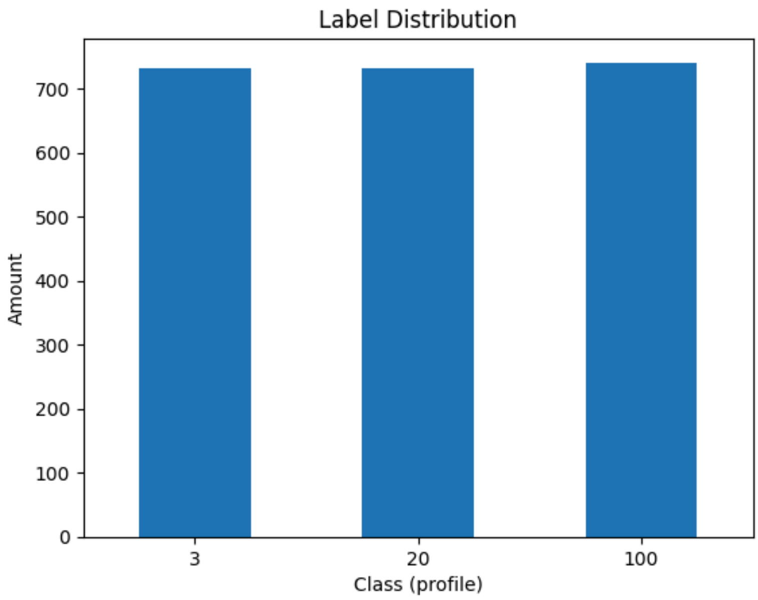
1. Condition Monitoring of Hydraulic Systems
Analysis using public datasets from:
https://www.kaggle.com/datasets/jjacostupa/condition-monitoring-of-hydraulic-systems

Dataset preview:

	CE_0	CE_1	CE_2	CE_3	CE_4	CE_5	CE_6	CE_7	CE_8	CE_9	...	VS1_56	VS1_57	VS1_58	VS1_59	profile_0
0	47.202	47.273	47.250	47.332	47.213	47.372	47.273	47.438	46.691	46.599	...	0.544	0.545	0.535	0.543	3
1	29.208	28.822	28.805	28.922	28.591	28.643	28.216	27.812	27.514	27.481	...	0.540	0.533	0.531	0.534	3
2	23.554	23.521	23.527	23.008	23.042	23.052	22.658	22.952	22.908	22.359	...	0.545	0.544	0.530	0.534	3
3	21.540	21.419	21.565	20.857	21.052	21.039	20.926	20.912	20.989	20.882	...	0.544	0.543	0.543	0.542	3
4	20.460	20.298	20.350	19.867	19.997	19.972	19.924	19.813	19.691	19.634	...	0.549	0.542	0.533	0.537	3

Each .txt file contains measurement data from different sensors. For example:
CE.txt: Cooler efficiency
CP.txt: Cooler power
EPS1.txt: Efficiency factor of pump 1
FS1.txt, FS2.txt: Flow sensors
PS1.txt to PS6.txt: Pressure sensors
SE.txt: System efficiency
TS1.txt to TS4.txt: Temperature sensors
VS1.txt: Valve sensor
description.txt: A brief description of the data and its use.
documentation.txt: A detailed description (metadata), including units, sensor position, and data meaning.
profile.txt: Labels/targets related to system conditions, often used to classify system conditions.

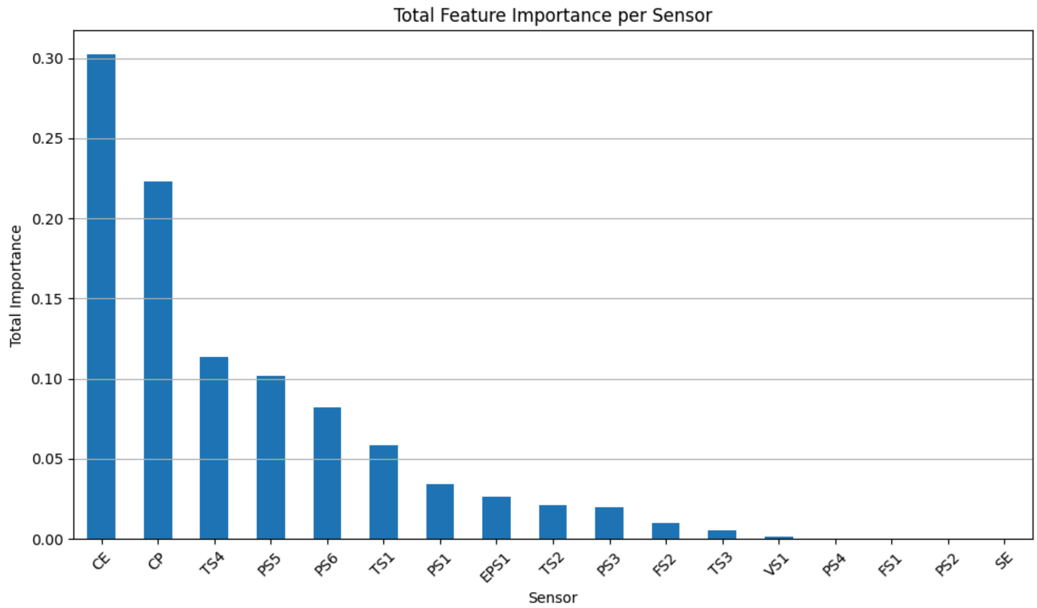
Classification Report (Random Forest):



```
=== Classification Report ===
              precision    recall  f1-score   support

           3       1.00      1.00      1.00       147
          20       1.00      1.00      1.00       146
         100       1.00      1.00      1.00       148

    accuracy                           1.00       441
   macro avg       1.00      1.00      1.00       441
weighted avg       1.00      1.00      1.00       441
```

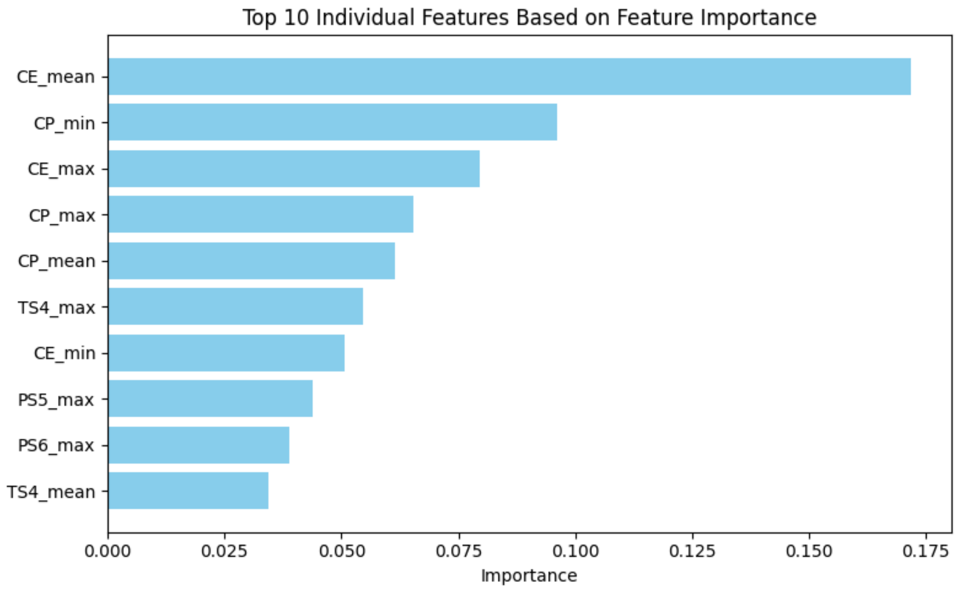
The dataset is very balanced and well-prepared. The Random Forest model can distinguish between the 3, 20, and 100 conditions very well. But the model is too perfect—additional validation is needed to ensure it doesn't overfit.

Visualization of Feature Importance per Sensor:



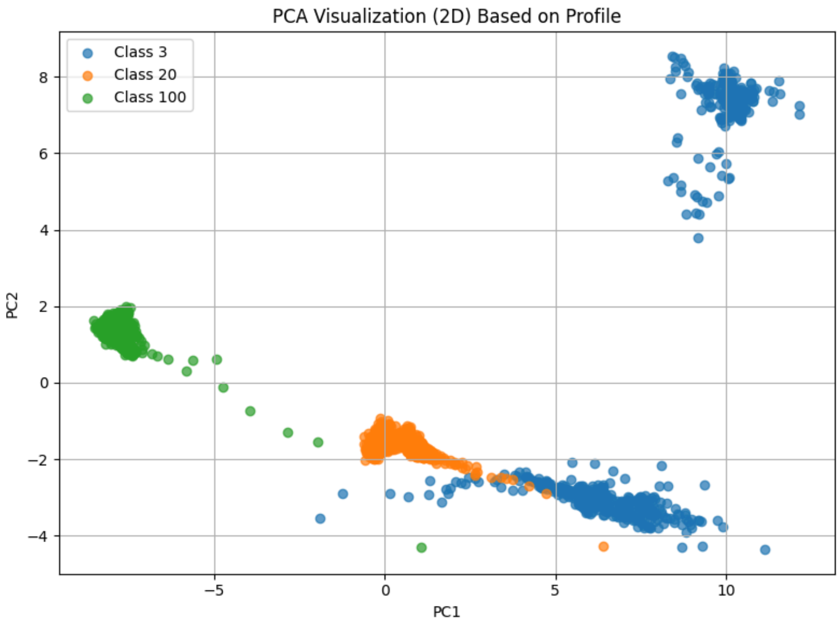
A bar chart showing which sensors have the greatest influence on the condition classification (profile). For example, if PS6 or EPS1 is dominant, it means that pump pressure or efficiency significantly impacts system conditions.

Top 10 Individual Features Based on Importance:



A list of 10 features, such as PS6_max, EPS1_mean, etc. We can see which specific statistical features and sensors are most dominant in label prediction. Helps identify the most informative sensors and statistics types.

2D PCA Visualization of Sensor Summary Data:



If the classes (3, 20, 100) appear separate or form clusters, this means the feature effectively differentiates system conditions. If they overlap, this could mean: The feature is not representative enough and A non-linear method is needed.

Compare the accuracy of the model with and without the pressure feature to determine: Is the pressure sensor critical? How much does accuracy decrease if pressure is removed?

```
Model accuracy with all features        : 1.0000
Accuracy of the model without pressure features (PS*) : 1.0000
```

If accuracy drops significantly without the pressure feature, it means the pressure sensor is critical for classification. If accuracy remains high, the system has other features (e.g., EPS1, SE, TS*) that are strong enough to distinguish classes.

How sensitive the model is to each pressure sensor (PS1–PS6). Comparison of accuracy stability through cross-validation. Relative performance of tree models (Random Forest) vs. linear models (SVM, Logistic Regression):

	Sensor_Dropped	Model	Accuracy_Mean	Accuracy_Std
2	None	Logistic Regression	0.998186	0.001697
0	None	Random Forest	0.998639	0.001111
1	None	SVM	0.998639	0.001111
5	PS1	Logistic Regression	0.998186	0.001697
3	PS1	Random Forest	0.998639	0.001111
4	PS1	SVM	0.998639	0.001111
8	PS2	Logistic Regression	0.998186	0.001697
6	PS2	Random Forest	0.998639	0.001111
7	PS2	SVM	0.998639	0.001111
11	PS3	Logistic Regression	0.998186	0.001697
9	PS3	Random Forest	0.998639	0.001111
10	PS3	SVM	0.998639	0.001111
14	PS4	Logistic Regression	0.998186	0.001697
12	PS4	Random Forest	0.998639	0.001111
13	PS4	SVM	0.998639	0.001111
17	PS5	Logistic Regression	0.998186	0.001697
15	PS5	Random Forest	0.998639	0.001111
16	PS5	SVM	0.998639	0.001111
20	PS6	Logistic Regression	0.998186	0.001697
18	PS6	Random Forest	0.998639	0.001111
19	PS6	SVM	0.998639	0.001111

See if accuracy drops drastically when a particular sensor is removed → that sensor is crucial. Check model stability using Accuracy_Std. Compare: are non-tree models (SVM, LR) more sensitive to the loss of pressure features than Random Forest?

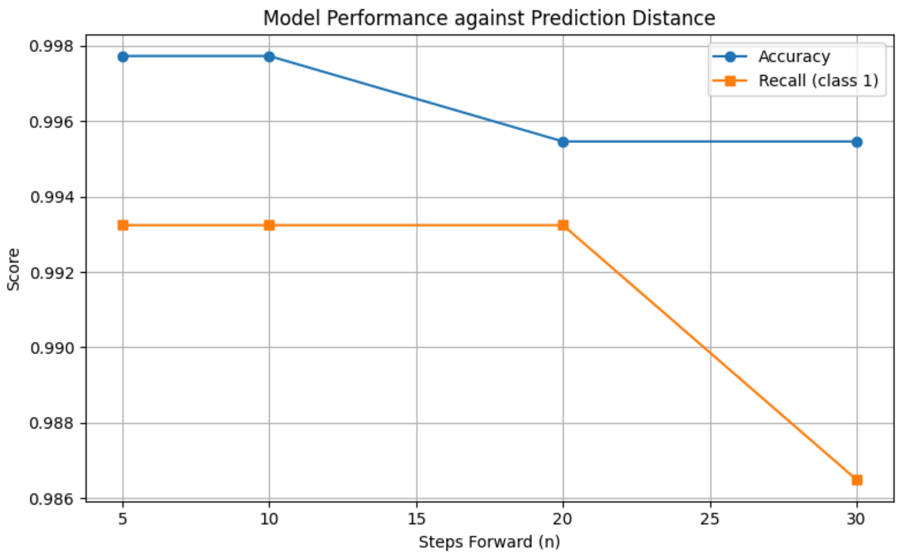
See the Most Important Features per Model:

```
Top 10 Most Important Features - Random Forest
    feature  importance
0   CE_mean    0.136198
3    CE_max    0.096303
2    CE_min    0.076144
4   CP_mean    0.072152
6    CP_min    0.070862
63  TS4_max    0.068526
7    CP_max    0.055263
43  PS6_max    0.051773
62  TS4_min    0.043609
39  PS5_max    0.040253

Top 10 Most Important Features - Logistic Regression
    feature      coef
49  TS1_std  0.512582
7    CP_max  0.486424
6    CP_min  0.461457
4   CP_mean  0.458941
53  TS2_std  0.382774
27  PS2_max  0.308741
2    CE_min  0.295878
0   CE_mean  0.294715
3    CE_max  0.294128
17  FS2_std  0.293218

Top 10 Most Important Features - SVM (Linear)
    feature      coef
7    CP_max  0.157290
4   CP_mean  0.148010
6    CP_min  0.146463
49  TS1_std  0.092772
2    CE_min  0.092640
3    CE_max  0.091198
0   CE_mean  0.090631
27  PS2_max  0.069767
23  PS1_max  0.065565
61  TS4_std  0.065230
```

System Failure Prediction (Fault Prediction): Predict future system conditions (whether they will deteriorate or remain normal) based on current sensor data to support preventive maintenance before failure occurs.

This dataset doesn't have an explicit time index, but we can assume the data is sequential (since it's time-series). The profile label represents the current state of the system:
3 = good
20 = declining
100 = bad / broken
We'll create a new label: "will break in the next n steps."

```
=== Multi-step Evaluation Summary ===
   step  accuracy  precision_1  recall_1      f1_1
0     5  0.997732     1.000000  0.993243  0.996610
1    10  0.997732     1.000000  0.993243  0.996610
2    20  0.995465     0.993243  0.993243  0.993243
3    30  0.995465     1.000000  0.986486  0.993197
```



High and stable accuracy up to 30 steps ahead, Current sensor data is highly informative for future predictions. Recall decreases at n=30, The model begins to fail to detect some cases that would otherwise fail if predictions were too far ahead.

Predictive Window Maintenance Simulation: real-time system monitoring to predict whether the system will fail in the near future, this approach is closest to the real application of predictive maintenance.

```
=== Real-Time Monitoring Simulation Evaluation ===
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        10
           1       0.98      1.00      0.99       431

    accuracy                           0.98       441
   macro avg       0.49      0.50      0.49       441
weighted avg       0.96      0.98      0.97       441
```

The model only predicts one class (label 1). All inputs are predicted as "will fail." As a result, recall is high for class 1, but class 0 (normal) is not detected at all.

Accuracy is deceptive (≈ 98%). This appears high, but this is because the test data is dominated by class 1 (431 vs. 10). There are no correct class 0 predictions. This is called class imbalance bias.

The model is too oversensitive/false alarms. All conditions are considered potential failures, which can lead to: Over-maintenance (high costs). Continuous false alarms. Suboptimal systems or unnecessary downtime.

Improvement Recommendations: Address class imbalance, Use more precise metrics and Analyze time-to-failure for more flexible decision making.