**ANALISIS DATASET BATU PADA SUMUR**

Lithologi adalah gambaran umum tentang sifat fisik batuan di suatu daerah tertentu. Sifat-sifat ini biasanya mencakup komposisi batuan, ukuran butir, warna, atau teksturnya. Dalam dunia pengeboran sumur (well logging), lithologi merujuk pada ringkasan sifat fisik utama batuan. Jenis batuan yang paling sering ditemui biasanya adalah batu pasir (sandstone), batu tulis (slate), basal, atau batu kapur (limestone).

Memahami lithologi di dalam sumur sangat penting dalam proses pengeboran. Cara paling sederhana untuk mengetahui jenis lithologi adalah dengan menganalisis sampel batuan dari formasi tersebut. Namun, mengambil sampel ini tidak selalu mudah dan kadang-kadang tidak memungkinkan. Oleh karena itu, para ahli mengembangkan teknik lain untuk memprediksi jenis lithologi. Di kompetisi ini, kami mengusulkan untuk menggunakan data log sumur (well log data) untuk menebak jenis lithologi. Jadi, dengan data yang ada, kita mencoba menentukan batuan apa yang ada di dalam sumur tanpa harus mengambil sampel langsung.

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Metodelogi** | **Temuan** | **Aksi** |
| 1 | Memvisualisasikan dataset dengan visualisasi 3D menggunakan fitur (X\_Loc, Y\_Loc, Z\_Loc). | Visualiasi 3D menunjukkan bahwa dataset berbentuk seperti sumur yang terdiri dari beberapa sumur. Hal ini menunjukkan bahwa dataset merupakan dataset yang berasal dari sejumlah sumur yang berbeda. Dengan perbedaan sumur kemungkinan terdapat perbedaan karakteristik dari setiap sumur. | * Melakukan clustering berdasarkan X\_Loc dan Y\_Loc yang dikonversi menjadi integer berdasarkan lokasi sumur. * Perlu melakukan eskplorasi lebih lanjut mengenai **“apakah benar setiap sumur memiliki karakteristik berbeda?”** |
| 2 | Mendeskripsikan statistik pada seluruh dataset | Terdapat fitur yang memiliki NaN value lebih dari 90% sehingga bisa dikatakan kurang informatif yaitu:   * **(** **> 90%)**: DTS, SGR, ROPA, RMIC * **(50 – 80%)**:RXO, DCAL, RSHA, NPHT, BS, PEF | * Perlu dilakukan pengecekan terhadap apakah NaN tersebut tersebar ke seluruh kelas atau hanya di kelas tertentu saja **(Kemungkinan Penting untuk Deskripsi Salah Satu Kelas)** |
| 3 | Membagi dataset menjadi per kelasnya dan melakukan pemeriksaan statistik pada setiap fitur di setiap kelas. | Terdapat sejumlah fitur pada kelas tertentu yang 100% NaN yaitu:   * **99000 (Tuff):** DTS * **70032 (Chalk):** ROPA, RXO, RMIC, DTS, SGR * **90000 (Coal):** ROPA, DTS, SGR * **88000 (Halite):** ROPA, RXO, RMIC, DTS, SGR * **86000 (Anhydrite):** RMIC, SGR * **93000 (Basement):** ROPA, PEF, RXO, RMIC, DTS, SGR, RSHA | * Menghapus kolom yang memiliki 100% nan pada kelas tertentu karena akan sulit untuk digunakan sebagai predictor. |
| 4 | Melakukan pembagian dataset menjadi per cluster/sumur dan kemudian diurutkan berdasarkan Z\_LOC. | Setelah diurutkan terdapat beberapa data yang memiliki lonjakan yang tidak normal. Contoh fitur GR (20,30, 5000, 60) yang dianulir sebagai kesalahan perhitungan dan menjadi outlier. | Melakukan deteksi outlier dan melakukan pergantian nilai dengan nilai normal di sekitarnya. |
| Terdapat sejumlah fitur yang memiliki **kesamaan karakteristik** di kedalaman yang sama tetapi ada juga yang memiliki **karakteristik yang berbeda**.   * Karakteristik sama: |
| 5 | Melakukan perhitungan korelasi terhadap ['Z\_LOC','GR', 'RHOB', 'NPHI', 'DTC', 'RDEP', 'PEF', 'Lithology\_code'] | * Terdapat korelasi kuat 1 pada GR dan RDEP dan ternyata ditemukan bahwa nilai GR dan RDEP adalah sama. * Terdapat korelasi kuat DTC dan Z\_LOC * Terdapat korelasi kuat Z\_LOC dan RHOB pada beberapa cluster yang diperiksa | * Memanfaatkan fitur dengan korelasi tinggi untuk memprediksi dataset NaN * Menghapus fitur RDEP |
| 6 | Membuat file csv yang terpisah per cluster per kelas | * SP NaN 44% di 80000 dan NPHI 18% * NPHI NaN 45%, SP 27%, CALI 17%, dan DRHO 18% di 74000 * NPHI NaN 25% di 70032 * NPHI NaN 21% dan SP NaN 38% di 70000 * RHOB (20%), NPHI (45%), DTC (20%), SP (25%), DRHO (20%) dan CALI (20%) di 65030 * RHOB (6%), NPHI (23%) di 65000 * RHOB (7%), NPHI (25%), SP (18%) di 30000 | * **Fitur dengan NaN rendah** seperti RHOB dan CALI harus diprioritaskan. * **Fitur dengan NaN sedang** seperti NPHI dan DTC bisa dipertahankan dengan imputasi berbasis litologi. |
| 7 | Memisahkan kelas 93000 (kelas dengan jumlah terkecil) sendiri | * Pada 93000 tidak terdapat sama sekali nilai RHOB | * Drop fitur RHOB |

PIPELINE SCALUP

1. Clustering menggunakan DBSCAN berdasarkan nilai (X\_LOC dan Y\_LOC) dan lakukan pembulatan nilai X\_LOC dan Y\_LOC ke integer.
2. Split dataset berdasarkan cluster yang telah dibuat.
3. Order setiap dataset berdasarkan nilai Z\_LOC.
4. Fill NaN Value menggunakan logika berikut:
   1. Lakukan forward fill jika label di data setelahnya nilai nan memiliki label yang sama
   2. Lakukan backward fill jika label di data sebelum nilai nan memiliki label yang sama.
5. Lakukan split dataset berdasaran class dalam setiap clusternya.
6. Lakukan pemeriksaan outlier menggunakan IQR dan jika didapati data merupakan outlier maka lakukan interpolasi terhadap data tersebut.
7. Lakukan pengisian nilai nan menggunakan KNN menggunakan fitur yang relevan.
8. Gabungkan setiap dataset pada setiap fitur menjadi dataset per fitur.
9. Lakukan pengisian nilai nan berdasarkan KNN menggunakan nilai (X\_LOC, Y\_LOC, dan Z\_LOC).
10. Gabungkan menjadi dataset keseluruhan.
11. Lakukan normalisasi (opsional).

Saran Preprocessing Data Test

1. **Gunakan Parameter DBSCAN yang Sama:** Gunakan parameter yang sama (eps=1, min\_samples=5) seperti pada training untuk memastikan konsistensi clustering.
2. **Clustering Bersama dengan Training:** Untuk memastikan data test mendapatkan cluster yang sesuai dengan data training, gabungkan data test dengan data training (tanpa label) untuk clustering, lalu pisahkan kembali.
3. **Simpan Hasil Clustering:** Tambahkan kolom Cluster\_DBSCAN ke data test.
4. **Urutkan Data Berdasarkan Z\_LOC:** Dalam setiap cluster, urutkan data berdasarkan Z\_LOC.
5. **Terapkan Forward Fill:** Isi NaN dengan nilai dari baris sebelumnya jika jarak Z\_LOC tidak melebihi threshold (misalnya, 10).
6. **Terapkan Backward Fill:** Isi NaN yang tersisa dengan nilai dari baris berikutnya, juga dengan batasan jarak Z\_LOC.
7. **Deteksi Outlier:** Untuk setiap kolom numerik dalam cluster, hitung batas bawah dan atas menggunakan metode IQR (Interquartile Range).
8. **Ganti Outlier dengan NaN:** Nilai yang berada di luar batas IQR diganti dengan NaN.
9. **Interpolasi:** Gunakan interpolasi linear untuk mengisi nilai NaN yang baru dibuat dari outlier.
10. **Gunakan Fitur Prediktor:** Gunakan fitur prediktor yang sama seperti pada training (misalnya, NPHI menggunakan RHOB, DTC, Z\_LOC).
11. **Standarisasi Data:** Standarisasi fitur prediktor dan target sebelum imputasi.
12. **Terapkan KNN Imputation:** Gunakan KNN untuk mengisi nilai NaN.
13. **Kembalikan ke Skala Asli:** Setelah imputasi, kembalikan data ke skala aslinya.
14. **Gabungkan Data:** Gunakan pd.concat untuk menggabungkan semua DataFrame dalam test\_cluster\_dfs.
15. **Hapus Kolom Tidak Relevan:** Hapus kolom seperti Cluster\_DBSCAN yang tidak diperlukan untuk langkah selanjutnya (misalnya, training model).
16. **Simpan Hasil:** Simpan dataset test yang telah dipreproses ke file CSV.