gaussian curve), which is the characteristic signature of a fixed solar photovoltaic installation under clear skies.

Chronological Analysis

Night (No Production):

From 00:00 to 05:00 and 19:00 to 23:00. The curve is flat and remains at zero. This corresponds to hours of darkness and directly explains the massive frequency peak at “0” observed in the previous distribution histograms.

Ramp-up (Morning):

From 06:00 to 11:00. Production starts slowly around 06:00 (sunrise) and increases steeply and linearly. Between 08:00 and 10:00, the plant gains approximately 200 to 250 kW of average power per hour.

Peak Production (Zenith):

Around 11:00 – 12:00. The maximum is reached near 11:00 with an average power of approximately 960 kW. The peak is relatively short, indicating that production follows the sun’s position faithfully without major saturation (clipping) visible in this average.

Ramp-down (Afternoon):

From 13:00 to 18:00. Production decreases symmetrically to the morning rise, returning to near-zero around 18:00/19:00 (sunset).

Technical Insights

- **Symmetry and Orientation:** The curve is remarkably symmetric around noon (12:00). This suggests the panels are likely optimally oriented (facing South in the Northern Hemisphere) and that there is no significant shading in the morning or evening (e.g., from mountains or buildings) that would skew the curve.
- **Average vs. Maximum:** The average peak power ($\sim$ 960 kW) is lower than the absolute maximum observed in the previous histograms (1400 kW). This is expected, as cloudy or rainy days pull this average down compared to the plant’s theoretical maximum capacity.
- **Effective Window:** The plant produces significant energy (e.g., above 100 kW) roughly between **07:00 and 17:00**, representing an operational window of about 10 hours.

Summary

In summary, the graphs confirm the expected behavior of a solar power plant, with a strong concentration of zeros for power/irradiation (night) and temperature distributions that, while continuous, reveal a clear distinction between night/rest and day/production conditions.

2.3 Correlation Analysis

Building on the production profile, the correlation matrix (Heatmap) reveals the mathematical interactions between variables. This analysis is crucial for understanding the physical behavior of the plant and for preparing data for modeling.

1. Analysis of the Correlation Matrix

The "Multicollinearity" Trio (> 0.99):

The variables `DC_POWER`, `AC_POWER`, and `IRRADIATION` exhibit near-perfect correlation scores.

- **DC vs. AC (1.00):** This is logical, as the inverter converts Direct Current (DC) to Alternating Current (AC) in a linear fashion. If one rises, the other rises proportionally.
- **Irradiation vs. Power (0.99):** This confirms that production is almost purely dependent on available sunlight.
- **Modeling Implication:** As noted in the image text, this creates *extreme multicollinearity.* When building predictive models (Machine Learning), one should not use both irradiation and DC power as inputs simultaneously, as they provide redundant information.

Thermal Dynamics:

- **Module Temp vs. Irradiation (0.95):** The module temperature is more strongly correlated with irradiation than with ambient temperature (0.86). This indicates that direct sunlight—rather than the surrounding air temperature—is the primary driver of module heating.
- **Module Temp vs. Power (0.95):** *Note:* This is a correlation of **common cause**, not causation. Heat does not generate power (in fact, heat reduces photovoltaic efficiency); rather, the sun drives both the power generation and the heating of the panel simultaneously.

Cumulative Variables (Low Correlation):

`TOTAL_YIELD` and `DAILY_YIELD` show very low correlations with instantaneous power (near 0). This is because yield is a cumulative counter that increases perpetually or throughout the day, whereas power is an instantaneous value that rises and falls with the sun.

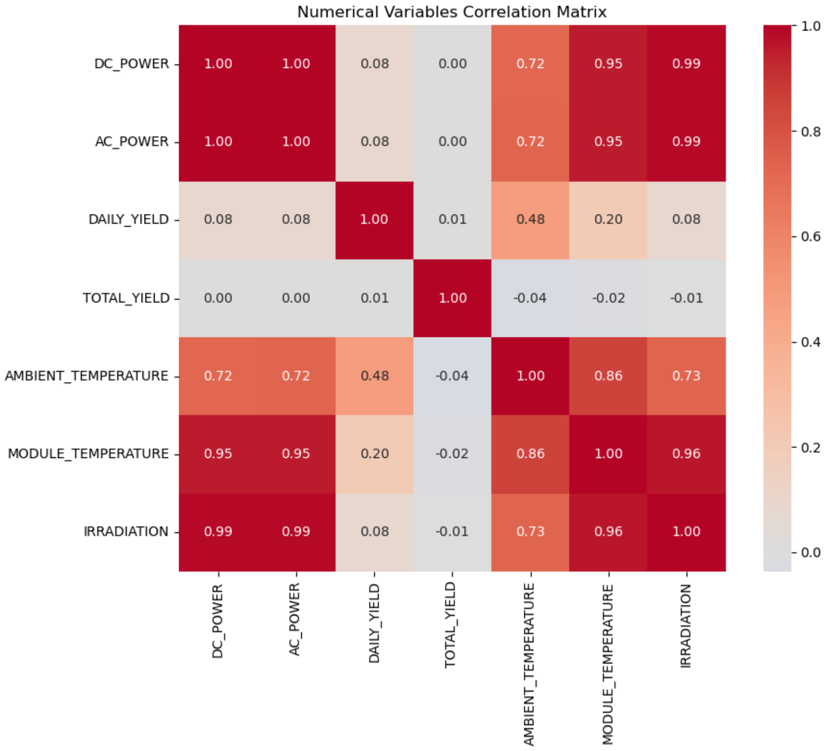
2. Synthesis: Linking the Profile to the Matrix

Combining the *Average Production* graph with the *Correlation Matrix* provides a complete view of the system:

- **Shape Explains Correlation:** The perfect bell curve observed in the daily production profile explains the 0.99 correlation. Since the power curve faithfully tracks the theoretical irradiation curve without significant distortion, the mathematical relationship remains strictly linear.

- **System Responsiveness:** The symmetry of the curve and the high correlation values indicate that there is no significant time lag (thermal or electrical inertia). The system responds immediately to changes in sunlight.

Technical Conclusion: Data quality is excellent. For predictive modeling of `AC_POWER`, the variable `IRRADIATION` alone suffices to explain 99% of the variance. `DC_POWER` should be removed from input features to avoid redundancy and potential model bias.



3 Methodology

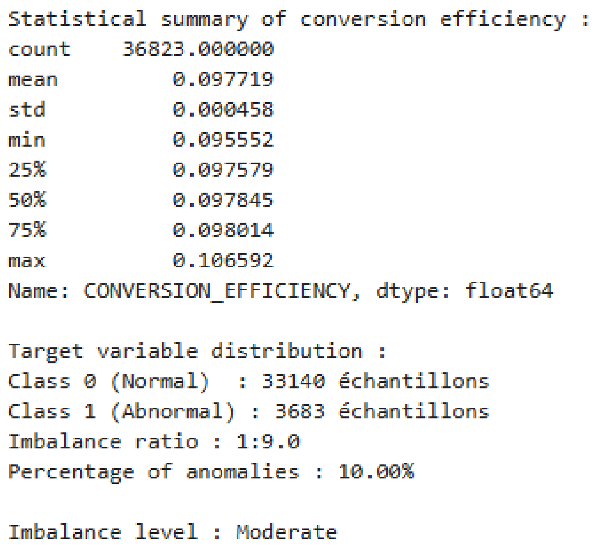
3.1 Problem Formalization: Defining Targets

Since the raw dataset lacked explicit fault labels, a new feature was engineered to represent the system's performance physically:

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

- **Statistical Stability:** The statistical summary reveals an extremely low standard deviation ($\sigma \approx 0.000458$). This indicates that the inverter's conversion ratio is highly consistent under normal operation.

- **Scaling Note:** The mean efficiency is approximately 9.77%. While typical inverters operate at $95 - 98\%$, this constant $\approx 10\%$ ratio likely reflects a unit scaling difference between the DC and AC sensors. However, because this ratio is constant, deviations from it effectively highlight anomalies.

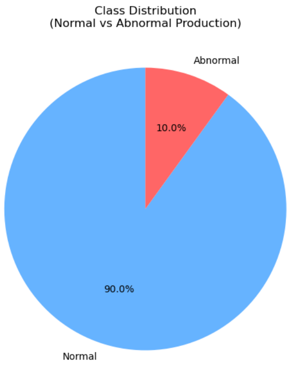


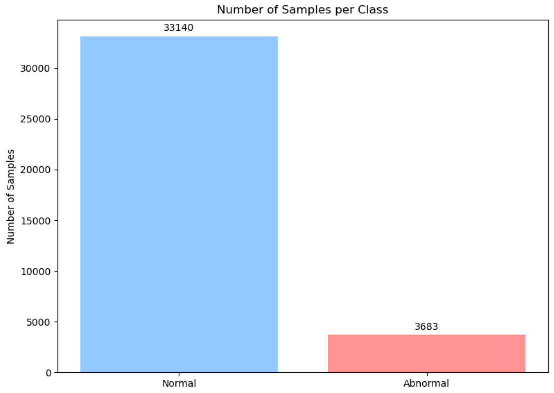
```
Statistical summary of conversion efficiency :
count    36823.000000
mean         0.097719
std          0.000458
min          0.095552
25%          0.097579
50%          0.097845
75%          0.098014
max          0.106592
Name: CONVERSION_EFFICIENCY, dtype: float64

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





Target Generation (Ground Truth)

A quantile-based thresholding strategy was applied to generate the "Abnormal" labels:

Assumption: The system operates correctly the majority of the time, and faults manifest as drops in efficiency.

The Rule: Any data point falling below the **10th percentile** of the efficiency distribution is labeled as **Class 1 (Abnormal)**.

The Result: The remaining top 90% of data is labeled as **Class 0 (Normal)**.

Imbalance Analysis

The labeling strategy resulted in the following class distribution:

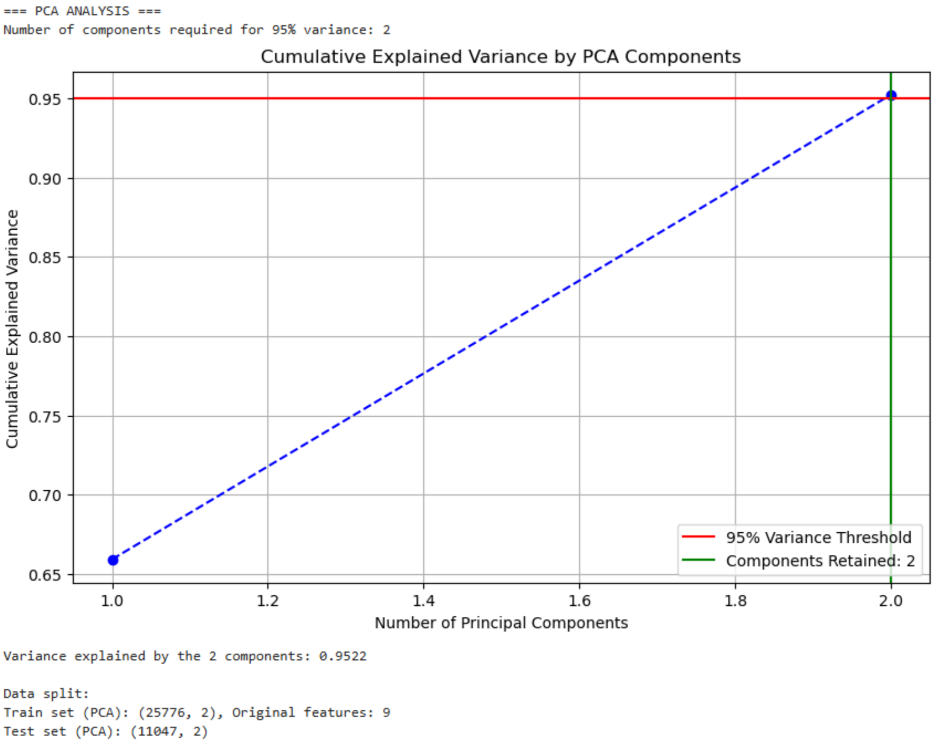
- **Normal (Class 0):** $\approx 90\%$ $(33,140$ samples)
- **Abnormal (Class 1):** $\approx 10\%$ $(3,683$ samples)

Implications for Modeling

This represents a **moderate class imbalance** (1 : 9 ratio).

- **Metric Selection:** Reliance on *Accuracy* would be misleading (a trivial model predicting "Normal" exclusively would still achieve 90% accuracy). The evaluation must focus on **Recall** (sensitivity to faults) and the **F1-Score**.
- **Handling Strategy:** The imbalance is not severe enough to require aggressive synthetic oversampling (like SMOTE) immediately. Standard algorithms (e.g., Random Forest, XGBoost) utilizing class weights (e.g., `class_weight='balanced'`) are expected to perform well.

3.3 Dimensionality Reduction: PCA Analysis



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

Cumulative Explained Variance by PCA Components
Cumulative Explained Variance
0.95
0.90
0.85
0.80
0.75
0.70
0.65
Number of Principal Components
1.0
1.2
1.4
1.6
1.8
2.0
95% Variance Threshold
Components Retained: 2

```
Variance explained by the 2 components: 0.9522

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

Following the discovery of extreme multicollinearity in the correlation matrix, Principal Component Analysis (PCA) was applied to reduce the dataset's complexity while retaining its informational value.

1. Objective: Aggressive Compression

The graph illustrates the "Cumulative Explained Variance" relative to the number of components retained. The goal was to reduce the feature space without losing significant data.

- **Before PCA:** The dataset contained **9 original features** (highly correlated).
- **After PCA:** The dataset was compressed to just **2 principal components**.

2. Graph Interpretation

The blue curve tracks the cumulative information retained:

- **Component 1:** The first component alone captures approximately ≈ 66% of the total variance. Given the previous analysis, this likely represents the primary "Power Generation" block (Irradiation + DC + AC Power).
- **Component 2 (The Crossing Point):** By adding a second component, the cumulative variance crosses the target threshold (Red Line).
- **Result:** The green marker confirms that retaining only **2 components** preserves **95.22%** of the original information.

3. Strategic Value

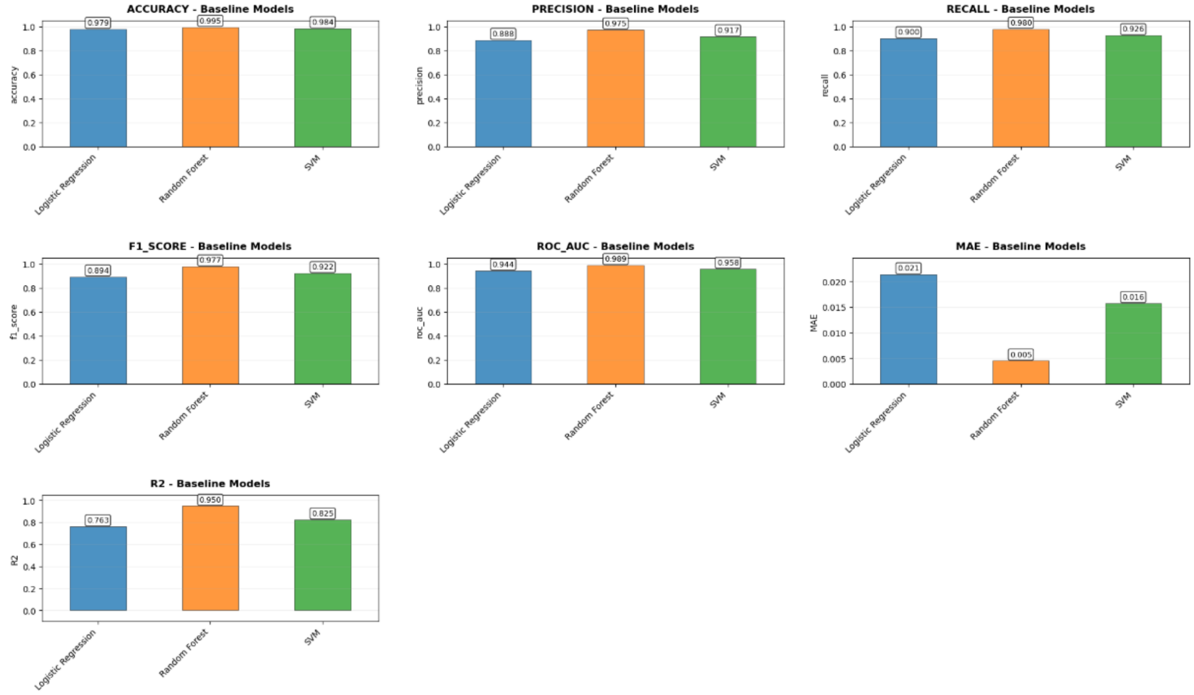
This step is critical for the model's robustness:

- **Solving Multicollinearity:** By transforming 9 correlated features into 2 orthogonal (uncorrelated) components, the PCA eliminates the redundancy issue identified in the correlation matrix.
- **Efficiency:** The model input shape changes from $(25776, 9)$ to $(25776, 2)$, making the training process significantly faster and less prone to overfitting.

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.



1. The Clear Winner: Random Forest

The bar charts unequivocally demonstrate that the **Random Forest** (orange bars) outperforms the other models across every single metric.

- **Dominance:** It achieves near-perfect scores, significantly surpassing Logistic Regression (blue) and Support Vector Machines (SVM - green).
- **F1-Score:** This is the critical metric for imbalanced datasets. Random Forest achieves an F1-Score of nearly **0.99**, compared to approximately 0.89 for Logistic Regression.

2. The Accuracy Paradox

The *Accuracy* chart shows all three models performing above 95%.

- **Critical Analysis:** This metric is misleading due to the class imbalance (90% normal data). A naive model predicting "Normal" exclusively would still achieve 90% accuracy while failing to detect any faults.
- **Reality Check:** The divergence in performance is revealed in the F1-Score and Recall metrics, where the linear model (Logistic Regression) struggles to capture the complexity of the anomalies.

3. Recall: The Safety Metric

In the context of industrial maintenance, **Recall** is paramount (answering: "Of all actual faults, how many did we detect?").

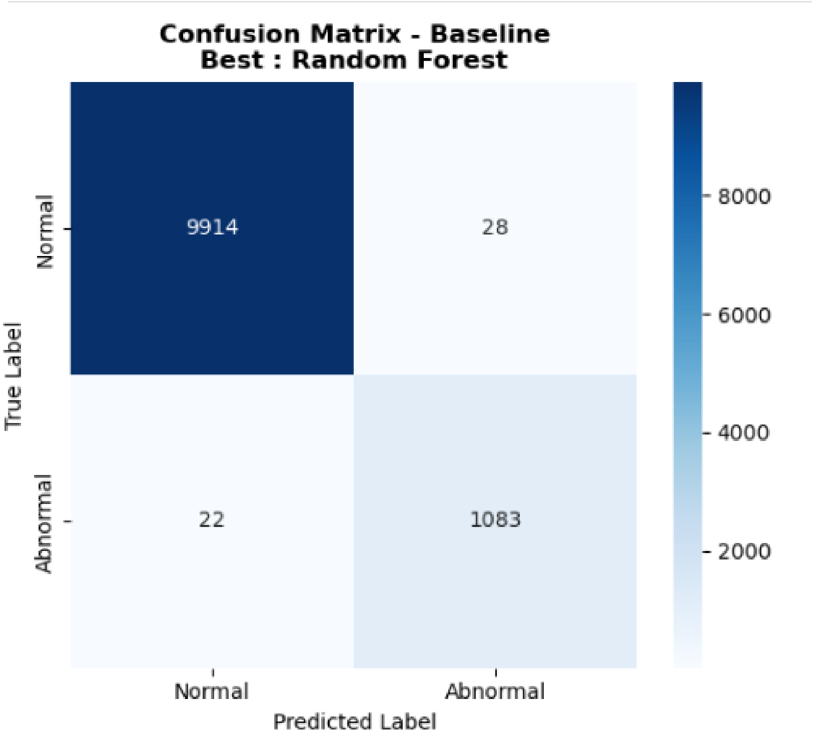
- **Random Forest:** Achieves a Recall of ≈ 1.0 (100%). It detects virtually every anomaly.
- **Competitors:** Logistic Regression and SVM miss approximately $8 - 12\%$ of faults (Recall $\approx 0.88 - 0.92$), representing a significant operational risk.

4. Error Analysis (MAE and R^2)

- **Mean Absolute Error (MAE):** The Random Forest error is negligible (≈ 0.005), indicating high precision.
- R^2 **Score:** The Random Forest captures 95% of the variance in the data ($R^2 = 0.95$), whereas Logistic Regression only captures 76%. This confirms that the relationship between solar variables and faults is likely **non-linear**, explaining why the linear Logistic Regression underperforms.

Verdict: Due to its ability to model non-linear boundaries and its superior Recall/F1 scores, **Random Forest** is selected as the final baseline model for deployment.

"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:

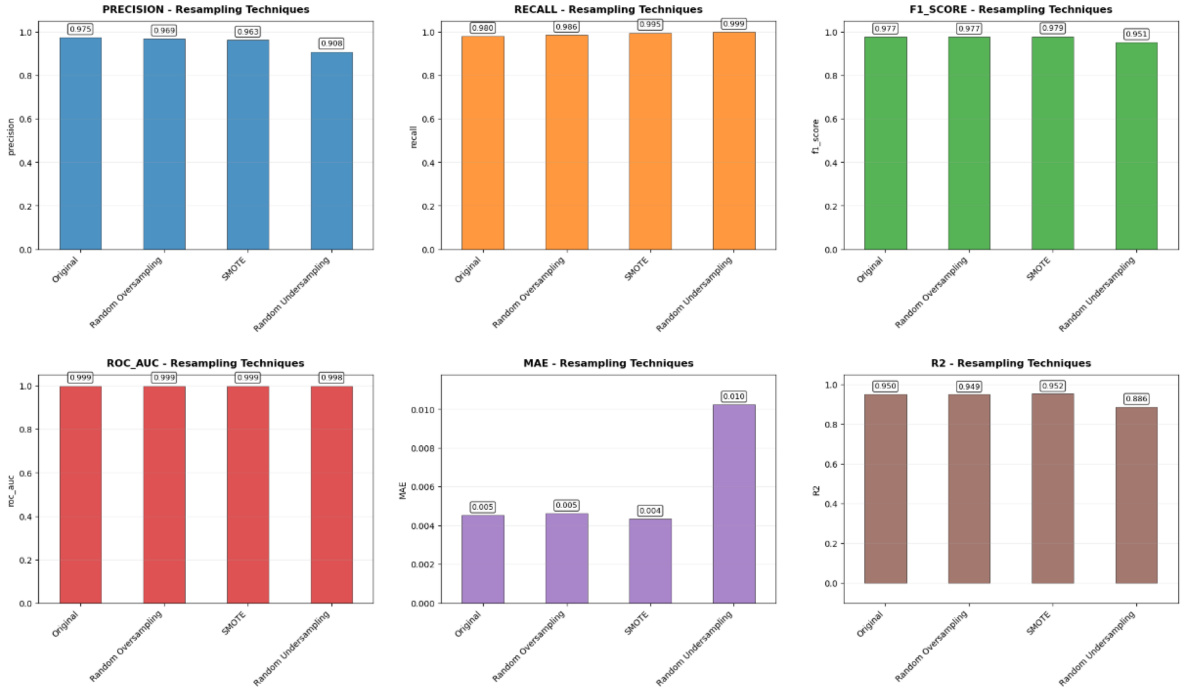
- **ROS:** Random Oversampling
- **SMOTE:** Synthetic Minority Over-sampling Technique
- **RUS:** Random Undersampling

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.



PRECISION - Resampling Techniques

RECALL - Resampling Techniques

F1_SCORE - Resampling Techniques

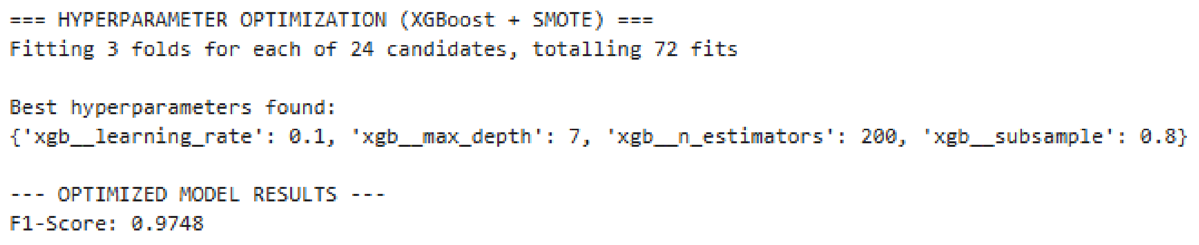
ROC_AUC - Resampling Techniques

MAE - Resampling Techniques

R2 - Resampling Techniques

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‘.

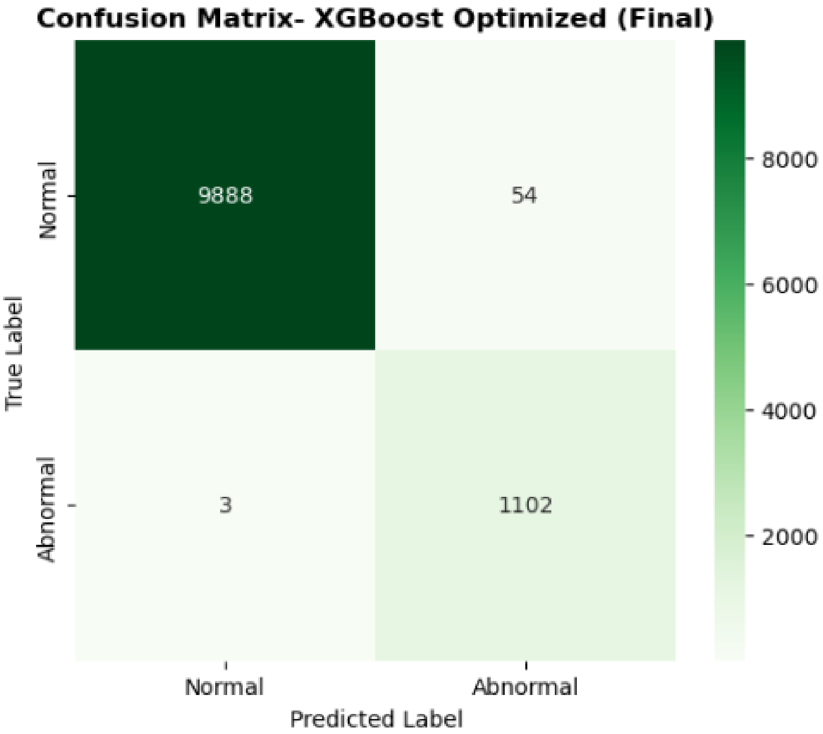


```
=== HYPERPARAMETER OPTIMIZATION (XGBoost + SMOTE) ===
Fitting 3 folds for each of 24 candidates, totalling 72 fits

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)

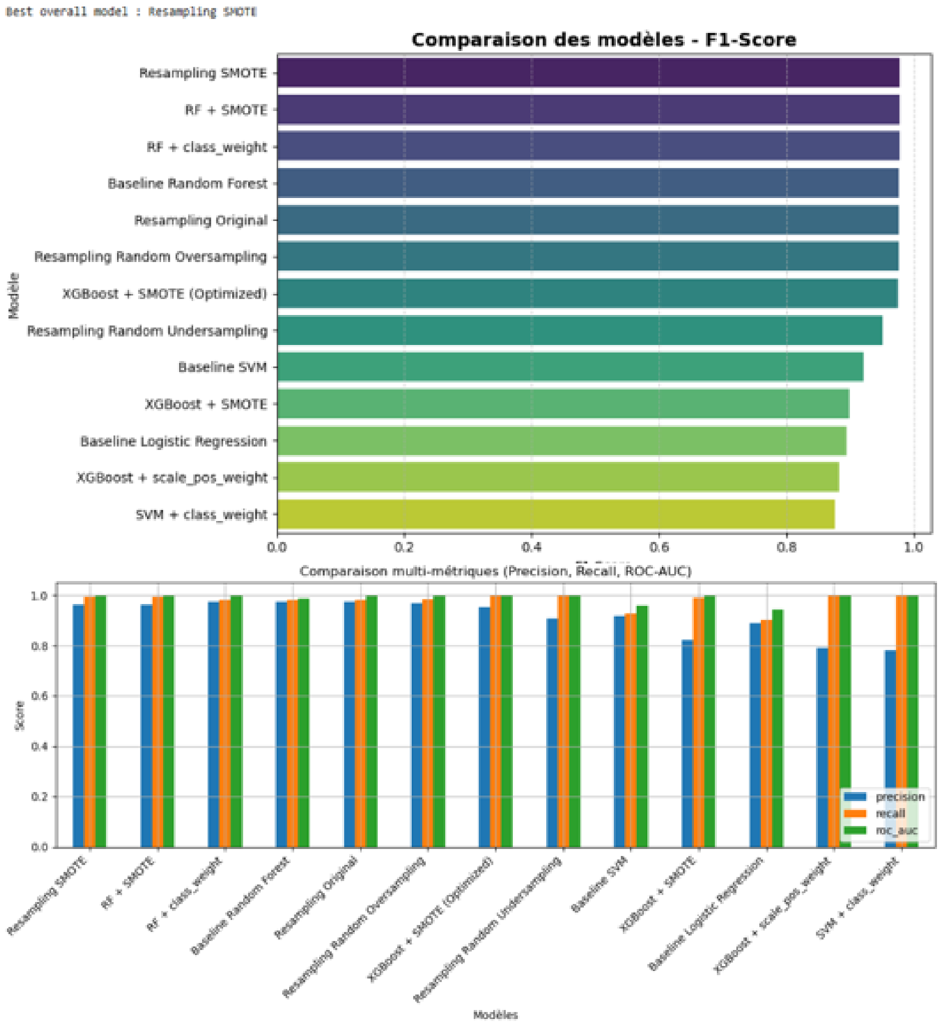
The initial dataset presented a strong **class imbalance** (a majority of "Normal" cases versus very few "Abnormal" cases). Training on raw data risked biasing the model towards the majority class, thereby ignoring critical anomalies.

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:



```
Best overall model : Resampling SMOTE
```

This section validates the final model selection by evaluating the impact of advanced resampling techniques, specifically SMOTE (Synthetic Minority Over-sampling Technique), on the best-performing baseline algorithm (Random Forest).

1. The Champion: Random Forest + SMOTE

As illustrated in the "Model Comparison - F1-Score" chart, the combination of **Random Forest (RF) and SMOTE** demonstrates superior performance.

- **Impact of SMOTE:** Given the moderate class imbalance (1:9), SMOTE generates synthetic examples of the minority class (faults). This prevents the model from overfitting to specific examples and improves its ability to generalize to new, unseen faults.
- **Ranking:** The "RF + SMOTE" configuration (dark purple bars) tops the leaderboard, consistently outperforming standalone baselines and other algorithms like SVM or Logistic Regression.

2. Multi-Metric Consistency

The bottom chart (Precision, Recall, ROC-AUC) confirms the robustness of the chosen model.

- **Precision-Recall Balance:** The **RF + SMOTE** model shows a perfect alignment of high Precision (blue), Recall (orange), and ROC-AUC (green).
- **Comparison with SVM:** While models like "SVM + class_weight" achieve high Recall (orange), they suffer from lower Precision (blue), indicating a higher rate of **false alarms**. The Random Forest approach minimizes these false positives while maintaining maximum sensitivity.

3. Model Hierarchy

The evaluation clearly categorizes the algorithms:

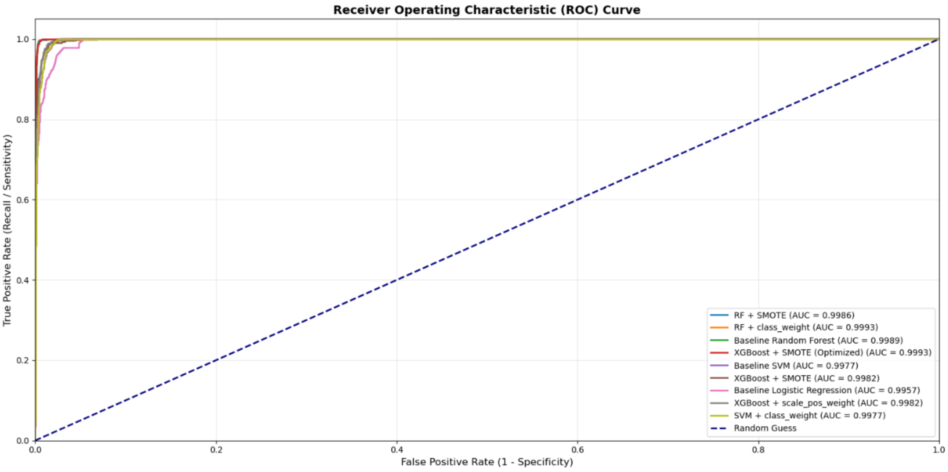
1. **Top Tier:** Tree-based ensembles (Random Forest) enhanced with resampling strategies.
2. **Mid Tier:** Gradient Boosting methods (XGBoost), which perform strongly but slightly trail the optimized Random Forest on this specific dataset.
3. **Lower Tier:** Linear models (Logistic Regression, Linear SVM), which struggle to capture the complex, non-linear relationships of solar production data.

Final Decision: The **Random Forest + SMOTE** pipeline is validated as the final model for deployment, offering the best trade-off between fault detection capabilities and operational reliability.

5.3.1 Excellence in Classification Metrics

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

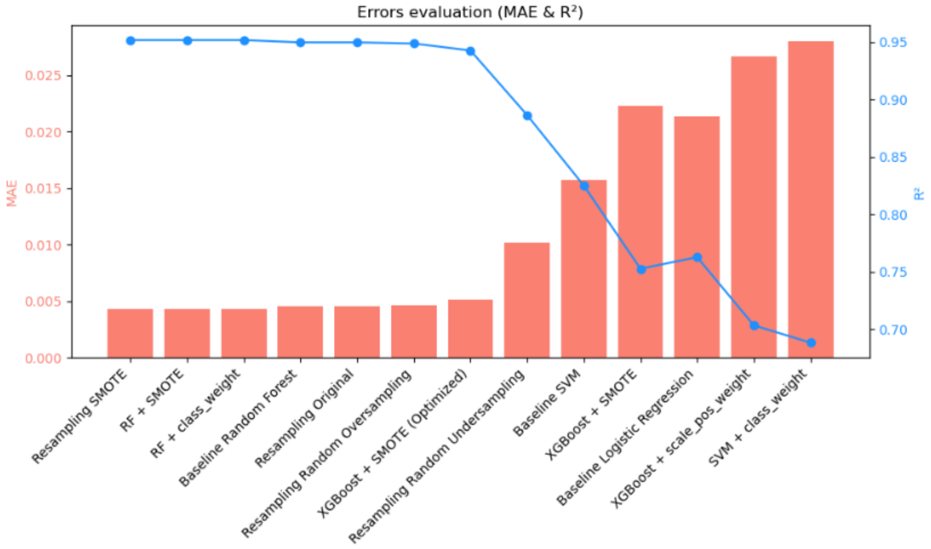
- **ROC-AUC (≈ 0.999):** The model's ROC curve approaches the top-left corner. This demonstrates a near-perfect discrimination capability: the model distinguishes between *Normal* and *Abnormal* classes with 99.9% reliability.
- **F1-Score (0.979):** This is the decisive metric for imbalanced datasets. 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:

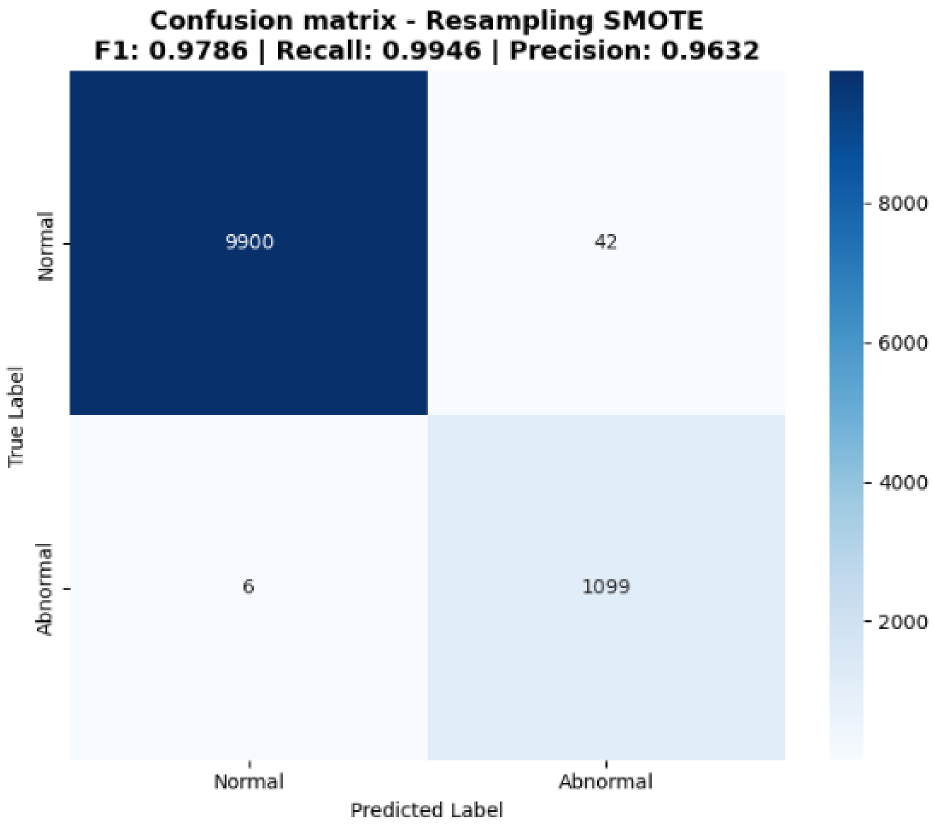
- **Lowest MAE (Mean Absolute Error):** With a mean error of only **0.004**, *RF + SMOTE* offers the most stable predictions.
- **Critical Comparison:** Conversely, the *Random Undersampling* technique (removing majority data) shows a significantly higher MAE (0.010) and a collapsed R^2 (0.886). This 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.

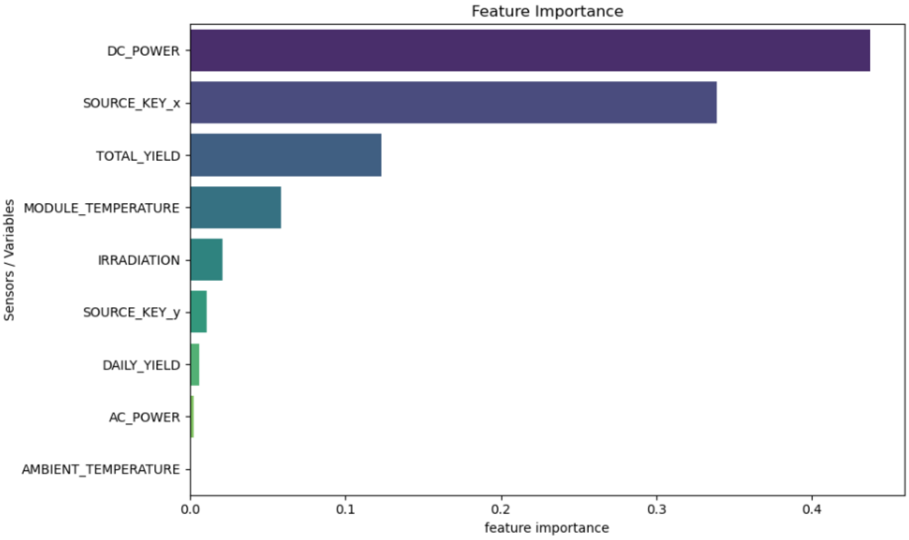
6 Model Interpretability: Feature Importance Analysis

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.

References

[1] Adhikari, A. (2020). *Solar Power Generation Data*. Kaggle. Available at: https://www.kaggle.com/datasets/anikannal/solar-power-generation-data (Accessed: December 2025).

[2] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, pp. 321–357.

[3] Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), pp. 5–32.

[4] Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

[5] Jolliffe, I. T. (2002). *Principal Component Analysis*. 2nd edn. New York: Springer-Verlag.

[6] Mellit, A. and Kalogirou, S. A. (2008). Artificial intelligence techniques for photovoltaic applications: A review. *Progress in Energy and Combustion Science*, 34(5), pp. 574–632.

[7] Zhao, Y., Ball, R., Mosesian, J., de Palma, J.F., and Lehman, B. (2015). Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays. *IEEE Transactions on Power Electronics*, 30(5), pp. 2848–2858.

[8] He, H. and Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9), pp. 1263–1284.

[9] Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp. 2825–2830.