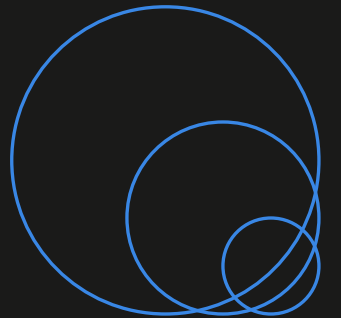




Ex.1 演習の進め方 —Volveフィールドデータ で学ぶ—

実践 貯留層工学 データ駆動型シミュレータ編 第1回 演習パワポ資料



本日のアジェンダ (1/2)

- 演習(Ex.1)の目的とゴール
- Python/Jupyter Notebook環境構築
- 使用データ: Equinor Volveフィールドデータ(演習用抜粋)
- データ読み込み・前処理
- 可視化・探索的データ分析(EDA)
- 特徴量整理と成果物

パートE

Ex.1 演習の進め方

E1: 演習(Ex.1)の目的

- 目的1: Python(Jupyter Notebook)環境を各自のPCで構築し、動作確認する
- 目的2: Equinor Volveフィールドデータ(調査部門/参考資料/Volveの演習用抜粋)を読み込み、可視化する
- 目的3: 代理モデル構築(第2回)に向けた特徴量整理の考え方を体験する

本日の演習(Ex.1)の流れ



E2: Python環境構築の手順

- Anaconda(Individual Edition)またはMiniconda(軽量版)の導入を推奨
- ``conda create -n reservoir-ml python=3.11`` のように講座専用の仮想環境を作成し、``conda activate reservoir-ml`` で有効化する
- ``conda install jupyterlab`` でJupyterLabを導入し、``jupyter lab`` で起動、ブラウザ上で動作確認する
- オンライン代替: ローカル構築なしでGoogle Colabを使う方法もある(社内データ利用時はガバナンス確認が必要)

E5～E7: 必要ライブラリ

- pandas・numpy: CSV形式のVolveデータ(生産量・坑底圧力)の読み込み・加工・集計に使用
- matplotlib・seaborn: 時系列プロット・欠損値の可視化・坑井間比較に使用
- scikit-learn: 回帰・ランダムフォレスト等
- `pip install pandas numpy matplotlib seaborn scikit-learn` で一括導入できる

使用データ

Equinor Volveフィールドデータ(演習用抜粋)

- › ノルウェー北海Volve油田(2008～2016年操業)の実データをEquinorが2018年に公開
- › 公開データ本体は約4万ファイル規模のため、本演習では生産量・坑底圧力データの抜粋(5ファイル)のみを使用する

E8: Volveデータセットの全体像と本演習での扱い

- 坑井検層・地震探査・生産/圧入履歴・シミュレーションモデル等を含む大規模公開データセット (Equinor Open Data Licenceのもとで研究・教育目的の利用が認められている)
- 本演習では、この中から生産量・坑井別坑底圧力に関するCSV5ファイル(演習用抜粋)を教材として使用する
- 抜粋データは第三者(pyreservoirリポジトリ)による再配布であり、原データの権利はEquinor Open Data Licenceに基づく点に留意する

格納データの構成(1) 生産量・PVT関連

- Volve_production.csv: 月次データ、2008年1月～2008年11月(全11行)。圧力 p ・累積生産量 $N_p/G_p/W_p$ ・累積圧入量 G_i/W_i 等
- Volve_Production_Data_Processed.csv: 月次データ、2007年1月～2016年12月(全112行)。より長期間の生産量・圧力推移を追える
- volve_input_to_mbal.csv: マテリアルバランス解析向けに整形した入力データ(2007年9月～2016年12月、112行)。上記と同系列の変数をモデル入力形式で保持

格納データの構成(2) 坑井別坑底圧力

- Volve_BHFP_data.csv: 坑井7本(15/9-F-1C・F-4・F-5・F-11・F-12・F-14・F-15D)の日次坑底流動圧、全15,634行(2007～2016年)
- AVG_DOWNHOLE_PRESSURE列は欠損が約42.6%存在するが、坑井別に見ると水圧入井2本(15/9-F-4・F-5)で100%欠損、残り5本の生産井はほぼ欠損なし 一欠損の構造(坑井属性)を確認する題材として最適
- Volve_BHP_15_9-F-1C.csv: 坑井15/9-F-1C単独の抜粋(746行、2014年1月～2016年12月)。まず1坑井の可視化から始める導入教材として使う

5ファイルの役割分担

生産量・PVT

Volve_production.csv

Volve_Production_Data_Processed.csv

坑井別坑底圧力

Volve_BHFP_data.csv

Volve_BHP_15_9-F-1C.csv

MBAL入力

volve_input_to_mbal.csv

E11: データ読み込み・前処理の進め方

- まずVolve_BHP_15_9-F-1C.csv(1坑井・746行)のような小さいファイルから`pandas.read\_csv`で読み込み、操作に慣れる
- DATEPRD/Date列を`pd.to\_datetime`で日付型に変換する。注意:
Volve_Production_Data_Processed.csvはDD/MM/YYYY表記のため`dayfirst=True`を指定しないと月次データが疑似的な日次データに誤読される(実検証で確認済み)
- AVG_DOWNHOLE_PRESSURE等の欠損値を確認する 一坑井別に見ると、水圧入井2本(15/9-F-4・F-5)が100%欠損、他5本はほぼ欠損なしという構造的な偏りが実データ検証で判明している
- NPD_WELL_BORE_NAME列で坑井ごとにgroupbyし、坑井別のデータフレームに整理する
- 数値としての「0」も物理的にありえない欠損・異常値であることがある(坑井停止区間等)。`isna()`だけに頼らず、実際にプロットして目視確認する

E12: 特徴量整理

- 入力候補: ON_STREAM_HRS(稼働時間)、坑井、経過日数・期間
- 出力候補: AVG_DOWNHOLE_PRESSURE(坑底圧力)、月次の累積生産量 $N_p \cdot G_p \cdot W_p$
- 「稼働時間・時間経過 → 坑底圧力」のような単純な入出力関係の整理を、第2回の代理モデル構築の土台として体験する

E13: 可視化(探索的データ分析)

- matplotlib/seabornで坑井別の坑底圧力の時系列プロットを作成する
- 欠損区間を可視化し、どの坑井・期間でデータが取得できていないかを確認する
- 複数坑井(7本)の圧力推移を重ね描きし、坑井間の傾向差を比較する
- NaNだけでなく、物理的にありえない「0」の混入(坑井停止区間等)も可視化で確認する 一実データ検証で複数坑井に見つかっている

検証メモ: 演習を実際に実行して分かったこと

- 本資料の内容に沿って、実際にVolveデータでEx.1を一通り実行し、調査部門/成果物/データ駆動型シミュレータ編_第1回_演習notebook.ipynbとして記録した
- 気づき1(前処理): Volve_Production_Data_Processed.csvの日付はDD/MM/YYYY表記。dayfirst=Trueを付け忘れると月次データが疑似的な日次データに崩れる
- 気づき2(欠損の構造): 坑底圧力の欠損(42.6%)は、水圧入井2本(15/9-F-4・F-5)への100%集中によるもので、単純補間では対処できない構造的欠損だった
- 気づき3(0の混入): 欠損はNaNだけでなく、物理的にありえない「0」としても現れる(複数坑井の停止区間)ため、目視でのEDAが欠かせない

演習ノートブックの構成案



参考資料一覧(出典) (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(データ再配布元)
<https://github.com/yohanesnuwara/pyreservoir/tree/master/data/volve>
- 調査部門/参考資料/Volve/README.md(格納データの詳細・出典・ライセンス注意)
- 調査部門/成果物/データ駆動型シミュレータ編_第1回_演習notebook.ipynb(本資料の内容を実データで検証した記録)
- Anaconda公式ドキュメント <https://docs.anaconda.com/anaconda/install/>
- Jupyter公式ドキュメント <https://jupyter.org/install>
- scikit-learn公式ドキュメント <https://scikit-learn.org/stable/>

参考資料一覧(出典) (2/2)

■ matplotlib公式ドキュメント <https://matplotlib.org/>