Midterm Project #2: End-to-End Predictive Modeling (10 Points Total)

This project requires you to execute a complete data-driven modeling workflow using energy, petroleum engineering, geophysics, or engineering dataset. The core emphasis is on rigorous machine learning practices, robust modeling, interpretability, and real-world deployment readiness. 3 notebooks, 1 train-test dataset, and 1 deployment dataset need to be submitted. All explanations, observations, and conclusions need to be added to notebooks.

Part 1: Problem Formulation & Data Acquisition (2 Points)

Task	Description	Points
1.1 Objective, Contribution & Source Citation *(In Notebook #1 with Train-Test Dataset)*	Clearly define the objective of your predictive modeling task and identify the existing work/source of inspiration. Articulate what problem the ML workflow is solving and why ML is necessary (i.e., why traditional analytical methods, physics-based models, or simple statistics are insufficient or less effective). The points for this task are heavily weighted on the demonstrated novelty, improvements, creativity, and complexity: - Novelty: Describe if you are attempting a new problem or adapting a known solution to a new domain/dataset. - Improvement/Modification: If referencing an existing solution, clearly state how much you were able to improve the existing solution (e.g., X% better performance, Y new features added, more robust model), or the significant modification you introduced.	1.5
1.2 Data Acquisition & Target Type *(In Notebook #1 with Train-Test Dataset)*	Identify and acquire a relevant energy or engineering dataset. Clearly document the dataset source and confirm your problem is a Regression, Multi-class Classification, or Multi-label Classification task (**binary classification not allowed**). Perform the train-test split at the very beginning of your process to ensure no future information mixing.	0.5

Part 2: Data Preprocessing & Feature Engineering (1.5 Points)

Task	Description	Points
2.1 Data Preprocessing *(In Notebook #1)*	Implement proper data preprocessing (e.g., handling missing values, encoding categorical data). Crucially, transform features that are skewed or highly non-Gaussian to improve model convergence and performance. Don't mix information between train/test.	0.5
2.2 Creative Feature Engineering *(In Notebook #1)*	Create and justify new, domain-relevant features that help significantly in the prediction task. Explain the physical, statistical, or empirical basis for why these engineered features improve model performance.	0.5
2.3 Avoid Leakage Check *(In Notebook #1)*	Explicitly demonstrate and document that your preprocessing steps (e.g., scaling, imputation) were fitted using only the training data before being applied to the test data.	0.5

Part 3: Model Building, Optimization, & Comparison (3.5 Points)

Task	Description	Points
3.1 Predictive Modeling *(In Notebook #1)*	Implement several distinct predictive techniques (e.g., Random Forest, XGBoost, SVR, or Deep Neural Network) relevant to your problem type.	0.5
3.2 Hyperparameter Tuning *(In Notebook #1)*	Perform hyperparameter tuning (e.g., Grid Search, Random Search, or a more advanced technique) for all predictive techniques.	1.0
3.3 Reliable Metric Comparison *(In Notebook #1)*	Compare the performance of all techniques using reliable, domain-appropriate metrics (e.g., F1-Score, AUC, MAE, or RMSE) and NOT just the default .score() method. Aim to achieve a very good predictive memorization and generalization performances, which in terms of default metrics should be above 0.9.	0.5

3.4 Visualization & Interpretability *(In Notebook #1)*	Use good visualizations (e.g., confusion matrices, residual plots, ROC curves) to measure and compare memorization and generalization performances. Analyze and report on Feature Importance for your best model, explaining why those features were important from an engineering perspective. Can your models be trusted, are they reliable? Are they physically consistent? Are they overfitting?	1.5

Part 4: Robustness and Deployment (3.0 Points)

Task	**Description**	**Points**
4.1 Robustness & Uncertainty *(In Notebook #2)*	Select the single best-performing technique from Part 3. To test its robustness and stability, you must retrain and/or re-tune this technique multiple times (e.g., using a method like Bootstrapping on the training data or various train-test splits). Quantify the predictive uncertainty by: 1. Calculating the 95% Confidence Interval for a subset of the best model's key predictions on the test set. 2. Justify the method chosen for uncertainty quantification (e.g., Bootstrapping, Dropout Sampling, or Model-Specific methods like Quantile Regression/Bayesian NN) and explain why it is appropriate for your chosen technique and problem. *Explore approaches that you prefer.*	2
4.2 Deployment Simulation *(In Notebook #3 with Deployment Data)*	Create a separate, self-contained Python notebook and a corresponding dataset that simulates the real-world deployment of your complete, trained workflow using the best model obtained in Part 3. This notebook must: 1. Take a small batch of new data without the target variable; 2. Apply all necessary preprocessing, feature engineering, and the final model seamlessly and sequentially in the correct order. 3. Demonstrate that the full workflow is deployment-ready and functional end-to-end.	1

Suggested Resources for Energy & Petroleum Data Analytics

You can choose any engineering- or energy-relevant dataset. To help you get started, here are reliable sources for datasets in relevant domains (Petroleum, Geophysics, and Industrial Engineering):

Domain	**Dataset Source/Reference**	**Type of Data**
Core Domain (Petroleum/Subsurface)	**Specific Sources:** SEG/Kaggle ML challenges (e.g., Well Log Prediction, Seismic Interpretation), publicly shared SPE papers' supplemental data, open-source data from major fields (e.g., Volve Field data).	Well Logs (petrophysical properties), Seismic Attributes (classification, regression), Production Time-series (forecasting, anomaly detection), Rock/Fluid Properties (PVT modeling).
Porous Media & Fluid Flow	**CFD/Reservoir Simulation Data:** Datasets derived from high-fidelity simulations (e.g., OpenFOAM, CMG) for building fast ML-based *surrogate models* for complex processes (permeability, relative permeability).	Digital Rocks, Flow Simulation Results, Transport Phenomena Data
Industrial/Predictive Maintenance	**NASA Prognostics Data Repository:** Contains predictive maintenance data (e.g., Turbofan Engine Degradation, bearing vibration data) for Remaining Useful Life (RUL) prediction. **IoT/Sensor data for refineries or pipelines.**	Sensor Time-series, Vibration, Acoustic Data, Failure Events (Classification/Regression)

Energy Systems Forecasting	**U.S. Energy Information Administration (EIA) Open Data / NREL Open Energy Data:** Look for granular time series data on power generation, grid load, or renewable energy output for market/grid forecasting tasks.	Power Generation/Consumption, Grid Load, Renewable Output Forecasting (Time-series)
Business/Economic Analytics (Related)	**World Bank Open Data, Federal Reserve Economic Data:** Utilize economic indicators or commodity prices (Crude Oil, Natural Gas futures) for financial modeling and forecasting/regression tasks.	Economic Time-series, Financial Indicators, Market Prediction
General ML Benchmarks	**UCI Machine Learning Repository or Kaggle Public Datasets:** Utilize high-quality, non-trivial datasets that often relate to engineering or manufacturing quality control (e.g., concrete strength prediction, material synthesis).	General Engineering, Manufacturing, and Environmental data.