Global Oil Wells: A Machine Learning and Network Theoretic Analysis of Geolocation and Oil Well Characteristics

Benjamin Hunt

April 7, 2025

Contents

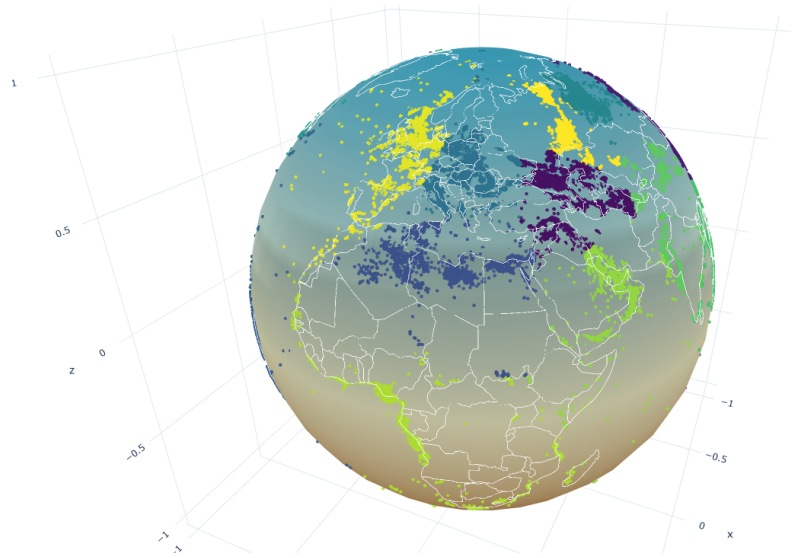

Figure 1: 3D visualization of k-means clustering applied to the full dataset, overlaid on a scaled representation of the Earth. For further details, see Section 4.2.

1 Introduction and Motivation

The global oil industry is a cornerstone of the world economy, driving energy markets, geopolitics, and industrial growth through its production capacity. In 2025, global oil demand is projected to reach 103.9 million barrels per day (mb/d) (International Energy Agency, 2025), underscoring the immense scale of this sector. Furthermore, the industry's economic significance is evident from its 2024 performance, with oil and gas companies distributing nearly $349 billion in dividends and buybacks (Deloitte, 2025), highlighting the financial stakes tied to production efficiency. Understanding the factors that govern oil output is critical for optimizing extraction, forecasting trends, and informing strategic decisions. Among these factors, geolocation, which encompasses latitude, longitude, well depth, and local geological conditions, plays a pivotal role in determining oil well productivity.

Recent advances in machine learning (ML) offer a transformative approach to uncovering patterns in oil well performance. While traditional geological and statistical methods have long supported exploration and production forecasting, they often demand significant domain expertise and labor-intensive interpretation. In contrast, machine learning provides an efficient, data-driven alternative, excelling at pattern recognition, managing vast datasets, and enhancing predictive accuracy.

This research investigates the interplay between geolocation attributes and oil output through machine learning techniques. By analyzing a comprehensive dataset of oil wells from diverse regions, we seek to pinpoint the geospatial and geological factors most influential to production levels. The study evaluates a range of ML models—including regression, decision trees, and neural networks—to assess their efficacy in predicting oil yield based on geolocation features.

The outcomes of this analysis hold practical implications for oil and gas companies, policymakers, and geoscientists, offering actionable insights for exploration and production strategies. Furthermore, this work advances the application of geospatial analytics in energy resource management, highlighting machine learning's potential to boost efficiency and sustainability in the oil sector.

2 Data

2.1 Datasets and Data Cleaning

The data used in this report are from the following three datasets:

1. 'Wells.csv' (Wells): Large global dataset for detailed information on location and geo-spacial well information (720k Rows)

2. 'adhoc_reports__wells_drilled_by_operator_with_depth.csv' (Ad-Hoc): Large global dataset for well location and operator information (1,028k Rows)

3. 'GTA_Oil_Data_03Jan2018.csv' (GTA): Smaller inter-country directional oil transport dataset between 2015 and 2018 (19k Rows)

The first two datasets, Wells (1) and Ad-Hoc (2), contain similar types of information, with Wells (1) providing more detailed geospatial data for each well. These two datasets will be the **main datasets** used throughout this report. The primary overlap between these datasets includes well locations (longitude and

latitude), the initial drilling date, the country of origin, and the total vertical depth (TVD). To consolidate this information, a merged dataset (**Merged**) was created, combining well locations and depths while removing missing data. This refined dataset serves as the foundation for subsequent modelling.

The following figures give a visual representation of the 'missingness' within the data and the correlation within the datasets between missing data on each row.

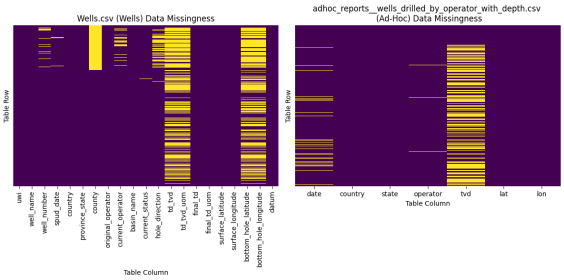

Figure 2: Data Missingness Matrix for the Wells and Ad-Hoc Ad-Hoc datasets. GTA is a complete dataset and hence not included here

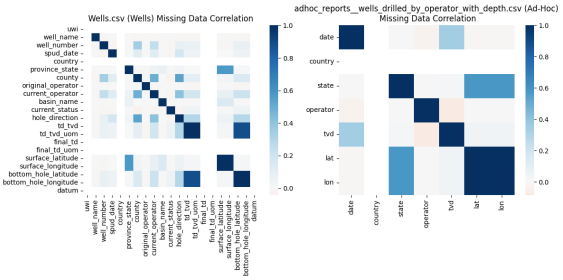

Figure 3: Data Missingness Correlation Matrix between columns in the Wells and the Ad-Hoc datasets. Again, GTA not included due to completedness.

2.2 Exploratory Data Analysis

Before beginning this individual report, we were divided into groups to select a dataset and develop a foundational understanding before starting our own analyses. As a team, we collaborated on the exploratory data analysis to gain deeper insights into the dataset. With over a million rows in total, the vast amount of data required us to use graphical representations to identify key patterns and trends.

2.2.1 Global Location

A crucial link between the two largest datasets was the country field, which provided a valuable basis for comparison and further analysis.

In the figure 4 above, we can get a intuitive understanding on the country spread of the data. North America, specifically the United States and Canada dominate this dataset with over 64% of the rows (where the country field is populated). This isn't to say that the US and Canada have the majority

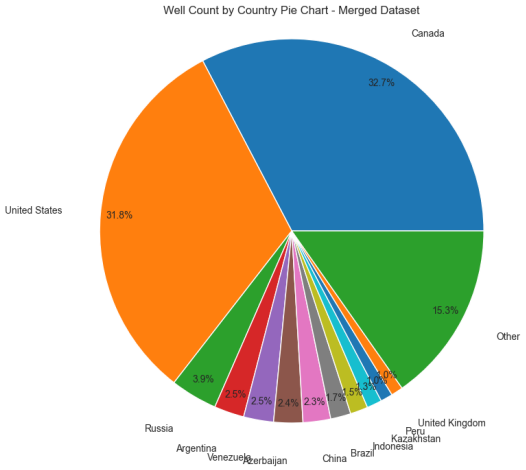

Figure 4: Well count by country pie chart.

Figure 5: Sample of 2000 wells from the Merged dataset displayed on a world map.

of the oil wells globally as a general statement but this dataset contains a considerable number of North American wells.

To provide an intuitive understanding of the spatial distribution of the data, Figure 5 presents a subset of 2000 wells plotted as markers on a world map. This sample offers insight into the physical spread of the data, revealing a high density of wells in Canada and the United States. This observation aligns with the country-level distribution shown in the pie chart figure 4.

2.2.2 Dates of Discovery and Geopolitical Links

The data within the two main datasets (Wells and Ad-Hoc) contains the date in which the wells were initially drilled. Below, in figure 6, we can see the distribution of these well discoveries overtime.

While the hypothesis ultimately explored in this report does not center on geopolitical or macroeconomic factors, it remains important to consider the major influences that have shaped the timeframe covered by these datasets.

The following events significantly influenced oil well discovery, as reflected in the drilling trends shown in figure 6.

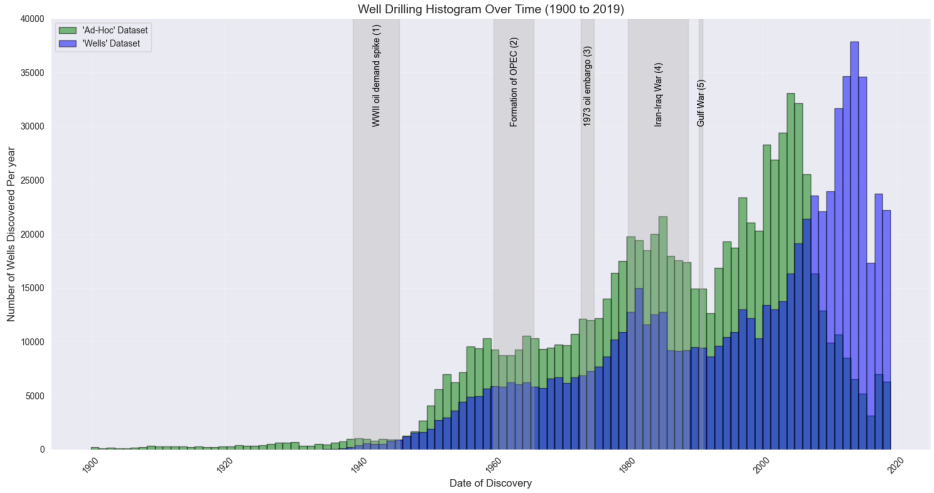

Figure 6: Histogram of oil well discoveries from 1900 to 2019, comparing the Wells and Ad-Hoc datasets. Shaded regions highlight major geopolitical events affecting drilling activity.

1. **WWII Oil Demand Spike (1939–1945):** During World War II, global oil demand surged due to military needs, driving increased drilling activity, particularly in the U.S. and Middle East. This period marked a shift toward more systematic exploration to meet wartime requirements (Wikipedia Contributors, 2023d).

2. **Formation of OPEC (1960–1965):** The establishment of OPEC in 1960 shifted control of oil production to member countries, leading to higher prices and encouraging exploration in non-OPEC regions like the North Sea and Alaska, as companies sought to diversify supply sources (Wikipedia Contributors, 2023e).

3. **1973 Oil Embargo (1973–1974):** The 1973 oil embargo by Arab OPEC nations caused a sharp rise in oil prices, prompting a surge in drilling in non-OPEC regions to reduce dependency on Middle Eastern oil, significantly boosting global exploration efforts (Wikipedia Contributors, 2023a).

4. **Iran-Iraq War (1980–1988):** The Iran-Iraq War disrupted oil supplies from the Persian Gulf, leading to increased exploration in alternative regions such as the North Sea and offshore fields, as global markets sought to mitigate supply risks (Wikipedia Contributors, 2023c).

5. **Gulf War (1990–1991):** The Gulf War further disrupted Middle Eastern oil production, driving exploration in politically stable regions like North America and the North Sea, as companies aimed to secure more reliable oil sources (Wikipedia Contributors, 2023b).

2.2.3 'Deeper Dive' Into Location and Depth

Earlier in figure 4 and 5 we have seen that the wells contain a high density of wells in the US and Canada. Taking this a subset of the data, I decided to explore deeper into the relationships in the data. To start with, I plotted a heat map over north america to get a better understanding of the distribution within individual countries.

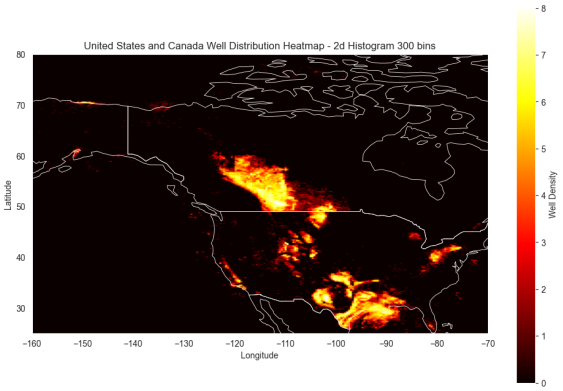

Figure 7: US and Canada well location heat map. A 2d histogram with 300^2 bins accross North America.

I initiated this in-depth analysis by creating a heat map overlaid on a subset of data from the US and Canada. As shown in Figure 7, it is evident that there are distinct **clusters** of high-density areas where wells are particularly prominent. Additionally, the map reveals a notable population of wells that do not lie on the North American landmass, suggesting that the dataset encompasses offshore oil rigs (this is also evident in figure 5).

Continuing this analysis, I then began to include well depth (Total Vertical Depth or TVD) in relation to the location. The results found in the following two plots formed the foundation for the hypothesis that is investigated in this paper.

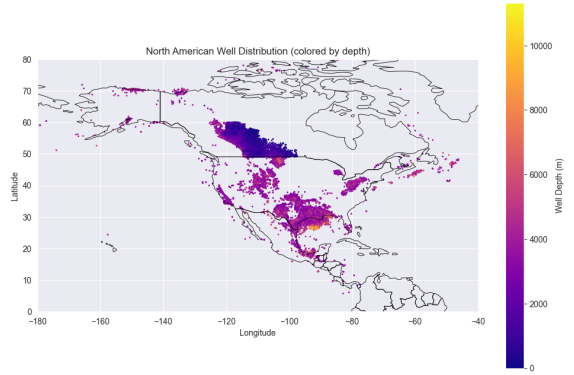

Figure 8: A 2D scatter plot of longitude, latitude with the colour of the marker representing the depth of the well.

Figures 8 and 9 present 2D and 3D visualizations of the relationship between geolocation and Total Vertical Depth (TVD) for wells across the US and Canada. In Figure 8, a 2D depth map reveals distinct clusters of wells with similar depths to their nearest neighbors, indicated by consistent coloring. This pattern is especially striking near the northwestern US-Canada border, where a dense concentration of Canadian wells forms a prominent dark blue region, signifying shallower depths. In contrast, the US appears to host deeper wells overall, with a notably "yellower" area in the south suggesting a cluster of greater depths. The 3D plot in Figure 9 enhances these observations, with red circles highlighting these clusters for clarity. Although the perspective in the 3D view slightly obscures precise locations, a top-down examination confirms that the shallower wells align with the dark blue region identified in Figure 8.

3 Hypothesis

Following the findings from the exploratory data analysis, the following hypothesis is proposed for the report:

"*Can machine learning models and network theoretic approaches, leveraging geolocation (latitude, longitude) and well depth data, predict well depth variations across oil-producing regions.*"

In the remainder of this paper, we explore a range of machine learning methods alongside a network-theoretic approach, progressing from basic to more advanced techniques, to identify a suitable solution to this question.

4 Methodology And Approach

We now introduce and justify the use of various machine learning approaches to find the best **regression** we can find to find to solve the following generic problem:

Suppose the true relationship between our data points (or features) $\mathbf{x}_i = (x_i^{(0)}, x_i^{(1)}, \ldots, x_i^{(k)})$ and true 'labels' y_i expressed as:

$$y_i = f(\mathbf{x}_i) + \epsilon$$

where $f(\mathbf{x}_i)$ is a function (linear or non-linear) that maps data points to labels, for example $f(\mathbf{x}_i) = \beta_0 + \beta_1 x_i^{(1)} + \beta_2 x_i^{(2)2}$ for a quadratic case, or another form like $f(\mathbf{x}_i) = \beta_0 e^{\beta_1 x_i^{(0)} + x_i^{(1)}}$. For generality, we keep $f(x)$ unspecified, denoting the true underlying relationship, while the noise ϵ, assumed to be normally distributed, accounts for the variability inherent in any empirical dataset. We then seek to find an equation of the form:

$$\hat{y}_i = \hat{f}(\mathbf{x}_i)$$

that minimizes some cost (error) function, for example we will use, but not exclusively, the Mean Squared Error (MSE) metric:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

To explore the spatial variation of well depths across a region, we construct a dataset of geographical features and corresponding depth measurements. The features capture each well's location, while the labels represent its vertical depth. This formulation facilitates modeling the relationship between spatial coordinates and depth, leveraging techniques such as regression.

$$\mathbf{x}_i = (x_i^{(0)}, x_i^{(1)}) = (\text{longitude}_i, \text{latitude}_i)$$
$$y_i = \text{TVD}_i$$

where:

- $\mathbf{x}_i \in R^2$ is the feature vector for the i-th well, with $x_i^{(0)}$ as the longitude and $x_i^{(1)}$ as the latitude, both measured in degrees
- y_i is the Total Vertical Depth (TVD) of the i-th well, measured in units such as meters or feet, representing the true vertical distance from the surface to the well's bottom
- $i = 1, 2, \ldots, n$, where n is the total number of wells in the dataset.

Baseline Model

As a baseline for comparison, we define the **Country Average (CA)** model as follows:

For well i let $c(i)$ be the set of wells in the country that i is located. The predicted output for well i under the Country Average model is given by:

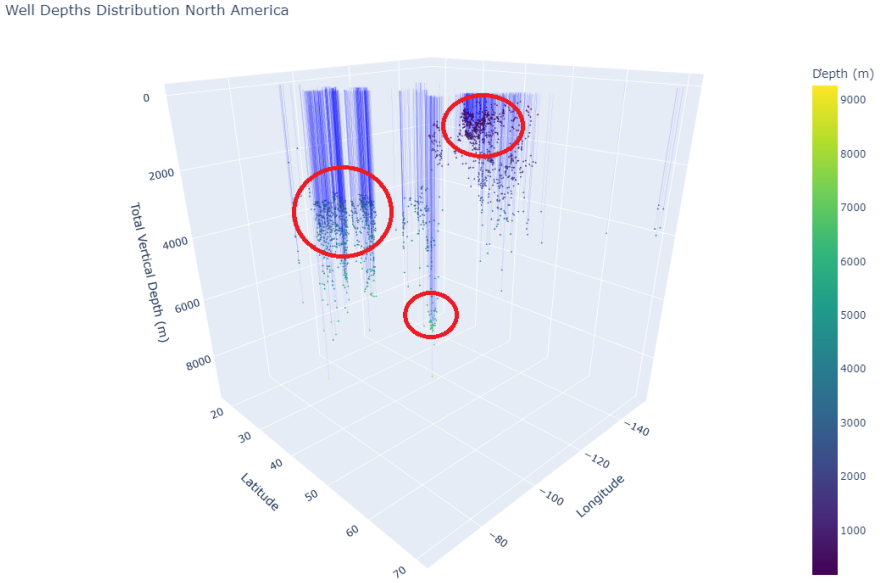

Figure 9: A 3D scatter plot depicting longitude, latitude, and well depth for a sample of 2,000 wells across North America. Vertical traces highlight the vertical extent of each well. Clusters of wells with comparable locations and depths are accentuated by red circle annotations.

$$\hat{y}_i = \frac{1}{|c(i)|} \sum_{j \in c(i)} y_j$$

where:

- $|c(i)|$ is the number of wells in country $c(i)$ in the training data.
- The summation runs over all wells j in the training set that belong to country $c(i)$.

This model assigns the average oil output of all wells in the same country as the prediction for a given well.

4.1 k-Nearest Neighbour Regression (k-NNR)

4.1.1 k-NNR Model Design

The first model we will introduce is an adaptation of the well-known k-Nearest Neighbour algorithm. On it's surface, this is the most basic model that we will use and the most intuitive based on the initial EDA.

We will compare and select one of the following two variations:

1. Uniformly (U) weighted average

$$\hat{y}_i = \frac{1}{k} \sum_{j \in N_k(\mathbf{x}_i)} y_j = \hat{f}_{\mathrm{U}}(\mathbf{x}_i | k)$$

2. Inverse distance (ID) weighting

$$\hat{y}_i = \frac{\sum_{j \in N_k(\mathbf{x}_i)} w_j y_j}{\sum_{j \in N_k(\mathbf{x}_i)} w_j} = \hat{f}_{\mathrm{ID}}(\mathbf{x}_i | k)$$

Where

$$w_j = \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)}$$

where $N_k(\mathbf{x}_i)$ is the set of k nearest neighbors to $\mathbf{x}_i$, in our implementation determined by the L2 Norm (Euclidean distance):

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_i^{(0)} - x_j^{(0)})^2 + (x_i^{(1)} - x_j^{(1)})^2}.$$

where k is selected to balance bias and variance. This approach enables depth prediction from spatial coordinates and reveals clustering patterns in the data. We choose to also include the Inverse

4.1.2 K-NNR Model Justification and Parameters

We employ k-NNR due to the observed spatial autocorrelation in the dataset, where proximal data points exhibit similar characteristics, suggesting that local neighbourhoods effectively capture the underlying variability in well depths. Figure 9 reveals a discernible relationship between geographical location and the proximity of neighboring wells, suggesting that spatial adjacency influences well characteristics.

To optimize the choice of k in k-NNR, we minimize the expected loss over the training data, for this we will use the **mean absolute error (MAE)** as the loss function, hence we need to optimize the following for the two models

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - \hat{f}_{\mathrm{m}}(\mathbf{x}_i | k) \right| \quad : \quad \mathrm{m} \in \{\mathrm{U}, \mathrm{ID}\}$$

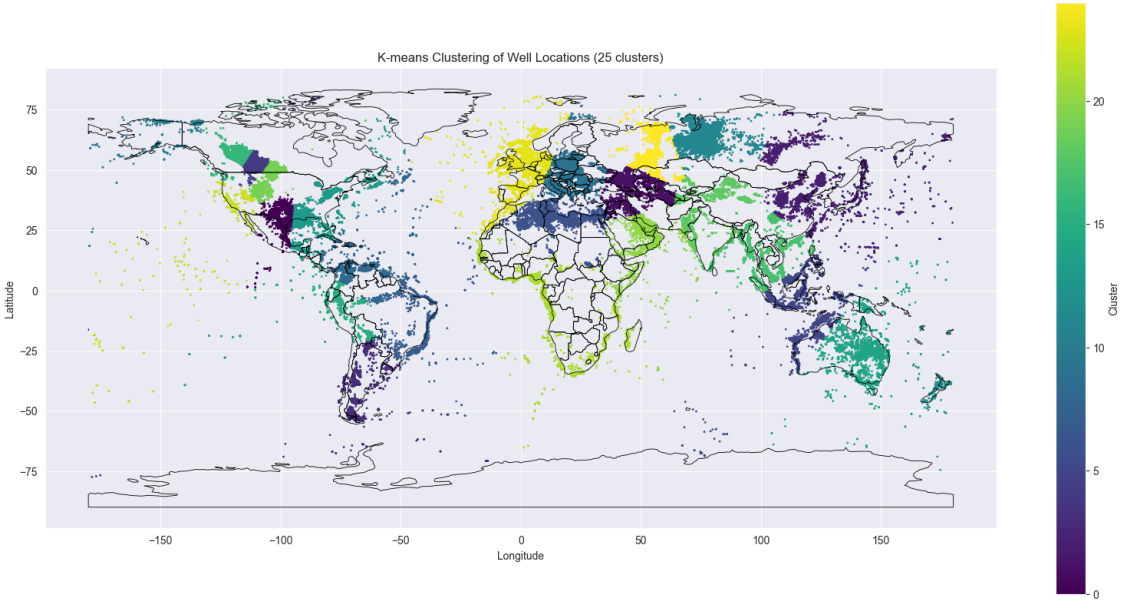

Figure 10: k-Means Clustering performed on the full merged dataset plotted as a coloured scatter plot on the world map (k = 25 clusters).

4.2 k-Means Clustering Regression (k-MCR)

4.2.1 k-MCR Model Design

Typically, a k-Means Clustering approach is used to discover properties, especially groupings within a dataset over certain features. Its unsupervised nature means that it does not rely on predefined labels, instead identifying inherent structures based on feature similarity. In the context of k-Means Clustering Regression (k-MCR), we extend this technique by integrating clustering with regression to model.

The k-Means algorithm partitions a dataset into k clusters by iteratively optimizing cluster assignments and centroids. Below is a pseudocode representation:

Algorithm 1 k-Means Clustering

1: **Input:** Dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, number of clusters k
2: **Output:** Cluster assignments $C_1, C_2, \ldots, C_k$, centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$
3: Randomly initialize k centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$
4: **repeat**
5: *Assignment Step:* For each $\mathbf{x}_i$, assign to cluster C_c where $c = \arg\min_{c'} \|\mathbf{x}_i - \boldsymbol{\mu}_{c'}\|_2^2$
6: *Update Step:* For each cluster c, recompute centroid $\boldsymbol{\mu}_c = \frac{1}{|C_c|} \sum_{i \in C_c} \mathbf{x}_i$
7: **until** centroids converge or maximum iterations reached
8: **Return:** Clusters $C_1, C_2, \ldots, C_k$, centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$ =0

After obtaining the k centroids $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k$, we perform regression within each cluster, specifically fitting a **linear regression** model to the data in each cluster. With these models established, computing the output function $\hat{f}(\mathbf{x}_i)$ is straightforward: we determine the nearest centroid to $\mathbf{x}_i$ using the L2 norm and then either apply the corresponding pre-fitted linear regression model or use the cluster's uniform mean to predict the output. This is similar to the approach in 4.1.1, we will obtain and compare $\hat{f}_{\text{U}}$ (uniform average) and $\hat{f}_{\text{LR}}$ (linear regression based function).

4.2.2 k-MCR Model Justification and Parameters

Similarly to k-NNR due, we adopt k-Means Clustering Regression (k-MCR) due to the observed spatial clustering in the dataset, where wells within the same geographical clusters exhibit similar depth characteristics, suggesting that partitioning the data into local clusters effectively captures the underlying variability in well depths. Figure 9 again can be used to suggest relationships between geographical location and the proximity of neighbouring wells, indicating that spatial clustering influences well characteristics.

In Figure 10, the scatter plot illustrates the global clustering of well locations using k-Means with $k = 10$, as indicated by the color-coded clusters. While the number of clusters may be insufficient to capture detailed insights, the visualization effectively demonstrates the spatial distribution of wells across the dataset and provides an initial view of the clustering output generated by the first stage of k-MCR.

Similar to k-NNR we are searching for the 'best' value of k, thus, we optimize the following mean absolute error (MAE) for the cluster-specific regression models:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{f}_m(\mathbf{x}_i|k, C_w(i))\right| : m \in \{\text{U}, \text{LR}\}$$
$$\text{and} \quad C_w \in \{C_1, C_2, \ldots, C_k\}$$

where $\hat{f}_m(\mathbf{x}_i|k, C_w(i))$ is the prediction of the linear regression model fitted to the cluster $Cw(i)$ that $\mathbf{x}_i$ belongs to, and $C_w(i)$ is determined by the centroid closest to $\mathbf{x}_i$.

4.3 Decision Tree Regression (DTR)

4.3.1 DTR Model Design

In k-MCR, to gain the regions in which to initially categorise a new data point, we relied on unsupervised learning to 'discover' the clusters within the data. Here we take a different approach, using Decision Trees we are going to recursively split the data (and hence the longitudes and latitude) into regions based on a specific criteria. The benefit of this is that we can choose these decisions (and hence splits) to maximize a specific criterion.

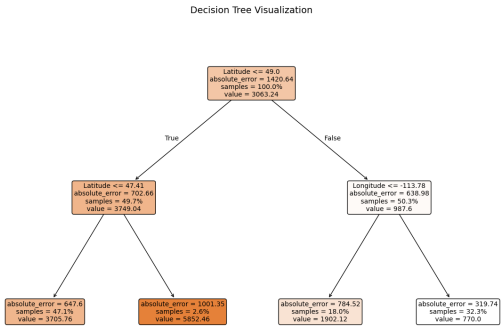

Figure 11: Simple example of a decision tree splitting the data based on latitude and longitude to predict TVD, with darker shades of orange indicating higher predicted values. The algorithm minimizes mean absolute error at each split.

Figure 11 shows a shallow decision tree (depth = 2) that splits North America into regions using latitude and longitude, minimizing mean absolute error (*absolute_error*) at each node. The tree first splits on latitude, then on latitude and longitude, creating four regions with distinct average TVD values.

The algorithm works as follows: the decision tree splits the data by minimizing the mean absolute error (MAE) at each node. That is, for a node with a set of target values $y = \{y_1, y_2, \ldots, y_n\}$, the MAE is defined as the average absolute deviation from the mean prediction $\bar{y}$:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}| \quad \text{where} \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

At each split, the tree evaluates a feature (latitude or longitude) and a threshold (e.g., latitude ≤ 49.0) to partition the data into two child nodes: a left node with values $y_L = \{y_i \mid x_i \leq \text{threshold}\}$ and a right node with values $y_R = \{y_i \mid x_i > \text{threshold}\}$. The tree selects the split that minimizes the weighted sum of the MAEs of the child nodes:

$$\text{Cost} = \frac{n_L}{n} \cdot \text{MAE}_L + \frac{n_R}{n} \cdot \text{MAE}_R$$

where n_L and n_R are the number of samples in the left and right nodes, respectively, $n = n_L + n_R$, and MAE_L and MAE_R are the mean absolute errors of the left and right nodes, computed as above. This process repeats recursively until a stopping condition (e.g., maximum depth) is met.

Once the terminal nodes (leaves) are determined, the model predicts the TVD of an out-of-sample data point by mapping its geographical position to the corresponding region and calculating the **mean** TVD of all sample points within that region.

4.3.2 DTR Model Justification and Parameters

Figure 12 effectively illustrates how the decision tree partitions North America into regions based on latitude and longitude splits, with each region colored by the average Total Vertical Depth (TVD) in dark blue shades. The density of splits—represented by the number of horizontal and vertical lines, corresponds to the concentration of wells, notably in southern Canada near the US border, where a dense cluster of shallow wells is evident. This density aligns with the well distribution shown in the earlier heatmap (Figure 7). Additionally, the symmetry between Figure 12 and Figure 8 highlights consistent patterns in TVD distribution across the region, with darker blue areas indicating shallower depths.

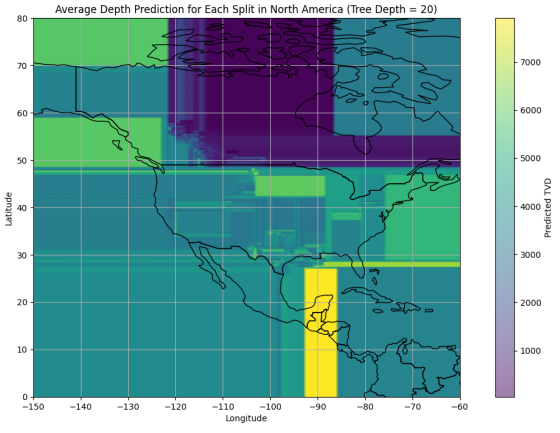

Figure 12: 2D Plot of Average TVD per Decision Tree Region (leaf or terminal node) in from a small sample in North America. Regions are colored by average TVD.

Aside from the error minimizing criterion which we will keep consistent with the other models, there are two other parameters that we will aim to optimize for this;

1. **Tree Depth (TD)** : The limit on how many splits to make and hence how many regions the map will be split into. Each layer doubles the number of regions so computationally this can be very expensive which we will take into consideration.
2. **Minimum Sample Split (MSS)** : The minimum number of samples required in each child node for a split to occur. For example, if `MSS = 50` and a node cannot be split such that both child nodes have at least 50 samples, the node becomes a terminal (leaf) node.

To determine the optimal value, we employ a brute-force **grid search** approach due to the empirical nature and large volume of the data.

4.4 Network Theoretic Regression (NTR)

In this section, we present an alternative to the three previously discussed machine learning models by introducing a novel approach grounded in graph theory. We propose a new theoretical framework, the **TVD Centrality Measure**.

4.4.1 NTR Model Design

To initialize the model, we construct a spatial network of oil wells, represented as a graph $G = (V, E)$, where each well is a node in V. The edges E are defined by connecting each well to its k-nearest neighbors based on geographical coordinates, using the Euclidean distance (L2 norm), consistent with the approach in k-NNR. While this method does not guarantee a fully connected graph, it ensures that every node belongs to a connected component, enabling regression analysis. Intuitively, these smaller connected components align with our prior clustering analysis, as we do not necessarily want or need these clusters to influence each other's final regression values.

TVD Centrality Measure

The True Vertical Depth (TVD) Centrality Measure introduced in this paper quantifies the centrality of a well's TVD relative to its neighboring wells. For each well i, the TVD Centrality c_i is defined as follows:

$$c_i = \exp\left(-\frac{|y_i - \overline{y}_{j\in N_k(i)}|}{\sigma^2_{N_k(i)} \cdot \alpha + \epsilon}\right)$$

Where:

- $N_k(i)$ the set of k nearest neighbours to well i
- y_i is the true vertical depth (TVD) of well i
- $\overline{y}_{j\in N_k(i)}$ is the mean true vertical depth (TVD) of all neighbors of well i
- $\sigma^2_{N_k(i)}$ is the variance of TVD values among the neighbors
- α is a sensitivity parameter, adjustable per dataset, to be optimized later to ensure significant TVD Centrality variance within local areas.

The exponential function ensures a smooth transition between high and low centrality values, while using the mean of neighbors instead of individual differences mitigates the impact of outliers. This measure naturally penalizes wells that deviate significantly from their local environment, and the variance term $\sigma^2_{N_k(i)}$ normalizes the measure to account for varying scales of TVD variation across different regions.

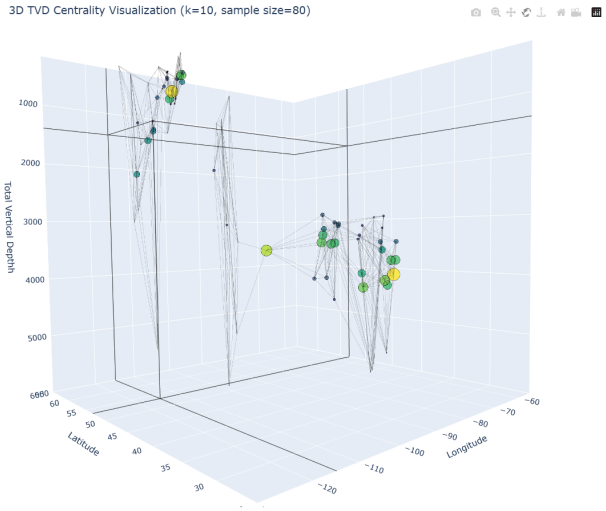

Figure 13: 3d Plot of longitude, latidude and TVD of a small sample of wells in North America. The size of the marker is proportional to the TVD centrality of the node. We can observe the more extreme (shallow and deeper relative) nodes with a **low** TVC Centrality and wells in the centre of their clusters with **high** TVD centrality.

Figure 13 presents a 3D plot that visualizes the TVD Centrality scores and their relationship to local TVD values. Within each local cluster, larger markers are positioned near the center of the TVD range, indicating higher centrality for wells with TVD values close to the mean of their neighbors. Conversely, wells with extreme TVD values, which deviate significantly from their neighbors' mean, exhibit smaller markers, reflecting their lower centrality scores.

Centrality to Regression

When predicting TVD for a new well location, we use these centrality values and inverse distance in our prediction as follows:

$$\hat{f}(\mathbf{x}_i) = \frac{\sum_{j\in N_k(i)} c_j \cdot w_j \cdot y_j}{\sum_{j\in N_k(i)} c_j \cdot w_j} = \hat{y}_i$$

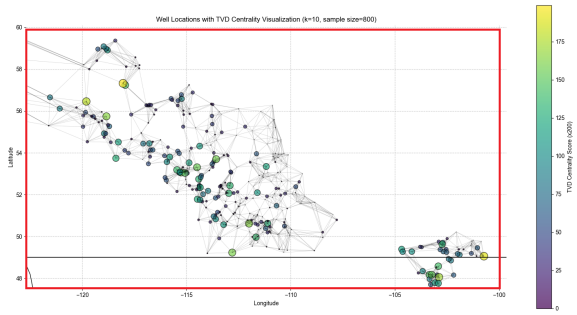

Figure 14: Graph showing TVD Centrality across a sample of wells in US and Canada. The edges connect k nearest neighbors, the colour / size represent the degree TVD centrality. This is a zoomed in area of figure 15 (the red rectangle).

with

$$w_j = \frac{1}{d(\mathbf{x}_i, \mathbf{x}_j)}$$

Where:

- c_j is the TVD Centrality of neighbor j
- $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between well i and neighbor j (L2 Norm)
- y_j is the TVD of neighbor j
- $N_k(i)$ is the set of k nearest neighbors

This formulation integrates TVD centrality and inverse distance weighting, prioritizing wells with high TVD centrality (representative of their area) and those geographically closer to the prediction point. This weighted approach ensures that wells better representing their local area exert greater influence on the prediction, while outlier wells have reduced impact, allowing the prediction to naturally adapt to local geological patterns.

The TVD Centrality Measure introduced in this study is inspired by classical network centrality metrics, such as degree centrality, but is tailored for geological applications. While degree centrality quantifies a node's importance through its number of connections, our measure evaluates a well's significance by assessing how representative a well's TVD with that of its neighbours. A high TVD Centrality indicates that a well effectively represents the TVD characteristics of its surrounding area, making it a strong candidate for greater weighting in the regression model to improve prediction accuracy.

4.4.2 NTR Model Justification and Parameters

In earlier approaches (k-NNR, k-MCR), we have looked into regressions based on local areas and what these geospatially neighbouring wells can do to help us predict characteristics, in new (test) data.

The biggest issue so far has been how to we **quantify** a neighbouring wells importance in the regression. We have looked into a pure mean in k-NNR as well as inverse distance relationships. While both of these appear to give good results in initial testing, we are suseptable to the influence of outliers in the calculation. Referencing figure 13 again, we can see that there are several very deep outliers on the plot. In a mean-based regression, these outliers will drag the mean value down disproportionately to how influence able we want it to be in the calculation. TVD Centrality attempts to address this by assigning these outliers very low centrality and hence in the regression calculation they will have a far dampened effect on the final value $\hat{y}_i$.

There are two parameters that we wish to optimize for this model:

1. **KNN** (k): The number of neighbors used to construct the initial graph, determining the connectivity of each well. This should directly affect the regression value and we can optimize this independently. Again, we use mean absolute error (MAE).

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{f}(\mathbf{x}_i|k)\right|$$

2. **Alpha** (α): The sensitivity parameter in the TVD Centrality formula, tuned to ensure sufficient variance in centrality scores, thereby creating a meaningful distinction between the most and least central wells.

$$\alpha^* = \arg\max_{\alpha} \text{Var}(\{c_i(\alpha|k) \mid i \in V\}) \quad (1)$$

where $c_i(\alpha|k)$ is the centrality score of well i with k-nearest neighbors (fixed k), and V is a subset of wells we use to evaluate the parameter. The variance is computed over the centrality scores for all wells, as detailed in Section 4.4.1.

5 Model Training, Validation and Parameter Selection

In this section we will discuss the methodology behind the training / validation process that was used for each of the models and select the optimal parameters that have been introduced in the justification and parameter sections (4.1.2, 4.2.2, 4.3.2, 4.4.2).

For all four models the same methodology for parameter selection, this being as follows:

1. The range of parameters to be tested was first identified. For example, for k-MCR, the values for k could be:

$$k \in \{5, 10, 15, 20, 25\}$$

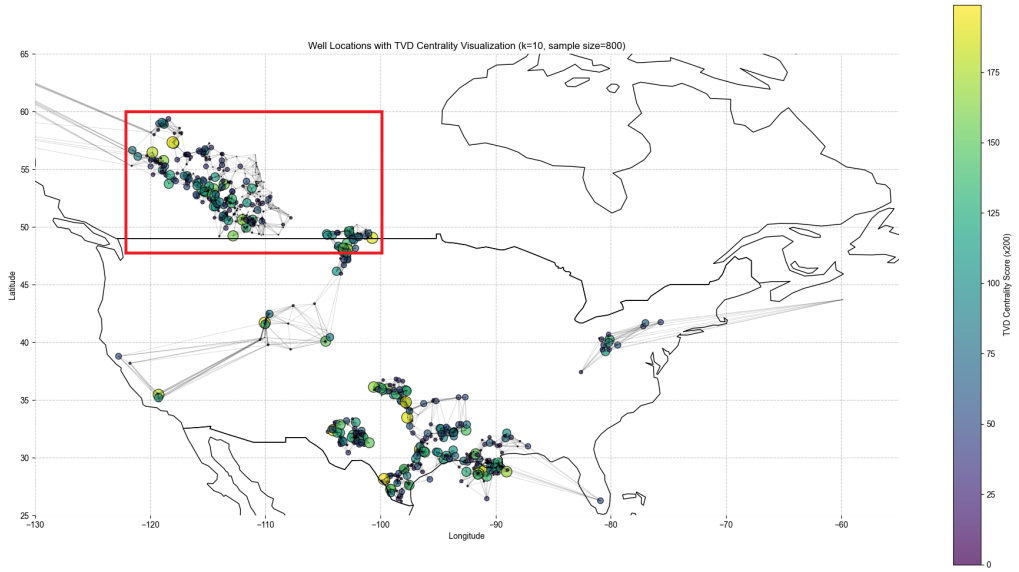

Figure 15: TVD Centrality visualization across North American oil wells. Node size and color intensity represent the TVD centrality measure. Edges connect each well to its k nearest geographical neighbors, illustrating the spatial network structure used for prediction. Figure 14 has more detail.

2. A random sample of 25,000 wells was drawn from the full merged dataset and split into an 80%/20% train/validation set.
3. This step was repeated 10 times, resampling from the merged dataset each time.
4. Based on the selected cost metric (e.g., lowest MAE), the average results were compared, and the optimal parameter value was selected.

5.1 k-NNR Validation

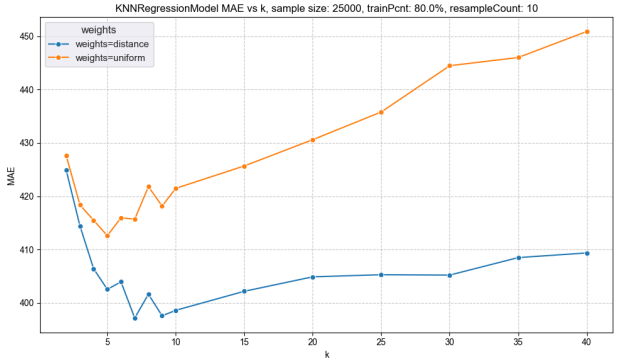

Figure 16: MAE for k-NNR validation, comparing over a range of k values for uniform weighting (orange) and inverse distance (blue) weighting.

In Figure 16, we compare the Mean Absolute Error (MAE) of $\hat{f}_{\mathrm{U}}(\mathbf{x}_i|k)$ (orange) and $\hat{f}_{\mathrm{ID}}(\mathbf{x}_i|k)$ (blue) across different values of k. The results clearly demonstrate that the inverse distance-based regression yields a more accurate model.

The upward trend of both lines as k increases suggests overgeneralization in the regions surrounding each well. As more neighbors are included in the calculation, the model incorporates excessive information from distant points, introducing bias and reducing its ability to capture local patterns. This behavior is indicative of underfitting, where the model becomes overly simplistic and fails to adequately represent the underlying data structure. The blue line, representing the inverse distance relation, is less affected by this issue due to its heavier weighting of closer values. This can be observed by the smaller gradient of the blue line as k becomes greater than 10 compared the orange line.

Regarding the optimal value of k, the MAE is minimized when k ranges between 7 and 10. Based on this analysis, we select $k = 7$ and the inverse distance method as our preferred model. Consequently, the regression is defined as $\hat{y}_i = \hat{f}_{\mathrm{ID}}(\mathbf{x}_i|k = 7)$.

5.2 k-MCR Validation

In figure 17, we can observe the two models, Linear Regrssion (blue) and average based (orange), over a wide range of clusters $k \in [5, 200]$. Both curves exhibit the same shape, with MAE descrasing sharply initially as k increases, but show asymptotic behaviour as becomes large. The regression based model is has lower MAE for all values of k aside from $k = 200$. From this we can infer that the linear regression implementation gives consistently lower error. In terms of the number of clusters k, we will choose $k = 100$ as our hyperparameter as there appears to be little benefit beyond this. Our final equation will be $\hat{y}_i = \hat{f}_{LR}(\mathbf{x_i}|k = 100)$.

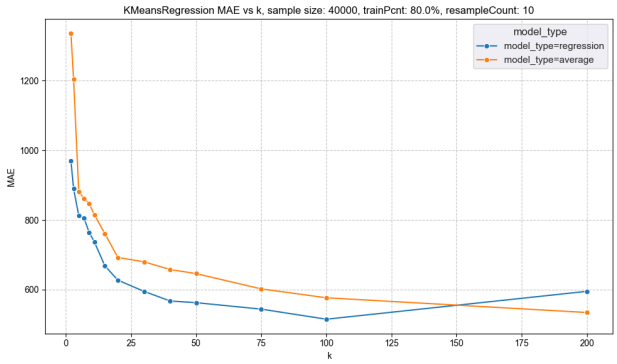

Figure 17: MAE for k-MCR validation comparing $k \in [5, 200]$ for the linear regression (blue) and uniform average (orange) approaches.

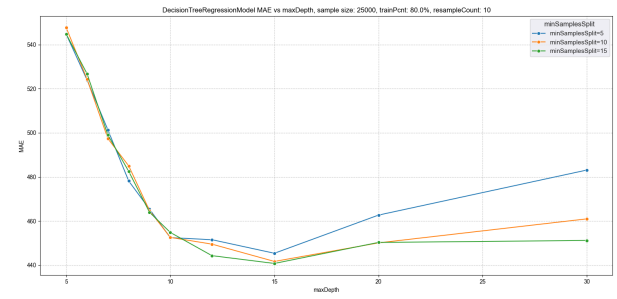

Figure 18: MAE for DTR validation we compare three Minimum Sample Splits (MSS) for varying Tree Depth (TD), $TD \in [5, 30]$.

5.3 DTR Validation

Unlike in 5.1 and 5.2, where we directly adjust the number of clusters or neighbors, Decision Tree Regression primarily involves varying the tree depth, as discussed in TD. Given the size of the datasets used in this report and the binary structure of a decision tree, the number of regions can potentially double with each additional level, assuming every node splits fully. For example, a tree of depth 11 could have up to twice as many regions as one of depth 10. However, this exponential increase assumes maximal splitting, which may not always occur due to data-driven stopping criteria or pruning. As a result, we must exercise caution to prevent overfitting, where each well could be isolated in its own region, reducing the regression to merely predicting the target value (e.g., TVD) of the nearest well.

This overfitting behavior is evident in Figure 18. A distinct minimum in MAE occurs at $TD = 15$, with higher values leading to increasing MAE. Regarding the Minimum Sample Split (MSS), we observe that for lower values of TD, the MAE remains largely indistinguishable across MSS values. However, as tree depth increases, $MSS = 15$ consistently yields the lowest error. Overall, MSS appears to have a far smaller impact on the error than TD. Based on this analysis, we select $TD = 15$ and $MSS = 15$, defining the regression as $\hat{y}_i = \hat{f}(\mathbf{x}_i | TD = 15, MSS = 15)$.

5.4 NTR Validation

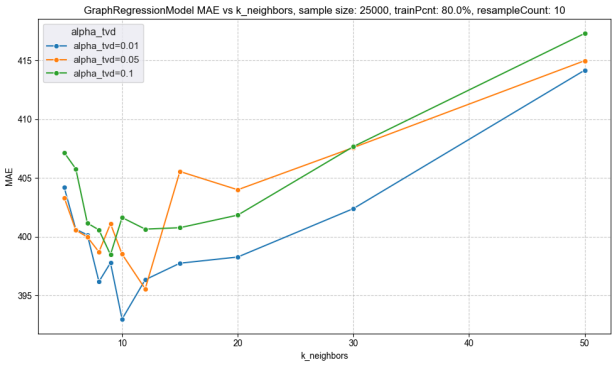

Figure 19: MAE for NTR validation, comparing $k \in [5, 20]$ and three different values of α. MAE appears to exhibit fairly large variance; however, the y-axis scale is narrow, spanning only an approximate 5% change.

At first glance, Figure 19 appears to exhibit significant variance, potentially complicating parameter selection. However, a closer examination of the y-axis scale reveals minimal variance, with the MAE spanning only approximately a 5% range. Each line corresponds to a different value of our newly introduced hyperparameter, α. All three lines display a minimum in the region $k \in [8, 12]$, with the blue line ($\alpha = 0.01$) demonstrating the best overall performance. These validation results show clear similarities to those in 5.1 (k-NNR validation), as expected, given the modification of the k-Nearest Neighbors selection in each validation. As k increases, bias emerges, causing the model to underfit and become overly generalized due to the inclusion of a larger catchment of wells. For the final NTR model, we select $\hat{y}_i = \hat{f}(\mathbf{x}_i | k = 10, \alpha = 0.01)$.

6 Test Results and Discussion

6.1 Performance Metrics

We now introduce how we will quantify the 'goodness' of out models on the test set.

We have spoken extensively about Mean Absolute Error (MAE) in this report for validation. Here we introduce all the metrics we will use to evaluate and compare the performance of the models on the test set with the final hyperameters. Each of these provides a different perspective on how well the models perform:

- **Mean Squared Error (MSE)**: Measures the average squared difference between the predicted and actual values. It penalizes larger errors more severely.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **R^2**: Indicates the proportion of variance in the dependent variable that is predictable from the independent variables.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the actual values.

- **Mean Squared Logarithmic Error (MSLE)**: Similar to MSE but uses logarithmic values. It is useful when target values span multiple orders of magnitude.

$$\text{MSLE} = \frac{1}{n}\sum_{i=1}^{n}\left(\log(1 + y_i) - \log(1 + \hat{y}_i)\right)^2$$

- **Mean Absolute Percentage Error (MAPE)**: Expresses the average error as a percentage of the actual value. It is scale-independent but unstable near zero values.

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

- **Mean Absolute Error (MAE)**: As presented earlier, this measures the average absolute difference between predicted and actual values. It is less sensitive to outliers than MSE.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **Kullback–Leibler Divergence (KLD)**: Quantifies how much one probability distribution diverges from a second, expected probability distribution.

$$D_{\text{KL}}(P\|Q) = \sum_{i=1}^{n} P(i)\log\left(\frac{P(i)}{Q(i)}\right)$$

where $P(i)$ is the true probability distribution of the empirical data $Q(i)$ is the predicted probability distribution by the model. The smaller this is the closer the distribution.

From these metrics, we can conclude that models with lower MSE, MSLE, MAE, MAPE, and KLD and higher R^2, are generally more desirable. They are scaled and transformed this way in figure 20 for visual analysis.

6.2 Results

To test and compare the four models, as well as the baseline, the Country Average (CA) we adopted the following approach:

1. For each model, we use the **full merged dataset** and randomize the order.
2. An 80%/20% train-test split was performed on the full merged dataset.
3. This process was repeated 10 times per model, and the results were averaged.

The dataset comprises a total of 1,028,396 wells. This split yields 822,717 wells for training and 205,679 wells for testing per test.

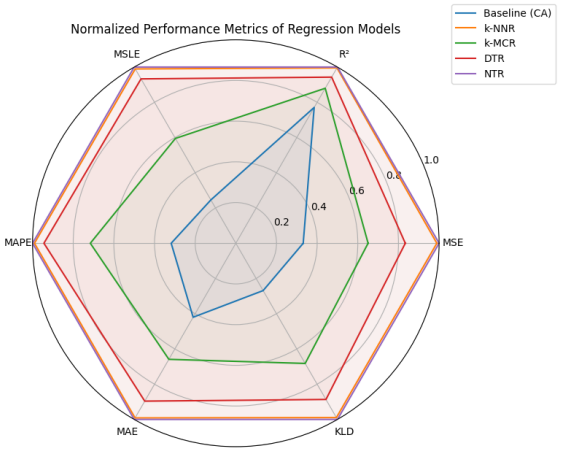

Figure 20: Normalised spider diagram of the metrics in table 1, with all metrics scaled to be $\in [0, 1]$ with 1 being the best for that particular metric. NTR (purple) outperforming all other models in all 6 tests, narrowly improving on k-NNR (orange).

From table 1 and the metrics spider diagram, figure 20, we can see that all four tested models outperformed the benchmark (Country Average). This was somewhat expected as all 4 models take an approach to find local wells in various ways and then perform some sort of regression on them. The country average, despite being the worst performing in all 6 key metrics, has an R^2 of 0.6924 which given the variance of the full dataset's TVD (2,580,468), it is quite remarkable.

6.2.1 First vs. Second

In terms of selecting the 'best' model, is is clear that is is a two horse race between k-NNR and NTR. Their performance metrics are the best and closest of all the models, this is evident from the spider diagram in figure 20 where they appear to be indistinguishable on the scale that is presented. The differences are clarified in table 1 where we can see that in fact NTR has beats k-NNR on all metrics, even if it close.

Table 1: **Performance Metrics** of regression Models on the test set

	Baseline (CA)	k-NNR	k-MCR	DTR	NTR
Metric					
MSE	794,100	272,108	540,633	395,293	263,236
R^2	0.6924	0.8941	0.7902	0.8470	0.8984
MSLE	0.2278	0.0584	0.1483	0.0713	0.0560
MAPE	45.850	15.008	27.610	17.146	14.583
MAE	689.65	295.64	524.95	361.29	289.67
KLD	0.0856	0.0239	0.0502	0.0327	0.0230

Table 2: **Average training runtime** (in seconds) for each model, computed over 10 repetitions, using the specified sample sizes.

Sample Size	Baseline (CA)	**k-NNR**	**k-MCR**	**DTR**	**NTR**
500	0.0032	0.0007	0.747	0.0275	0.430
1000	0.0064	0.0061	0.700	0.0821	0.887
2000	0.0064	0.0067	0.745	0.285	1.73
5000	0.0032	0.0126	0.761	1.34	4.34
10000	0.0063	0.0165	0.886	5.32	8.62
20000	0.0096	0.0327	1.065	20.96	16.97
50000	0.0096	0.104	1.956	131.56	43.20

To evaluate the statistical significance of the observed differences in performance metrics, we conducted a hypothesis test using the **Wilcoxon Signed-Rank Test**. This analysis utilized the absolute errors from the complete test set, comprising over 200,000 samples, for each model. The null hypothesis (H_0) posits that there is no difference in the median absolute errors between the models, while the alternative hypothesis (H_1) asserts that one model's errors are significantly lower than the other's. The significance test was performed with a threshold of $p = 0.001$ (0.1%). The Wilcoxon test yielded a p-value of 0.000×10^{-30}, effectively zero within Python's floating-point precision limits, due to the exceptionally large test set size. Consequently, we reject H_0 at the $p = 0.001$ level, indicating a statistically significant difference in model performance. For context, when the same test was applied to a subsample of 20,000 wells, the p-value was 5.619×10^{-23}, still far below the threshold, reinforcing the robustness of this conclusion despite the reduced sample size.

6.2.2 Computational Efficiency vs. Accuracy

When comparing computationally intensive models, it is essential to evaluate both the training cost and the feasibility of applying these models to large datasets.

Table 2 presents the average training runtimes for each model across a range of sample sizes, from 500 to 50,000, based on 10 repetitions. These results are visualized on a log-log scale in Figure 21. Each model employs distinct algorithms that scale differently with test set size and exhibit significant variations in runtime depending on the chosen hyperparameters. In both the table and figure, only the final, accuracy-optimized parameters are evaluated, which are not necessarily optimized for computational efficiency. For instance, the DTR model was configured with a maximum tree depth of 15, potentially yielding up to 2^{15} terminal nodes (and thus regions). However, this maximum is typically reached only with sufficiently large datasets, as smaller datasets often require fewer splits, leaving many terminal nodes unrefined. This was evident when training DTR over the full merged dataset when each run took approximately **three hours** where the other models were between 1 and 15 minutes.

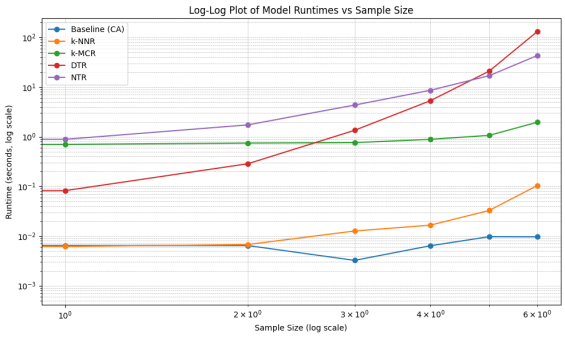

Figure 21: Log Log plot of table 2, comparing training times of all models for different sample sizes.

Overall, k-NNR offers the best balance of performance and computational efficiency. Although its accuracy ranks second to NTR, it is significantly faster, making it the preferred choice when speed is critical or when handling very large datasets. Conversely, DTR exhibits the poorest efficiency, as it struggles to manage the largest datasets effectively while maintaining sufficient granularity for regression tasks.

6.2.3 A *'Digression on the Regression'*

While identifying the top-performing models and understanding the factors contributing to their accuracy is crucial, it is also important to examine the underperforming models. In this section, we analyze the disparity in accuracy between the models to determine what the less effective models (DTR, k-MCR, and Baseline (CA)) inherently lack compared to the best performers (NTR and k-NNR), and how these deficiencies impact their predictive performance.

Here we will mainly focus on k-MCR, however the regression part of DTR has similarities and and also applies.

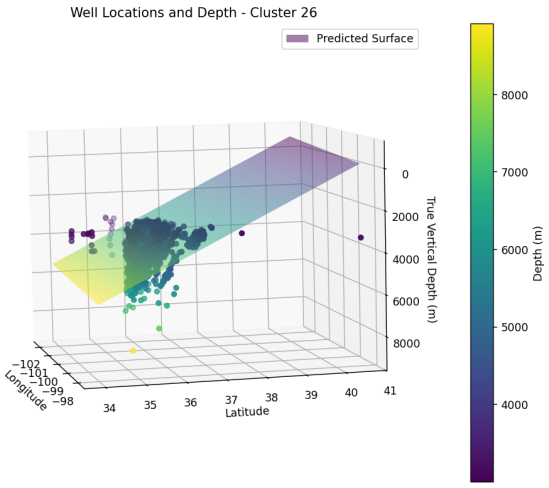

Figure 22: Example of a linear regression over a cluster, approximating a a cluster with a plane.

Figure 22 illustrates a linear regression plane fitted to a cluster of wells identified through a k-Means Clustering step in the k-MCR model. Within the primary cluster on the left, the linear regression plane effectively captures the underlying pattern, closely aligning with the general shape of the cluster. However, on the right side of the plot, a well with a depth similar to the main cluster is poorly approximated by the plane, predicting a depth significantly shallower than the actual value. During testing, when cluster diameters were sufficiently large, the k-MCR model frequently predicted negative depths for some wells—implying locations above the surface!. This issue arises because the linear approximations become less reliable as cluster diameters increase, leading to larger errors for wells that are physically distant from the main cluster body. This behavior is evidenced in the earlier validation, as shown in Figure 17. The figure demonstrates that increasing the number of clusters k results in smaller cluster diameters, which in turn reduces the impact of these linear approximations and mitigates the large errors associated with distant wells.

In addition to the tendency of the k-MCR model to significantly over- or under-predict TVD for wells distant from the cluster center, figure 23 highlights the severity of underestimation when it occurs. Specifically, the residual distribution $(\hat{y}_i - y_i)$ not only exhibits a pronounced left skew, as noted earlier, but also reveals that underestimates can be particularly extreme, with some residuals revealing a 5000m understimate. This indicates that when k-MCR underestimates the depth of a well, the error can be substantial, further highlighting the limitations of the model in capturing the full range of TVD values within a cluster. The residuals, defined as $\hat{y}_i - y_i$, exhibit a pronounced left skew, indicating that k-MCR frequently underestimates the depths of deeper wells. This limitation stems from the model's reliance on linear regression within clusters, which struggles to accurately approximate the deepest wells. This issue is further illustrated in Figure 22, where the deepest wells deviate significantly more from the linear regression plane than the shallowest ones. Approaching the data with a single linear plane per cluster is inherently limited, as it cannot effectively capture the variability across all regions within a cluster. In contrast, the NTR and k-NNR models mitigate this issue by avoiding linear approximations and instead leveraging local patterns—k-NNR uses the closest neighbors for prediction, while NTR employs a non-linear approach. This difference is reflected in their superior performance, as evidenced by the overall MAE values in Table 1.

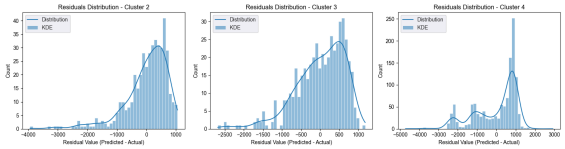

Figure 23: Histograms of residual errors (predicted minus actual Total Vertical Depth) for clusters three clusters from k-MCR with $k = 20$, followed by linear regression, applied to a sample of 100,000 wells in North America. The distributions exhibit a pronounced left skew and heavy left tails, indicating a tendency to underestimate well depths.

Although both DTR and k-MCR employ regression within partitioned regions, DTR achieves a notably lower MAE compared to k-MCR (361.29 vs. 356.37, as shown in Table 1). This difference can be attributed to the distinct algorithms used for region or cluster formation. The k-MCR model relies on an unsupervised k-Means Clustering approach, iteratively grouping wells based solely on their spatial proximity. In contrast, DTR constructs its regions by recursively splitting the feature space to minimize the MAE between the resulting partitions. This optimization process in DTR deliberately isolates wells that contribute to larger errors into separate regions, thereby reducing the overall prediction error and improving accuracy compared to the purely distance-based clustering of k-MCR.

7 Conclusion and Future Direction

7.1 Conclusion

7.2 Future Direction

7.2.1 Directional Data

Add a graph of the directional data here

7.2.2 Forecasting Supply / Demand

7.2.3 Time series Variations

7.2.4 Using empirical distributions to improve the NTR

7.2.5 NTR Beyond Oil Wells

References

Deloitte (2025). 2025 oil and gas industry outlook. https://www2.deloitte.com/us/en/insights/industry/oil-and-gas/oil-and-gas-industry-outlook.html. Accessed: March 31, 2025.

International Energy Agency (2025). Oil market report - march 2025. https://www.iea.org/reports/oil-market-report-march-2025. Accessed: March 31, 2025.

Wikipedia Contributors (2023a). 1973 oil crisis. https://en.wikipedia.org/wiki/1973_oil_crisis. Accessed: April 2, 2025.

Wikipedia Contributors (2023b). Gulf war. https://en.wikipedia.org/wiki/Gulf_War. Accessed: April 2, 2025.

Wikipedia Contributors (2023c). Iran–iraq war. https://en.wikipedia.org/wiki/IranIraq_War. Accessed: April 2, 2025.

Wikipedia Contributors (2023d). Oil well. https://en.wikipedia.org/wiki/Oil_well. Accessed: April 2, 2025.

Wikipedia Contributors (2023e). Opec. https://en.wikipedia.org/wiki/OPEC. Accessed: April 2, 2025.