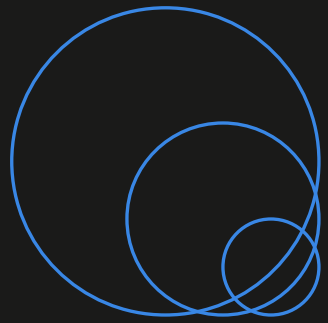
Ex.2 How to Proceed with the Exercise — Building a Surrogate Model with Volve Field Data —

Practical Reservoir Engineering — Data-Driven Simulator Series, Session 2: Exercise Slides



Today's Agenda (1/2)

- Purpose and goals of the exercise (Ex.2)
- Data used: Equinor Volve field data (continuing from Session 1)
- Design of input/output variables and preprocessing
- Exercise: building RSM / regression models
- Exercise: building a random forest model
- Exercise: building a neural network model (optional task)

Today's Agenda (2/2)

- Accuracy validation (hold-out, k-fold cross-validation)
- What the validation revealed — pitfalls of splitting time-series data

Part F

Ex.2 How to Proceed with the Exercise

› Aim of this material: enable you to carry out the Session 2 exercise (Ex.2) concretely, in line with the real Volve data in Research/References/Volve and the results of the Session 1 exercise

› Target data: from the Equinor Volve field data, the field-wide monthly production and pressure data (Volve_Production_Data_Processed.csv, 112 rows)

› Building on the insights from Session 1's Ex.1 (structure of missingness, spurious zeros, date-notation pitfall), practice the content of Parts B–E through to building a surrogate model

F1: Purpose of the Exercise (Ex.2)

■ Goal 1: From the Volve monthly production and pressure data loaded in Session 1, organize input/output data for surrogate-model building

■ Goal 2: Using the methods of Part B (RSM/regression) and Part C (random forest, optionally neural network), build a surrogate model that predicts reservoir pressure

■ Goal 3: Validate accuracy with the hold-out method and k-fold cross-validation (Part E), and compare across methods

Research/References/Volve/README.md ; Research/Deliverables 'DataDrivenSimulator_S1_Exercise_notebook.ipynb'

Flow of Today's Exercise (Ex.2)

F2: Finalizing the Dataset (Continuing from Session 1)

■ Session 1's detailed edition F2 listed Volve/SPE9/SPE10 as candidates, but we adopt the Volve data—already held in-house and usable for the exercise—as the finalized dataset

■ File used: Volve_Production_Data_Processed.csv (Sep 2007–Dec 2016, 112 monthly rows; contains reservoir pressure p, cumulative production Np/Gp/Wp, cumulative injection Gi/Wi)

■ The per-well feature table built in Session 1 (well_feature_table.csv, 7 wells) has too few rows (samples), so we keep it as reference only and use the field-wide monthly data of 112 rows as the main material for this exercise

Research/References/Volve/README.md

F3: Design of Input and Output Variables

■ Output variable (prediction target): p (psia) — reservoir pressure. In material-balance terms, a response quantity determined by cumulative production and injection

■ Input variables (candidates): elapsed_days, cumulative oil Np (STB), cumulative gas Gp (SCF), cumulative water Wp (STB), cumulative gas injection Gi (SCF), cumulative water injection Wi (STB)

■ Volve_input_to_mbal.csv holds the same kinds of variables in material-balance input form, corresponding to this exercise's input design

Research/References/Volve/README.md

Input and Output Variables

Input variables (X)	Output variable (y)
elapsed days	p (reservoir pressure, psia)
Np (cumulative oil)	
Gp (cumulative gas)	
Wp (cumulative water)	
Gi · Wi (cumulative injection)	

F4: Concrete Steps for Data Preparation

■ Load with pandas.read_csv and convert the Date column with pd.to_datetime(dayfirst=True) as in Session 1 (this file is also in DD/MM/YYYY notation)

■ Add elapsed_days—the number of days since the data start date—as a new input variable

■ Standardize the input variables with sklearn.preprocessing.StandardScaler and split into training/test data with train_test_split (see Part E1)

F5: Exercise on Building RSM/Regression Models (Validation Results)

■ Build a quadratic response-surface model with sklearn.preprocessing.PolynomialFeatures(degree=2) + LinearRegression

■ Hold-out validation (89 training / 23 test): linear regression gives RMSE 89.7 psia, R2 0.831; quadratic polynomial regression improves to RMSE 75.5 psia, R2 0.881

■ Among the inputs, Np and Gp are almost perfectly collinear (correlation 0.9998), serving as a concrete example for the multicollinearity countermeasures of Part B6 (standardization, variable reduction)

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

F5 supplement: Regression Equation (Linear Regression)

■ Training linear regression in the original units (no standardization) yields the following equation (RMSE 90.4 psia, R2 0.829, almost identical to the standardized result of RMSE 89.7 psia, R2 0.831)

■ $p\ (\text{psia}) = 4787.51 - 1.83\text{e-}08 \times \text{elapsed\_days} - 5.10\text{e-}05 \times Np - 3.38\text{e-}07 \times Gp - 2.04\text{e-}04 \times Wp + 0 \times Gi + 2.14\text{e-}04 \times Wi$

■ Note that the signs of the coefficients match physical intuition: the coefficients of production Np, Gp, Wp are negative (production lowers pressure), while that of injection Wi is positive (injection maintains pressure). Gi (gas injection) is zero throughout, so its coefficient is also 0

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

F5 supplement: Regression Equation (Quadratic Polynomial Regression, Top Terms)

Term (product of standardized input variables)	Coefficient
Intercept b0	4933.208
Wp × Wi	-515784.5
Gp × Wi	-375033.1
Wi^2	340464.4
elapsed_days × Wi	241879.7
elapsed_days × Np	-209449.8
Wp^2	191461.6
Np × Wp	167306.4
elapsed_days × Wp	-157251.6

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record). Of all 27 terms (including the constant), the top 8 by absolute coefficient value. RMSE=75.51 psia, R2=0.881

F5 supplement: The Polynomial-Regression Equation Cannot Be Used for 'Interpretation'

■ Quadratic polynomial regression is accurate (R2 0.881), but its top-term coefficients swing wildly on the order of hundreds of thousands—in contrast to linear regression, whose equation can be interpreted physically as-is

■ The cause is the multicollinearity among inputs confirmed in F5 (e.g., Np–Gp correlation 0.9998). Because the standardized interaction terms are also strongly correlated with each other, individual coefficients become unstable, and the equation's values alone cannot tell you 'which variable matters'

■ We confirmed with real data the limitation of RSM/polynomial regression: 'you can get accuracy but cannot interpret the equation.' To understand how variables act, the random forest's feature_importances_ (F6) is more reliable

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

F6: Exercise on Building a Random Forest Model (Validation Results)

■ Trained with sklearn.ensemble.RandomForestRegressor(n_estimators=300). Hold-out validation gives RMSE 49.3 psia, R2 0.949, and an OOB score of 0.91—more accurate than RSM

■ In feature_importances_, elapsed_days (0.23), Np (0.22), Gp (0.19), Wi (0.18), and Wp (0.18) are close together, while Gi (cumulative gas injection) has importance 0.00

■ Volve is mainly water-flood (water injection) with gas injection (Gi) zero throughout; real-data inspection pinpoints why the importance becomes zero—a concrete example that a variable that does not appear in the data gets no importance

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

F6 supplement: Predicted vs. Actual (Random Forest, 23 test points) (1/2)

Date	Actual p (psia)	Predicted p (psia)	Error (predicted − actual)
2007-09-01	4780.59	4772.64	-7.95
2008-01-01	4780.59	4765.57	-15.02
2008-07-01	4362.30	4354.78	-7.52
2008-08-01	4309.95	4284.65	-25.30
2008-09-01	4231.05	4155.71	-75.34
2009-03-01	4786.10	4608.08	-178.02
2009-11-01	4657.89	4626.86	-31.03
2010-04-01	4587.40	4676.65	89.25
2010-09-01	4881.97	4885.35	3.38
2011-01-01	4886.18	4866.64	-19.54
2011-03-01	4896.76	4877.81	-18.95
2011-05-01	4889.22	4890.64	1.42
2011-08-01	4910.69	4893.33	-17.36
2012-02-01	4907.06	4891.40	-15.66

F6 supplement: Predicted vs. Actual (Random Forest, 23 test points) (2/2)

Date	Actual p (psia)	Predicted p (psia)	Error (predicted − actual)
2012-04-01	4877.04	4862.64	-14.40
2013-02-01	4846.15	4854.09	7.94
2013-05-01	4893.72	4920.03	26.31
2013-06-01	4904.74	4934.73	29.99
2013-10-01	4924.32	4931.06	6.74
2014-05-01	4881.39	4943.09	61.70
2015-04-01	5052.10	5039.80	-12.30
2015-10-01	4985.67	5000.02	14.35
2016-04-01	5070.37	5106.49	36.12

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record). RMSE=49.32 psia, MAE=31.11 psia, R2=0.949, OOB=0.910

F6 supplement: Feature Importance (Random Forest)

Input variable	Importance
elapsed_days	0.231
Np (cumulative oil, STB)	0.216
Gp (cumulative gas, SCF)	0.191
Wi (cumulative water injection, STB)	0.182
Wp (cumulative water, STB)	0.180
Gi (cumulative gas injection, SCF)	0.000

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record). Gi is zero throughout Volve (no gas injection), so its importance is 0

F6 supplement: 5-fold Cross-Validation Results (Random Forest)

Fold	R2
Fold 1	0.950
Fold 2	0.901
Fold 3	0.768
Fold 4	0.897
Fold 5	0.972
Mean (std. dev.)	0.897 (0.071)

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record). Some folds are lower than the hold-out-only R2 (0.949), showing the importance of not relying on a single split

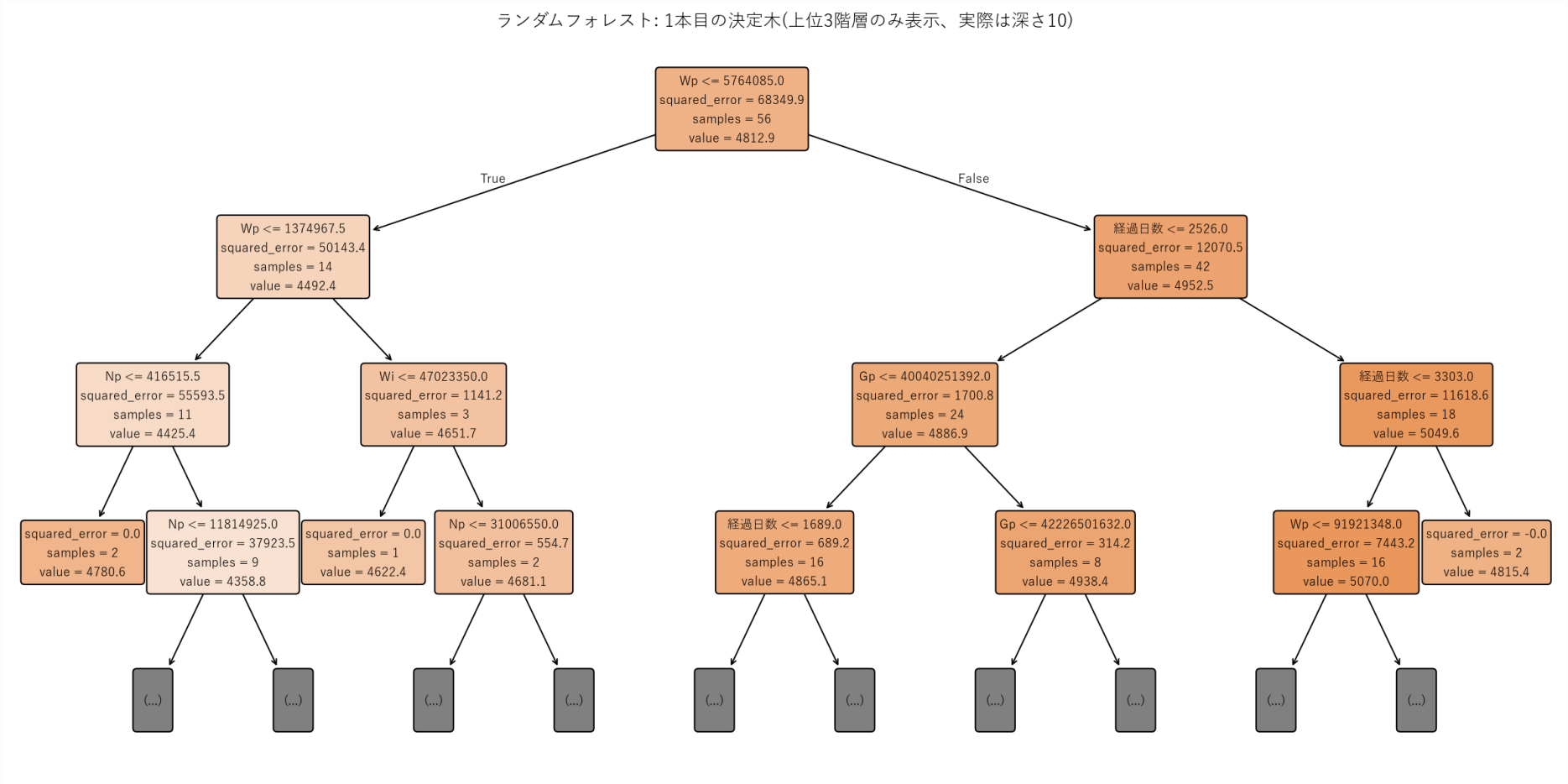
F6 supplement: A Look Inside the Random Forest

■ A random forest is ensemble learning that combines many decision trees (n_estimators=300 in this exercise); the final prediction is the average of the 300 trees' predictions (Part C2)

■ Each individual tree is built from a simple repeated rule: 'branch on some condition → branch again → upon reaching a leaf, predict the mean of the training data belonging to that leaf'

■ On the next slide we visualize the actual first tree that was trained and see which variables and thresholds it branches on

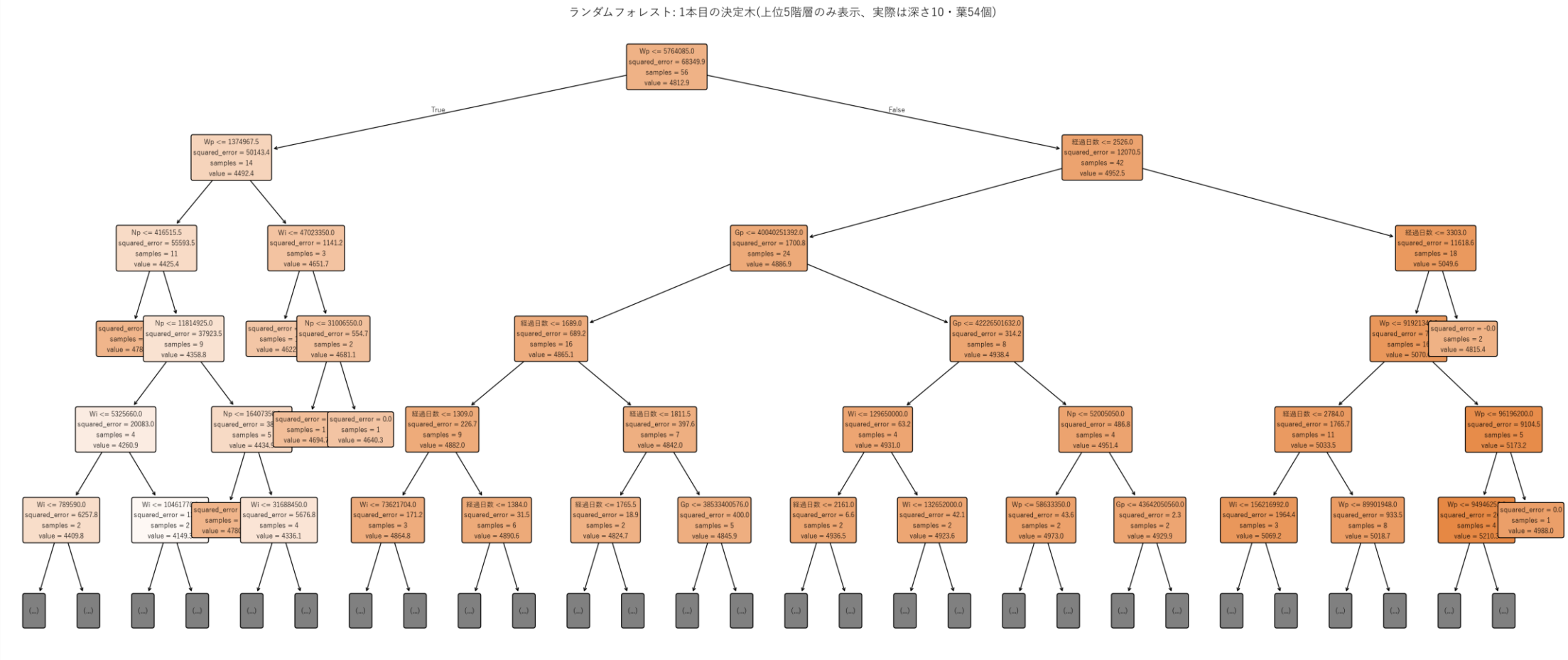
F6 supplement: Inside a Single Decision Tree (Overview)



The actual first tree of the random forest (300 trees). Only the top 3 levels are shown (the actual tree has depth 10 and 54 leaves)

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record). Generated with sklearn.tree.plot_tree

F6 supplement: Inside a Single Decision Tree (Detail)



The same tree shown down to the top 5 levels. The more branches, the finer the conditions become, and the terminal leaves make predictions based on fewer samples

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record). Generated with sklearn.tree.plot_tree

F7: Exercise on Building a Neural Network Model (Optional Task, Validation Results)

■ For participants with time and proficiency to spare, this is an optional task built with sklearn.neural_network.MLPRegressor (hidden layers of 32 and 16 nodes)

■ In hold-out validation it gave RMSE 137.2 psia, R2 0.606—below RSM (R2 0.881) and random forest (R2 0.949)

■ With only 89 training samples, the result confirms with real data Part C10's point that neural networks require large amounts of data. It also serves as a lead-in to the need for methods that incorporate prior knowledge (physical constraints), such as PINNs in Session 4

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

F8: Accuracy-Validation Exercise (Hold-out vs. k-fold Comparison)

■ For the three methods (RSM/polynomial regression, random forest, NN), compute RMSE, MAE, and R2 by the hold-out method and compare them in a table

■ Running 5-fold cross-validation for the random forest, R2 ranged 0.768–0.972 (mean 0.897), varying by fold—confirming the importance of not relying on a single hold-out result (R2 0.949)

■ Always align method comparisons on the same test data and the same evaluation metric (Part E4)

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

Model Evaluation Changes Completely Depending on the Split Method

Random 80/20 split	Even with the same data and same model	Time-series split (test from Feb 2015 onward)
RF: R2 0.949		RF: R2 -0.42
Polynomial (quadratic): R2 0.881		Polynomial (quadratic): R2 -12.15
Good accuracy		Accuracy breaks down (extrapolation)

F9: What the Validation Revealed — How to Split Time-Series Data (An Important Insight)

■ A random 80/20 split gave RF R2 0.949, but splitting chronologically—first 80% for training, last 20% (Feb 2015–Dec 2016) for testing—worsens RF R2 to -0.42 and polynomial regression to R2 -12.15

■ The cause is that the test-period pressure (max 5273.87 psia) exceeds the training-period pressure range (max 5114.90 psia), forcing the model to extrapolate outside the training range (a concrete example of the 'accuracy drop in the extrapolation region' from Parts E5 and B10)

■ A practically important lesson: do not take random-split validation at face value; for data with temporal structure, you must also validate with a time-series split

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

F10: Comparing and Visualizing Results

■ Scatter plot of predicted vs. actual (visually check accuracy by deviation from the diagonal; make one per method)

■ Compare RMSE by method with a bar chart (visualize that the random forest is the most accurate)

■ For the test interval of the time-series split, overlay the actual and predicted trends to visually confirm the divergence caused by extrapolation

Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (validation record)

Summary

■ In this exercise (Ex.2), using Volve's monthly production and pressure data (112 rows), we build surrogate models of reservoir pressure with three methods: RSM (polynomial regression), random forest, and (optionally) neural network

■ In hold-out validation we confirmed with real data that the random forest is the most accurate (R2 0.949), followed by RSM (R2 0.881), while the neural network does not improve due to insufficient data (R2 0.606)

■ This material's content has been validated on real Volve data (validation record: Research/Deliverables/DataDrivenSimulator_S2_Exercise_notebook.ipynb). The important insight that accuracy differs greatly between random and time-series splits has been reflected in the teaching material

List of References (Sources) (1/2)

■ Equinor, 'Volve field data set' https://www.equinor.com/energy/volve-data-sharing

■ energistics.org, "Equinor's Volve Field Test Data" https://energistics.org/equinors-volve-field-test-data

■ yohanesnuwara/pyreservoir (data redistribution source) https://github.com/yohanesnuwara/pyreservoir/tree/master/data/volve

■ Research/References/Volve/README.md (details of stored data, sources, license notes)

■ Research/Deliverables 'DataDrivenSimulator_S1_Exercise_notebook.ipynb' (Session 1 validation record; prerequisite for this exercise)

■ Research/Deliverables 'DataDrivenSimulator_S2_Exercise_notebook.ipynb' (record of validating this material's content on real data)

■ scikit-learn official documentation https://scikit-learn.org/stable/

■ scikit-learn, 'Cross-validation: evaluating estimator performance' https://scikit-learn.org/stable/modules/cross_validation.html

List of References (Sources) (2/2)

■ scikit-learn, 'Tuning the hyper-parameters of an estimator' https://scikit-learn.org/stable/modules/grid_search.html