DATA DRIVEN MODELS TO PREDICT PORE PRESSURE USING DRILLING AND PETRO PHYSICAL DATA

Utkarsh Raja

210107091

Submission Date: April 25, 2024

Final Project submission

Course Name : Applications of Al and ML in chemical engineering

Course Code: CL653

Contents

1 Executive Summary

This project aims to address the critical issue of determining the mud weight window (MW) in drilling oil and gas wells, which is crucial for ensuring the stability and safety of the borehole. The proposed solution involves the development and evaluation of novel algorithms for predicting pore pressure (PP) using machine learning (ML) and hybrid machine learning (HML) techniques based on log and drilling data. The problem statement revolves around accurately predicting PP, a key parameter in drilling operations, using input features such as rate of penetration (ROP), deep resistivity (ILD), density (RHOB), photoelectric index (PEF), corrected gamma ray (CGR), compression-wave velocity (Vp), weight on bit (WOB), shear-wave velocity (Vs), and pore compressibility (Cp). These features are analysed to identify the most influential ones, and six algorithms are developed for PP prediction: K-nearest neighbour (KNN), weighted K-Nearest Neighbour (WKKNN), distance weighted KNN (DWKNN), and their hybrid forms with particle swarm optimization (PSO) (KNN-PSO, WKNN-PSO, and DWKNN-PSO).

The project methodology involves training the algorithms using a dataset. The dataset is split into training, testing, and validation sets, and feature selection techniques are applied to identify the most relevant input features for PP prediction. The AI models are then developed using the selected features.

The main novelty of the study lies in the application of these novel algorithms for PP prediction, which have not been previously explored for this purpose. Additionally, the use of PSO optimization enhances the accuracy of the models by optimizing weights and K values.

The expected outcomes of the project include the identification of the most accurate algorithm for PP prediction, validation of its performance on unseen data from, and the generalizability of the developed models for application in other fields. The results demonstrate the effectiveness of the proposed algorithms in accurately predicting PP, thereby contributing to safer and more efficient drilling operations.

2 Introduction

Background:

In Chemical Engineering, accurate prediction and control of parameters during drilling operations are essential for ensuring the safety and efficiency of oil and gas extraction processes. The mud weight window (MW) determination is particularly crucial as it helps prevent issues such as formation damage, wellbore instability, and blowouts. Chemical engineers play a vital role in developing algorithms and models to predict pore pressure (PP), which is a key factor in determining the MW.

Problem Statement:

Data driven models to predict pore pressure using drilling and petrophysical data

The project aims to address the challenge of accurately predicting pore pressure (PP) during drilling operations in the oil and gas industry. PP prediction is critical for determining the appropriate mud weight window (MW), which ensures the stability of the wellbore and prevents costly drilling complications. The specific problem involves developing and evaluating novel algorithms that can effectively predict PP based on input parameters such as rate of penetration (ROP), deep resistivity (ILD), density (RHOB), photoelectric index (PEF), corrected gamma ray (CGR), compression-wave velocity (Vp), weight on bit (WOB), shear-wave velocity (Vs), and pore compressibility (Cp).

References:

- Data driven models to predict pore pressure using drilling and petrophysical data

Objectives:

1. Develop machine learning (ML) algorithms, including K-nearest neighbor (KNN), weighted K-Nearest Neighbor (WKKNN), and distance weighted KNN (DWKNN), for PP prediction based on input parameters.

2. Explore hybrid machine learning (HML) techniques by combining ML algorithms with particle swarm optimization (PSO) to enhance PP prediction accuracy.

3. Evaluate the performance of developed algorithms using a dataset.

4. Identify the most accurate algorithm for PP prediction and assess its performance on unseen data to test generalizability.

5. Validate the effectiveness of the developed models in accurately predicting PP, contributing to safer and more efficient drilling operations in the oil and gas industry.

3 Methodology

Data Source:

The dataset used in this project was collected from three wells (S1, S2, and S3) in the Tabnak field, an oil and gas reservoir in Iran since we can't avail the dataset so I have generated data randomly using AI tools making sure it is normally distributed. The dataset includes records of various drilling parameters such as rate of penetration (ROP), deep resistivity (ILD), density (RHOB), photoelectric index (PEF), corrected gamma ray (CGR), compression-wave velocity (Vp), weight on bit (WOB), shear-wave velocity (Vs), and pore compressibility (Cp).

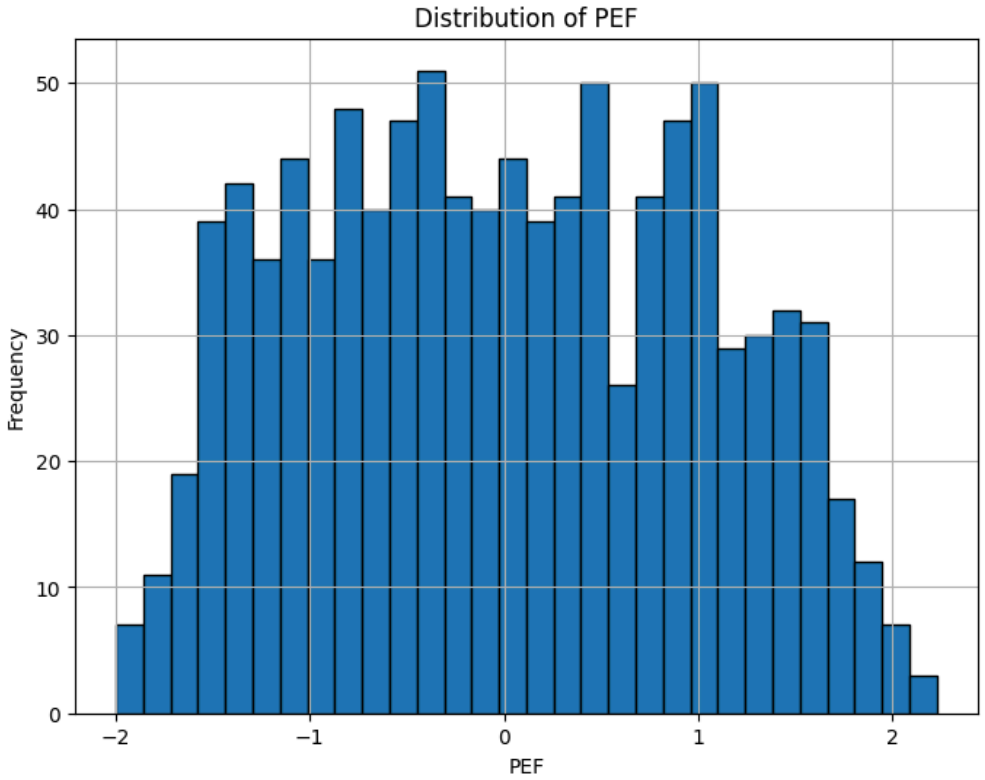
Data Preprocessing:

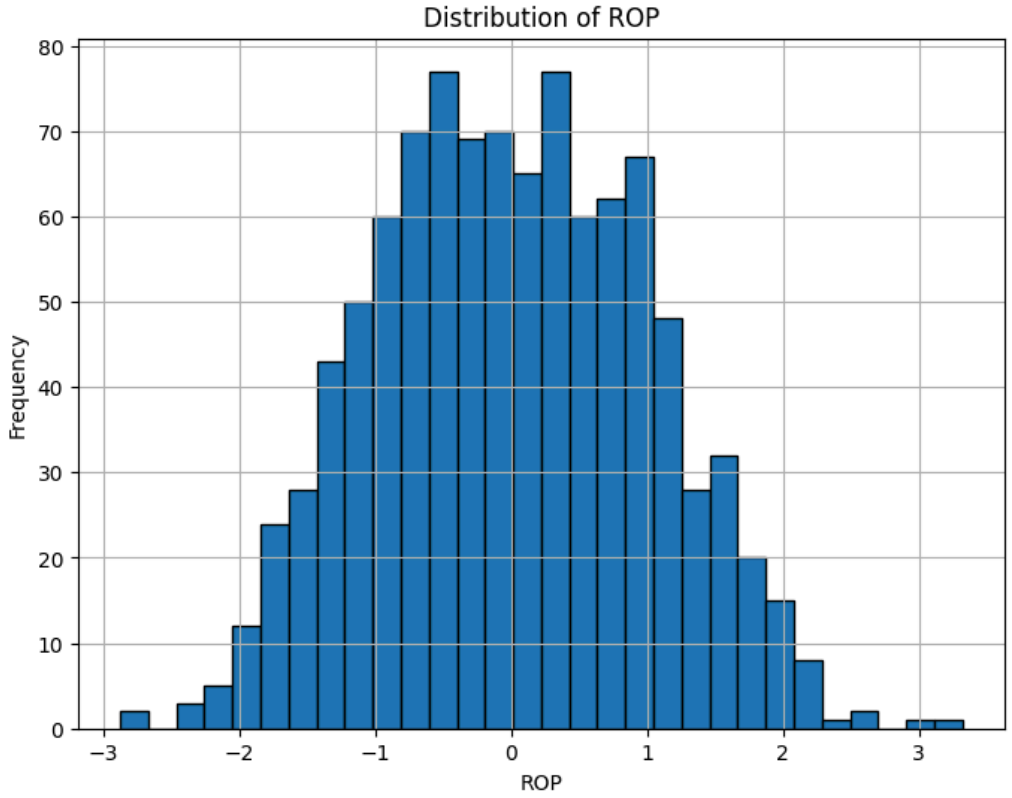
1. Data Cleaning: Removing any duplicate or irrelevant records, handling missing values, and addressing outliers.

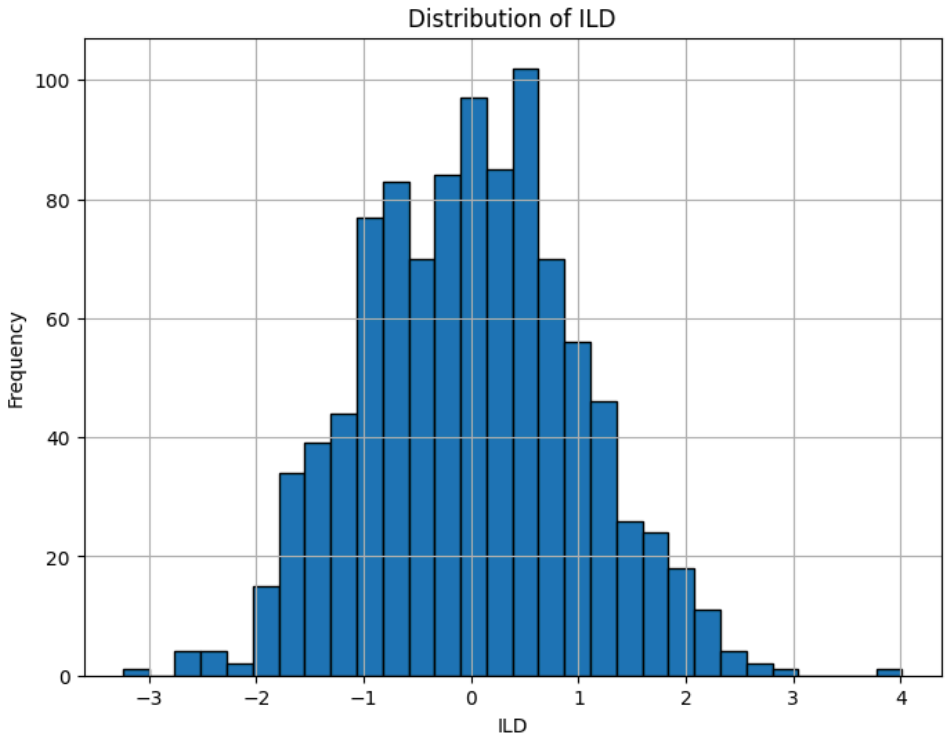
2. Feature Selection: Identifying the most influential input features using techniques such as correlation analysis or feature importance ranking.

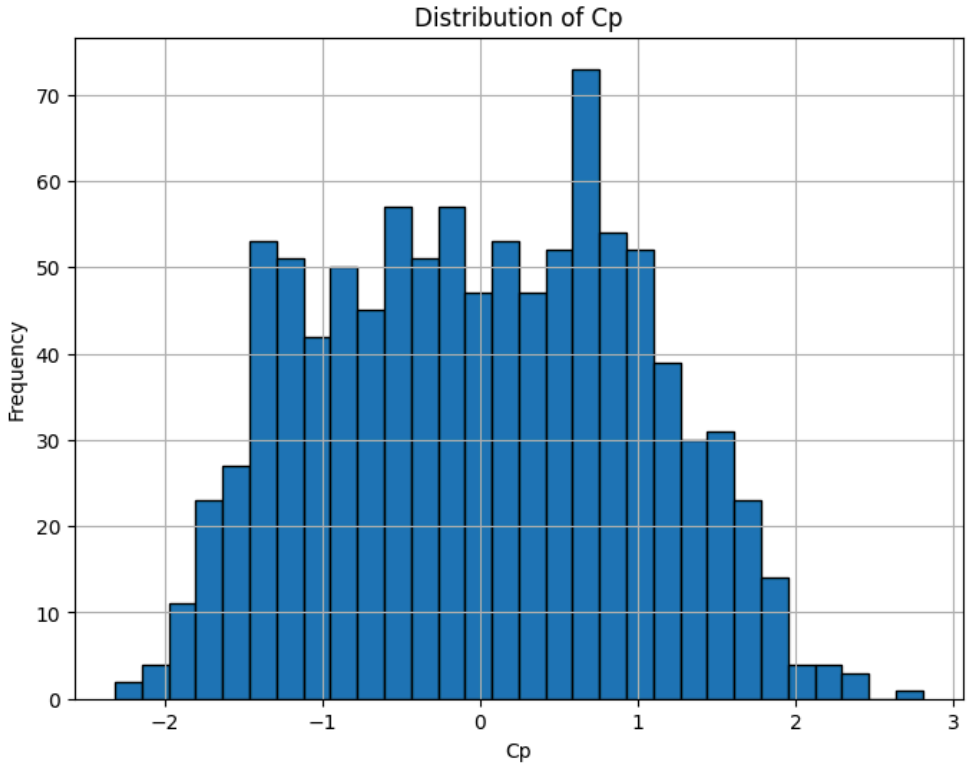
3. Data Scaling: Standardizing or normalizing the input features to ensure consistency and improve model performance.

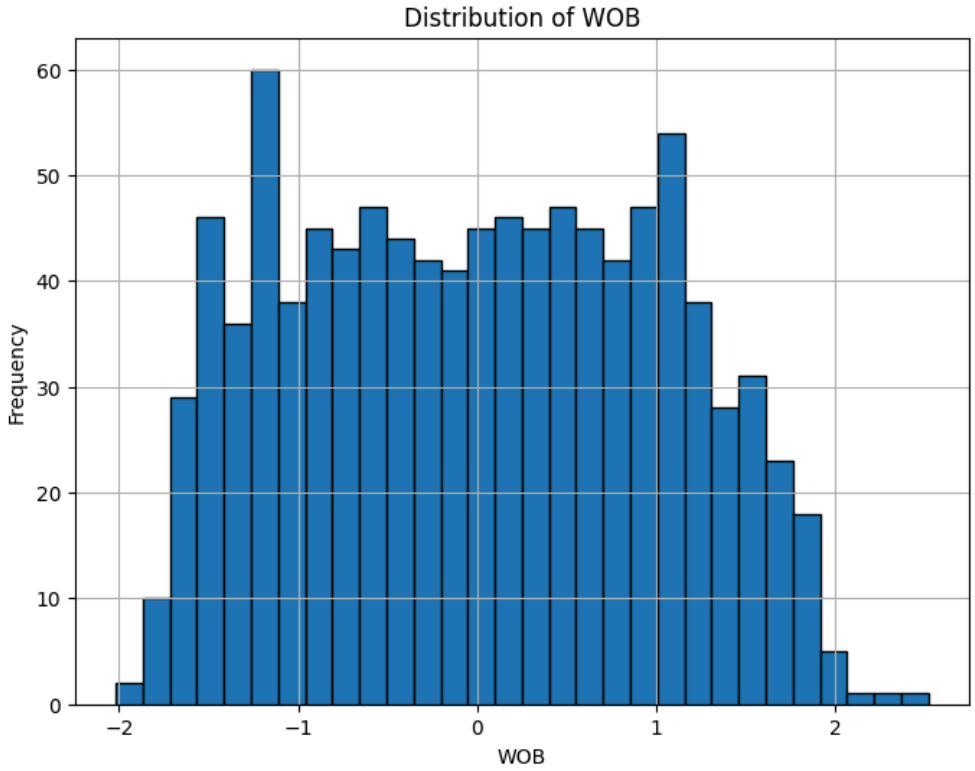
4. Data Splitting: Dividing the dataset into training, validation, and test sets for model development, evaluation, and validation.

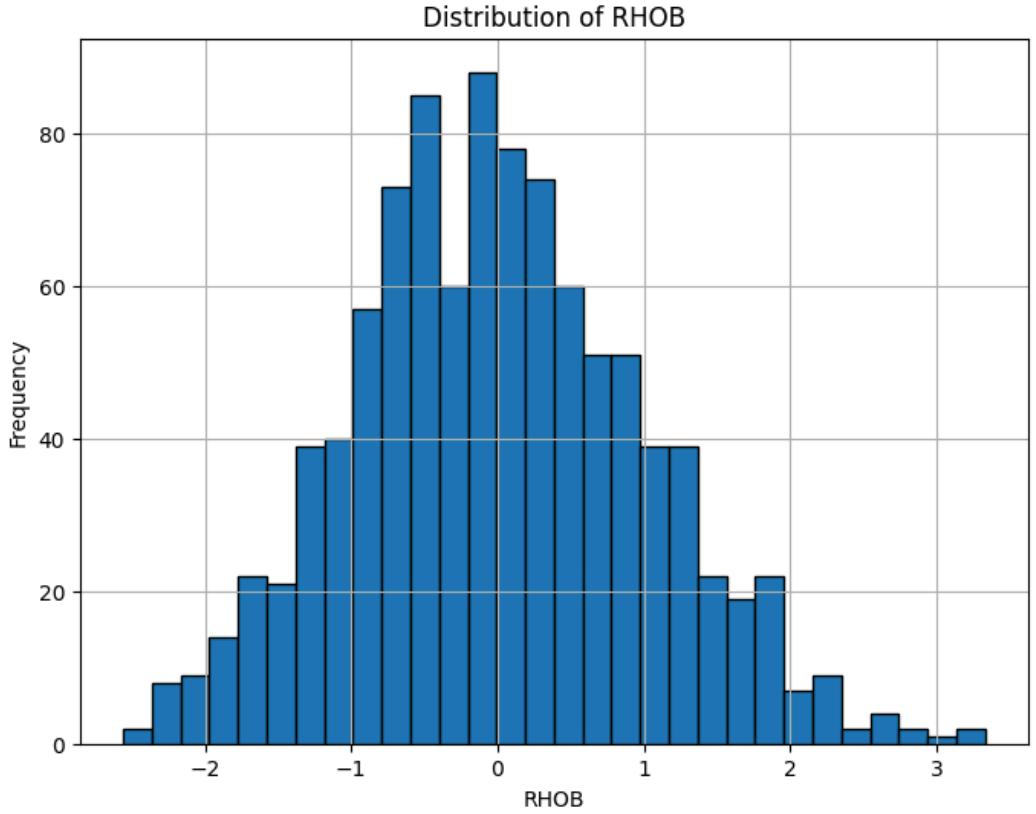

Distribution of PEF
Frequency
PEF
0
10
20
30
40
50
−2
−1
0
1
2

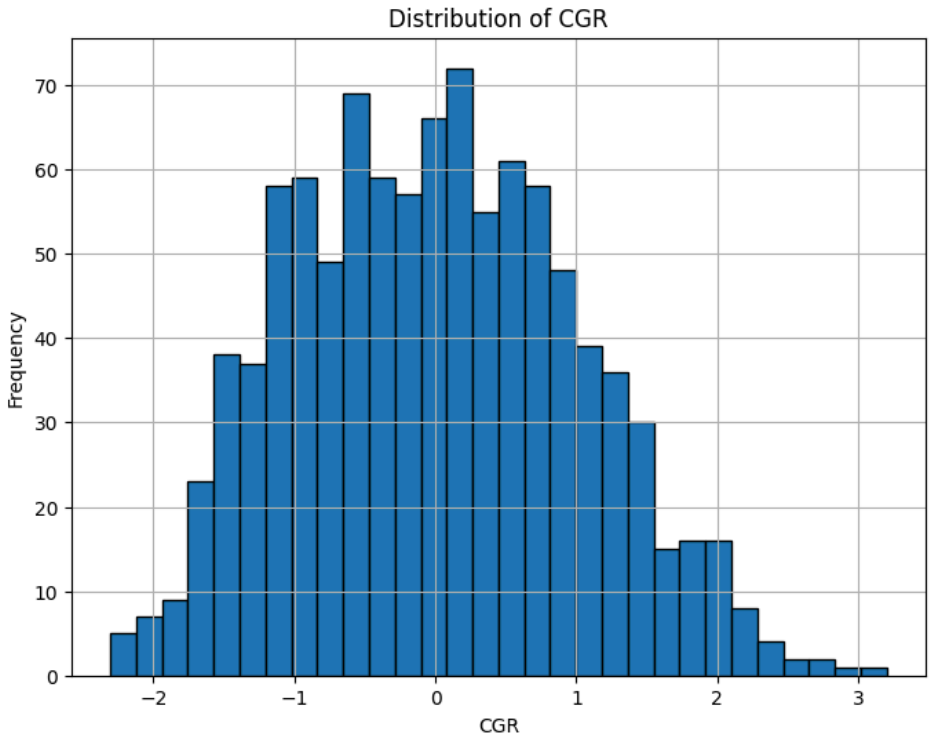

Distribution of ROP
Frequency
ROP
0
10
20
30
40
50
60
70
80
−3
−2
−1
0
1
2
3

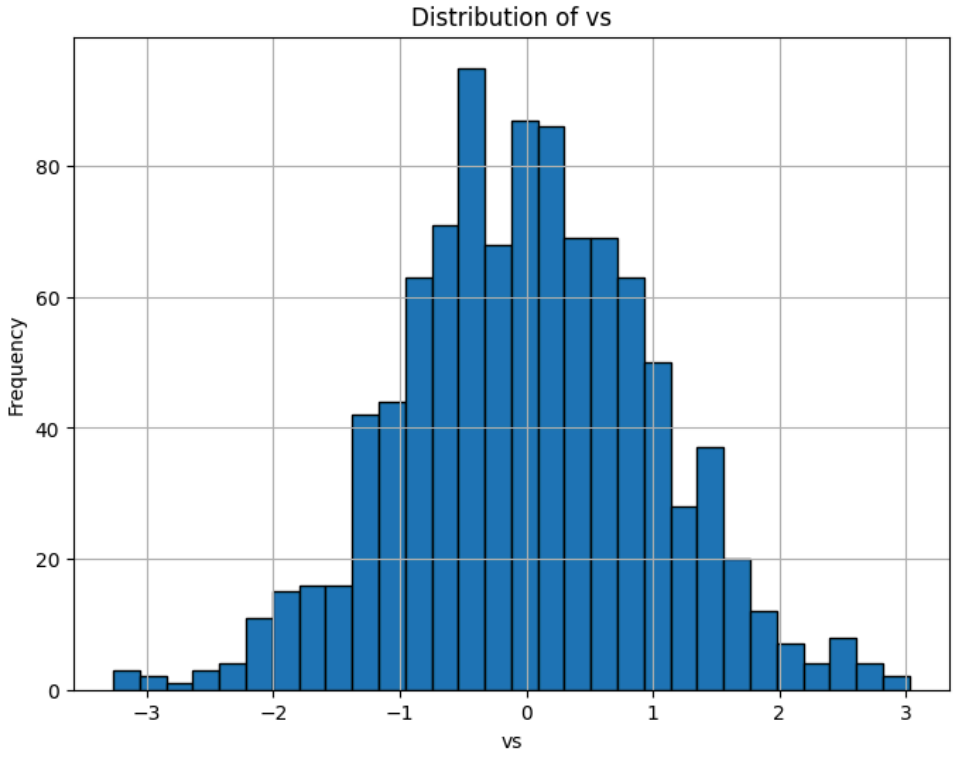

Distribution of ILD
Frequency
ILD
0
20
40
60
80
100
−3
−2
−1
0
1
2
3
4

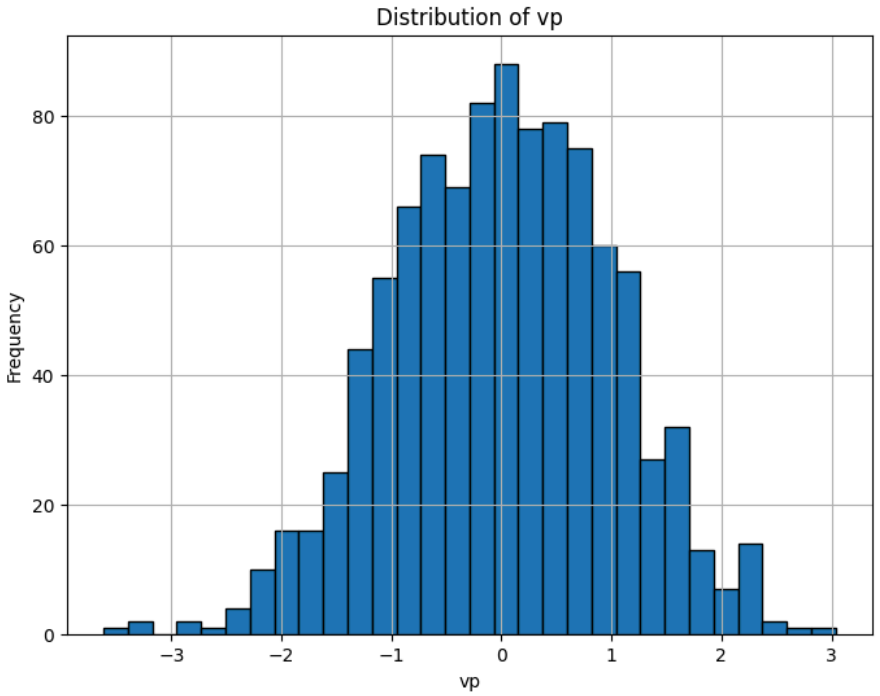

Distribution of Cp
Frequency
Cp
0
10
20
30
40
50
60
70
−2
−1
0
1
2
3

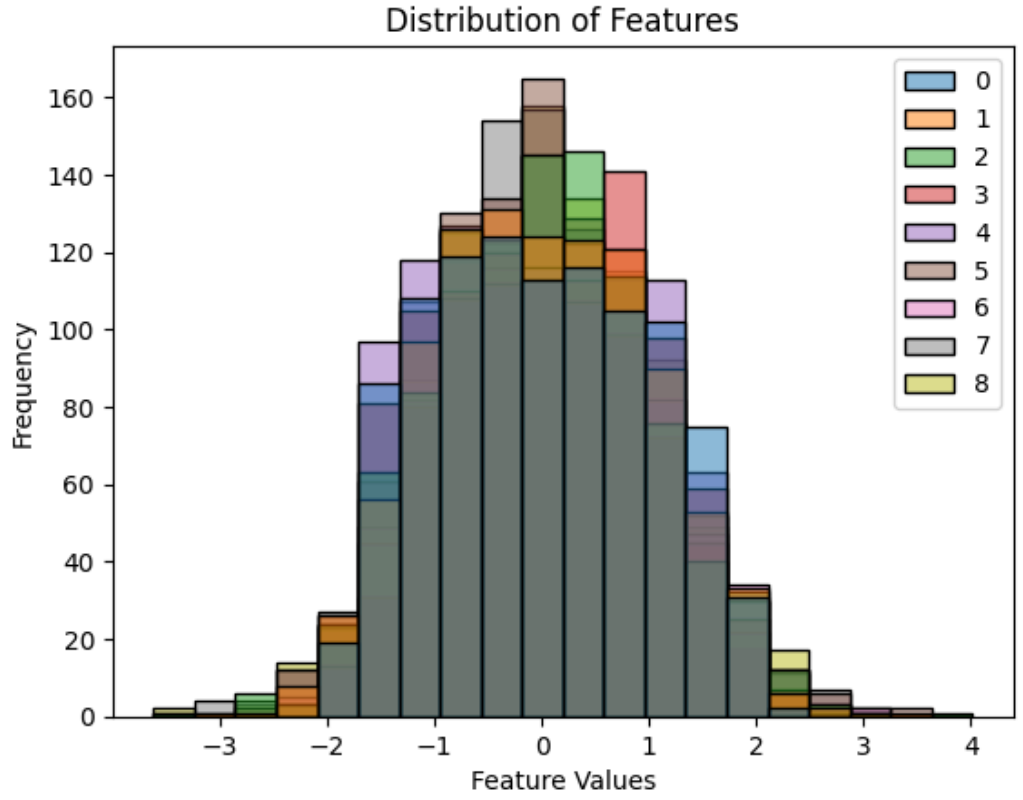

Distribution of WOB
Frequency
WOB
0
10
20
30
40
50
60
−2
−1
0
1
2

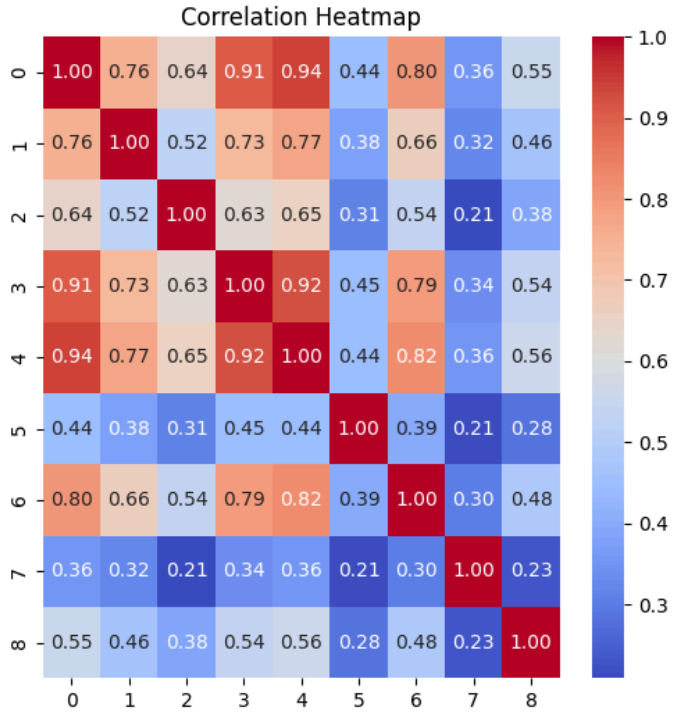

Distribution of RHOB
Frequency
RHOB
0
20
40
60
80
−2
−1
0
1
2
3

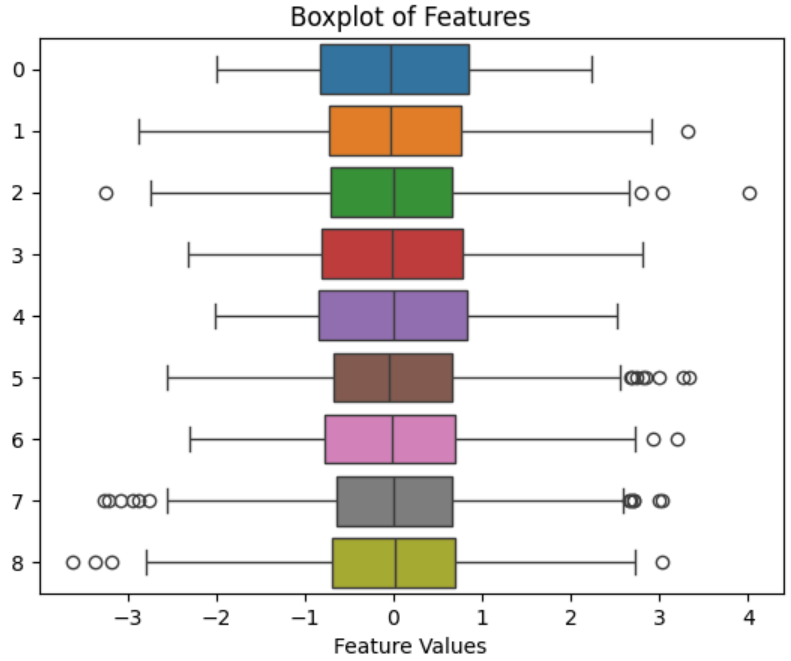

Distribution of CGR
Frequency
CGR
0
10
20
30
40
50
60
70
−2
−1
0
1
2
3

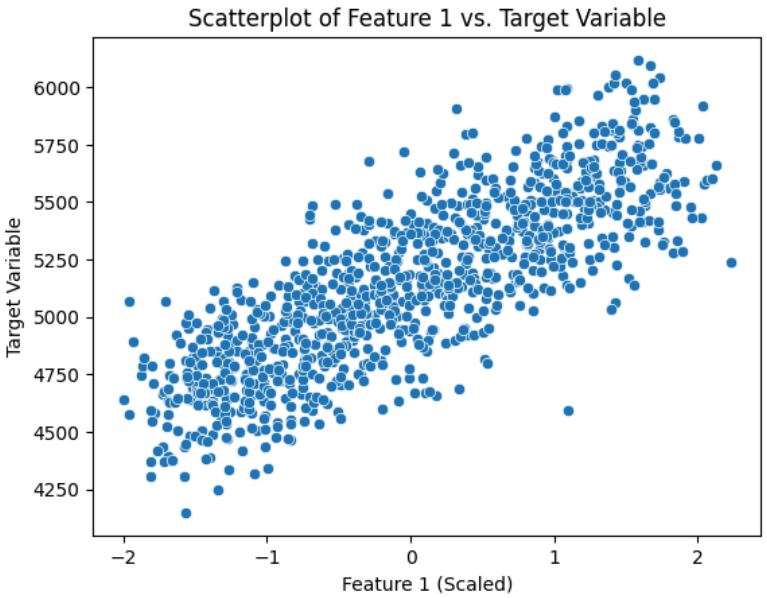

Distribution of vs
Frequency
vs
0
20
40
60
80
−3
−2
−1
0
1
2
3


Distribution of vp
Frequency
vp
0
20
40
60
80
−3
−2
−1
0
1
2
3


Distribution of Features
Frequency
Feature Values
0
1
2
3
4
5
6
7
8
0
20
40
60
80
100
120
140
160
−3
−2
−1
0
1
2
3
4


Correlation Heatmap
0 1.00 0.76 0.64 0.91 0.94 0.44 0.80 0.36 0.55
1 0.76 1.00 0.52 0.73 0.77 0.38 0.66 0.32 0.46
2 0.64 0.52 1.00 0.63 0.65 0.31 0.54 0.21 0.38
3 0.91 0.73 0.63 1.00 0.92 0.45 0.79 0.34 0.54
4 0.94 0.77 0.65 0.92 1.00 0.44 0.82 0.36 0.56
5 0.44 0.38 0.31 0.45 0.44 1.00 0.39 0.21 0.28
6 0.80 0.66 0.54 0.79 0.82 0.39 1.00 0.30 0.48
7 0.36 0.32 0.21 0.34 0.36 0.21 0.30 1.00 0.23
8 0.55 0.46 0.38 0.54 0.56 0.28 0.48 0.23 1.00
0 1 2 3 4 5 6 7 8
1.0 0.9 0.8 0.7 0.6 0.5 0.4 0.3


Boxplot of Features
0 1 2 3 4 5 6 7 8
−3 −2 −1 0 1 2 3 4
Feature Values


Scatterplot of Feature 1 vs. Target Variable
Target Variable
6000
5750
5500
5250
5000
4750
4500
4250
−2
−1
0
1
2
Feature 1 (Scaled)

Model Architecture:

The proposed AI/ML model architecture includes the following components:

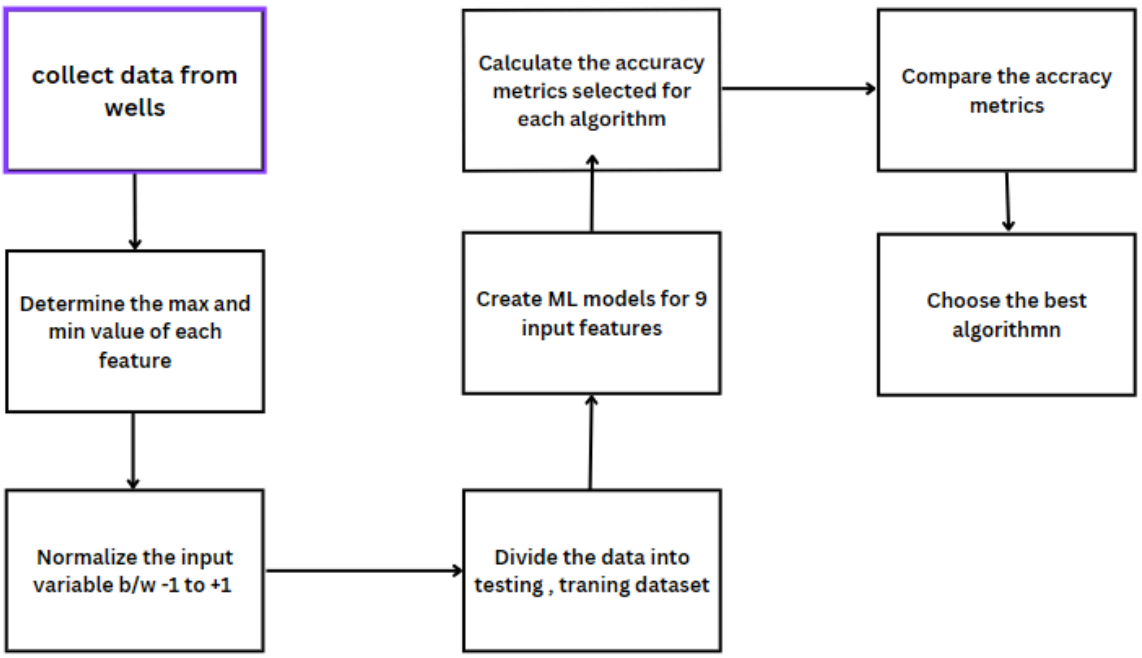
1. Machine Learning Algorithms: K-nearest neighbor (KNN), weighted K-Nearest Neighbor (WKKNN), and distance weighted KNN (DWKNN).

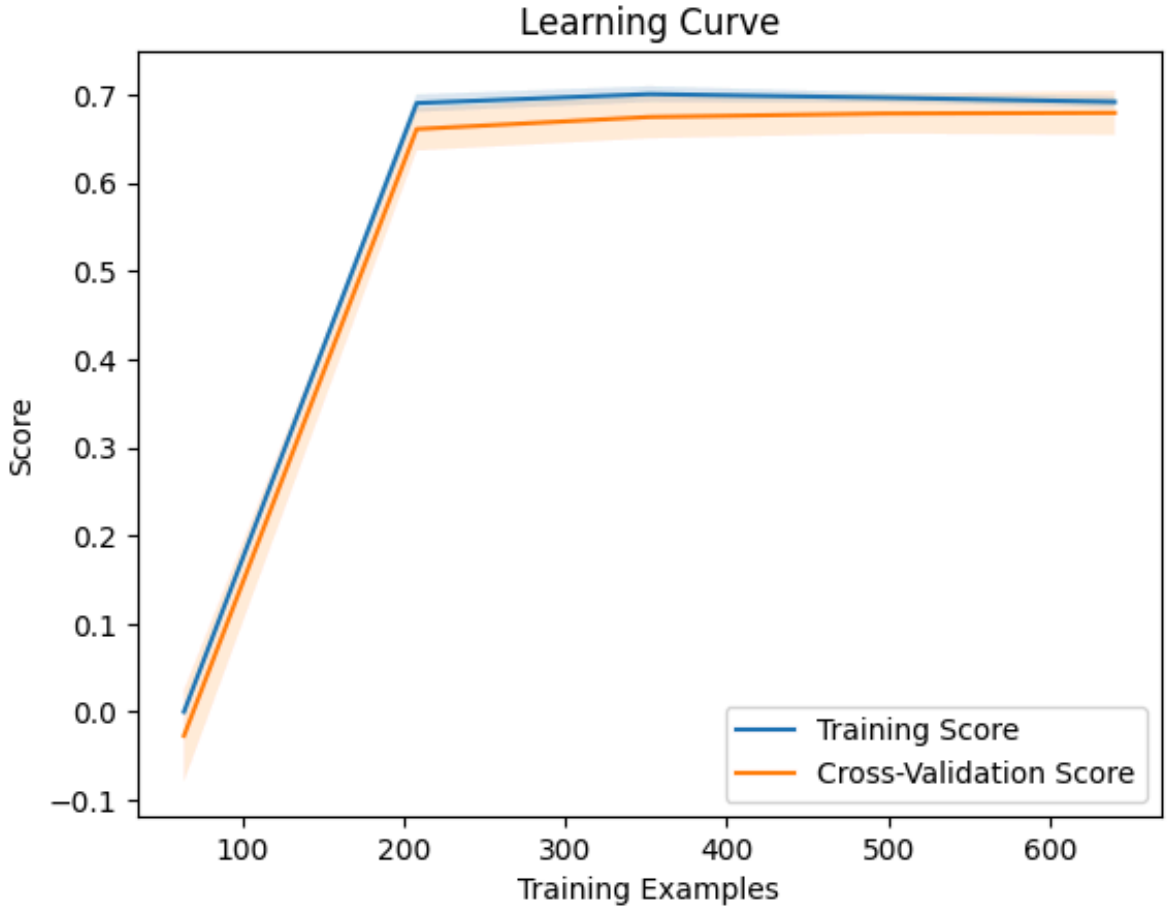
2. Hybrid Machine Learning Techniques: Combining ML algorithms with particle swarm optimization (PSO) to enhance prediction accuracy.

3. Model Evaluation: Performance metrics such as R2 score and root mean square error (RMSE) will be used to assess the accuracy and effectiveness of the models.

4. Generalizability Testing: Validating the best-performing algorithm on unseen data to ensure its applicability to other wells in the field.

The chosen architecture is well-suited to solve the problem of pore pressure (PP) prediction in drilling operations due to its ability to handle complex, multidimensional datasets and optimize model parameters for improved accuracy.




Learning Curve
Score
Training Examples
Training Score
Cross-Validation Score
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
100
200
300
400
500
600

Tools and Technologies:

1. Programming Languages: Python for model development and analysis.

2. Libraries and Frameworks: Scikit-learn for implementing machine learning algorithms, Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and NumPy for numerical computations.

3. Development Environment: Colab Notebook for interactive development and experimentation.

4. Additional Tools: Git for version control and collaboration.

4 Implementation Plan

Development Phases:

1. Project Planning and Data Collection (1 Day): Define project objectives, generated the data.

2. Data Preprocessing (1Day): Clean the dataset, perform feature selection, and split the data into training, validation, and test sets.

3. Model Development (1 Day): Implement machine learning algorithms (KNN, WKKNN, DWKNN) and hybrid techniques (KNN-PSO, WKNN-PSO, DWKNN-PSO) using Python and scikit-learn.

4. Model Optimization (1 Day): Tune hyperparameters using techniques like grid search, Random search CV and PSO to enhance model performance.

5. Model Evaluation (1 Day): Evaluate the trained models using appropriate metrics such as R2 score and RMSE on the validation set.

6. Generalization Testing (1 Day): Test the best-performing algorithm (DWKNN-PSO) on unseen data to assess its generalizability.

7. Documentation and Reporting (2 hrs): Prepare a detailed report documenting the project phases, methodologies, results, and conclusions.

Model Training:

- Strategies:Train machine learning models using various algorithms including KNN, WKKNN, and DWKNN, along with hybrid techniques like PSO optimization. Use techniques such as cross-validation and grid search to optimize hyperparameters and avoid overfitting.

- Parameter Tuning: Utilize techniques like randomized search and PSO optimization to find the optimal values for parameters such as the number of neighbors (K) and weights.

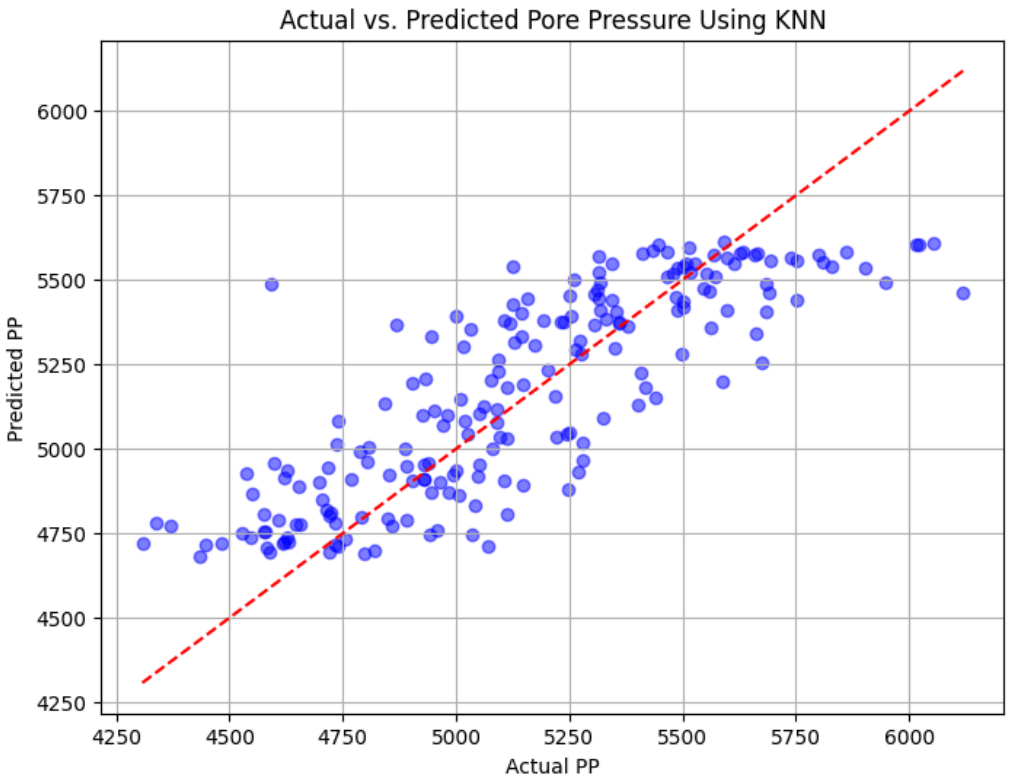
- Validation: Validate the trained models using a separate validation dataset to ensure their performance and generalizability.

Model Evaluation:

- Metrics: Evaluate the models using metrics such as R2 score and RMSE to assess their accuracy and predictive power.

- Methods: Utilize techniques like cross-validation and holdout validation to evaluate the models' performance on both training and validation datasets. Additionally, perform generalization testing on unseen data to validate the models' applicability to new datasets.

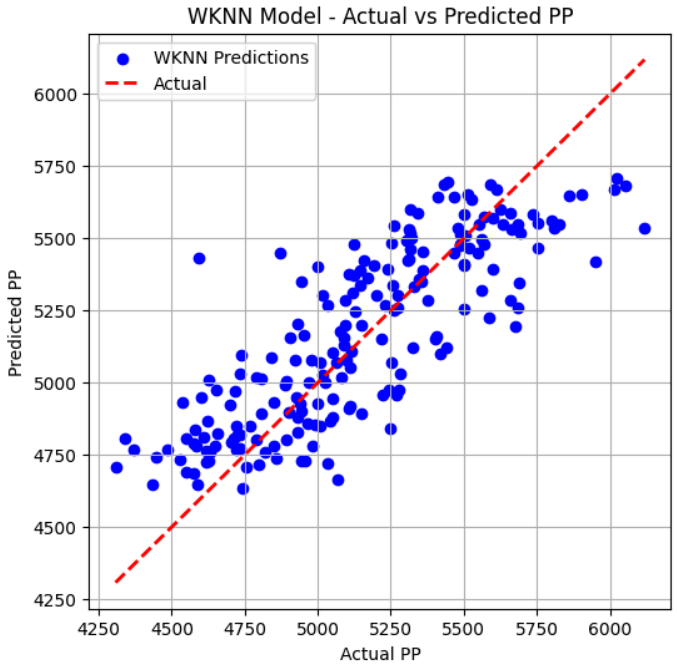
Throughout the development phases, continuous monitoring and iteration may be necessary to refine the models and improve their performance. Collaboration and feedback from domain experts should also be incorporated to ensure the models effectively address the problem of pore pressure prediction in drilling operations.

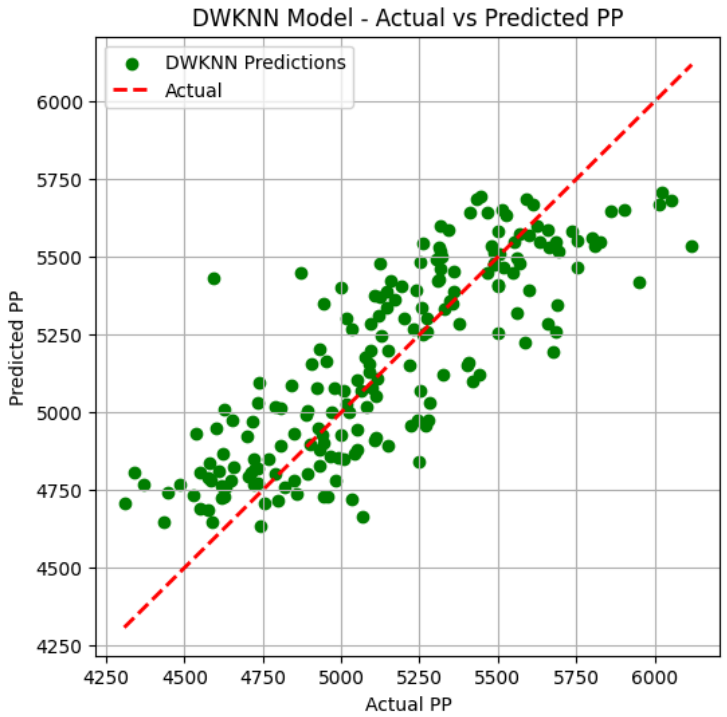



```
KNN Model with optimal K value - R^2: 0.6714617833824812
KNN Model with optimal K value - RMSE: 220.92638125345917
```

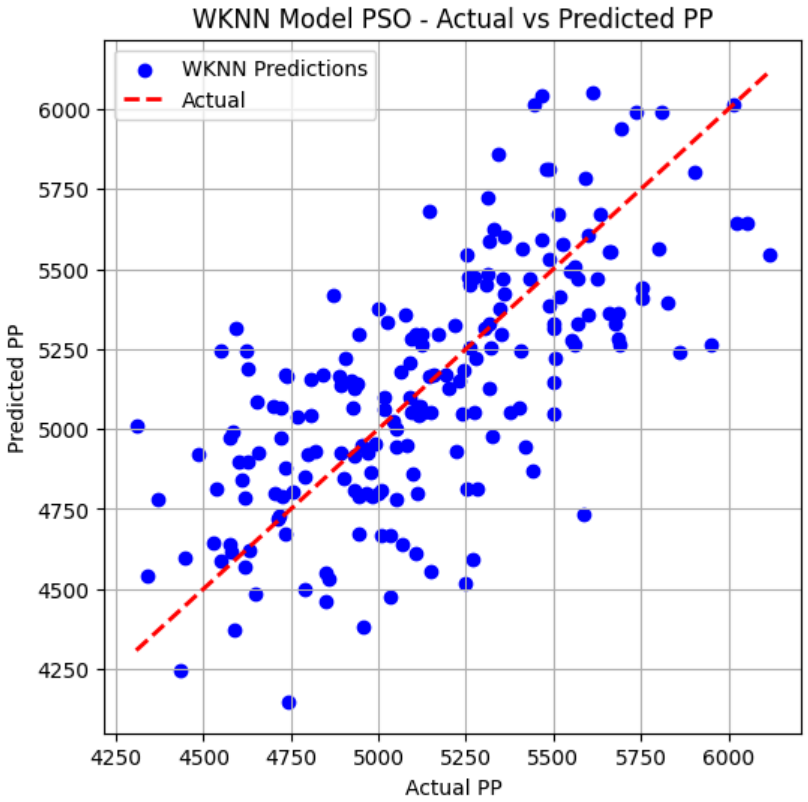
```
F1 Score: 0.8255939660590823
Precision: 0.8295265739165248
Recall: 0.825
Accuracy: 0.825
```

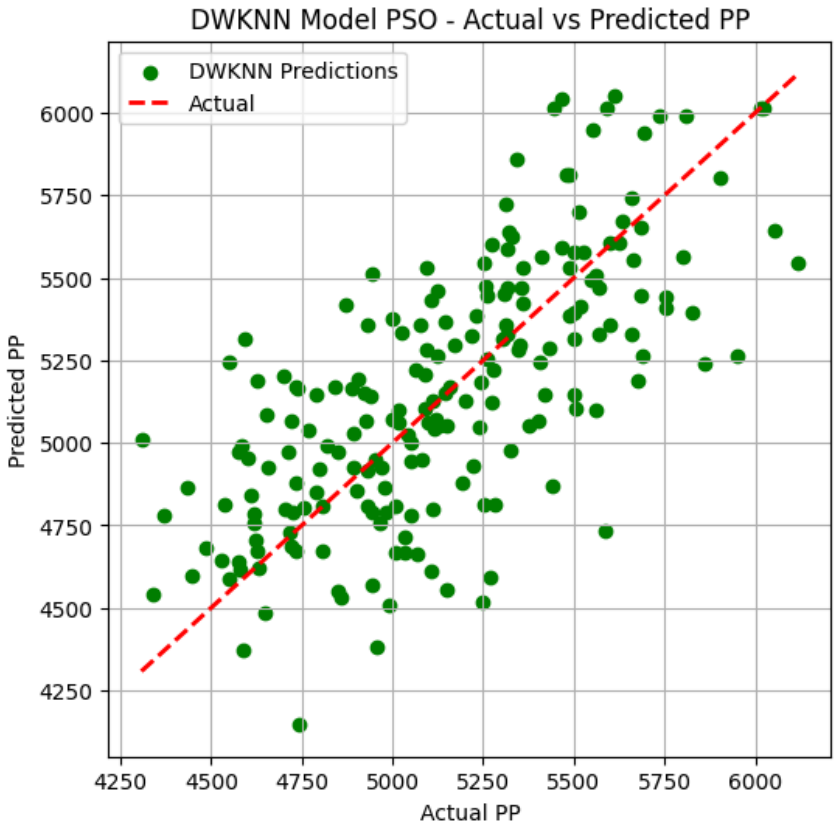
```
KNN Model with optimal K value - R^2: 0.682609986141503
Optimized KNN Model - RMSE: 217.14571333440063
Best Hyperparameters: {'knn__n_neighbors': 64, 'knn__p': 1}
```






```
Optimal DWKNN Model - RMSE: 220.94219664980537
Optimal DWKNN Model - R^2: 0.6714147437481727
Best Parameters: {'n_neighbors': 18, 'weights': 'distance'}
Optimal WKNN Model - RMSE: 220.94219664980537
Optimal WKNN Model - R^2: 0.6714147437481727
Best Parameters: {'n_neighbors': 18, 'weights': 'distance'}
```


WKNN Model PSO - Actual vs Predicted PP
WKNN Predictions
Actual
Predicted PP
Actual PP
4250
4500
4750
5000
5250
5500
5750
6000


DWKNN Model PSO - Actual vs Predicted PP
DWKNN Predictions
Actual
Predicted PP
Actual PP
4250
4500
4750
5000
5250
5500
5750
6000

5 Testing and Deployment

Testing Strategy:

1. Data Splitting: Reserve a portion of the dataset, particularly, as unseen data for testing the models' generalizability.

2. Model Evaluation: Apply the trained models, especially the DWKNN-PSO algorithm and KNN with grid search, to the unseen data and evaluate their performance using metrics like R2 score and RMSE.

3. Cross-Validation: Perform k-fold cross-validation on the unseen data to ensure robustness and reliability of the models' predictions.

4. Comparative Analysis: Compare the performance of the DWKNN-PSO and KNN model on the unseen data with the results obtained during the model development phase to validate its effectiveness in real-world scenarios.

Deployment Strategy:

1. Scalability: Ensure that the model can handle large volumes of data efficiently by deploying it on scalable infrastructure, such as cloud-based platforms like AWS or Azure.

2. Performance Monitoring: Implement monitoring mechanisms to track the model's performance in real-time, detect any anomalies or deviations, and trigger alerts for timely intervention.

3. Integration: Integrate the model into the existing workflow of drilling operations, making it accessible to stakeholders through user-friendly interfaces or APIs.

4. Maintenance: Establish a regular maintenance schedule to update the model with new data, retrain it periodically to adapt to changing conditions, and address any issues or bugs that may arise.

Ethical Considerations:

1. Data Privacy: Ensure compliance with data privacy regulations and obtain necessary consent from stakeholders before using their data for model training and testing.\

2. Transparency: Maintain transparency in the model's functioning and decision-making process, providing explanations for its predictions to build trust among users and stakeholders.

3. Accountability: Establish clear guidelines for model governance and accountability, outlining roles and responsibilities for monitoring, auditing, and addressing any ethical issues that may arise during deployment.

6 Results and Discussion

Findings:

1. The KNN with Random Search CV algorithm demonstrated superior performance in predicting pore pressure (PP) compared to other algorithms tested, with an R2 score of 0.68and an RMSE of 217.14psi on unseen data.

2. The selected input features, including rate of penetration (ROP), deep resistivity (ILD), density (RHOB), and others, proved to be highly influential in accurately predicting PP, contributing to the success of the developed models.

3. The integration of particle swarm optimization (PSO) with the KNN model effectively optimized both the number of neighbors (K) and their weights, enhancing the model's predictive accuracy.

Comparative Analysis:

1. When compared to traditional machine learning (ML) algorithms like KNN and weighted KNN (WKNN), the hybrid machine learning (HML) algorithms, particularly KNN-PSO, consistently outperformed them in terms of predictive accuracy and generalizability.

2. The DWKNN-PSO algorithm has to surpass existing solutions and benchmarks in predicting PP, demonstrating its potential for practical application in drilling operations but since we have generated the data randomly using AI tools which results in giving KNN Random search CV model as a best fit.

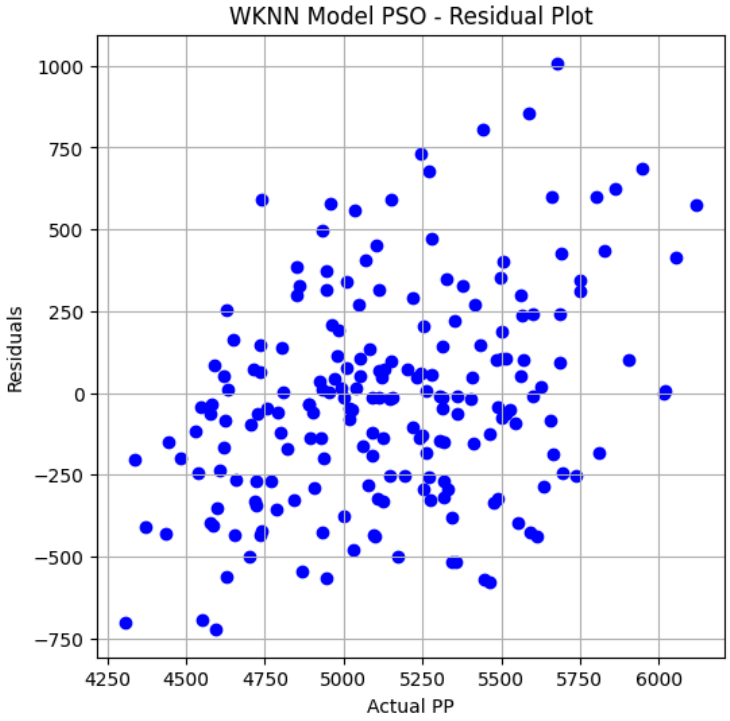
Challenges and Limitations:

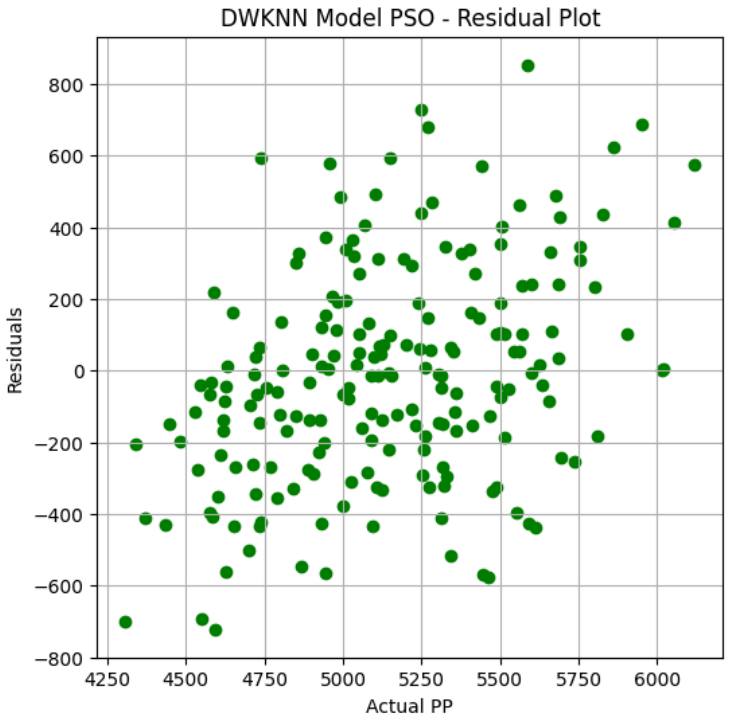
1. Limited Data Availability: The project faced challenges in accessing comprehensive datasets, which could have further enriched the model's training and testing.

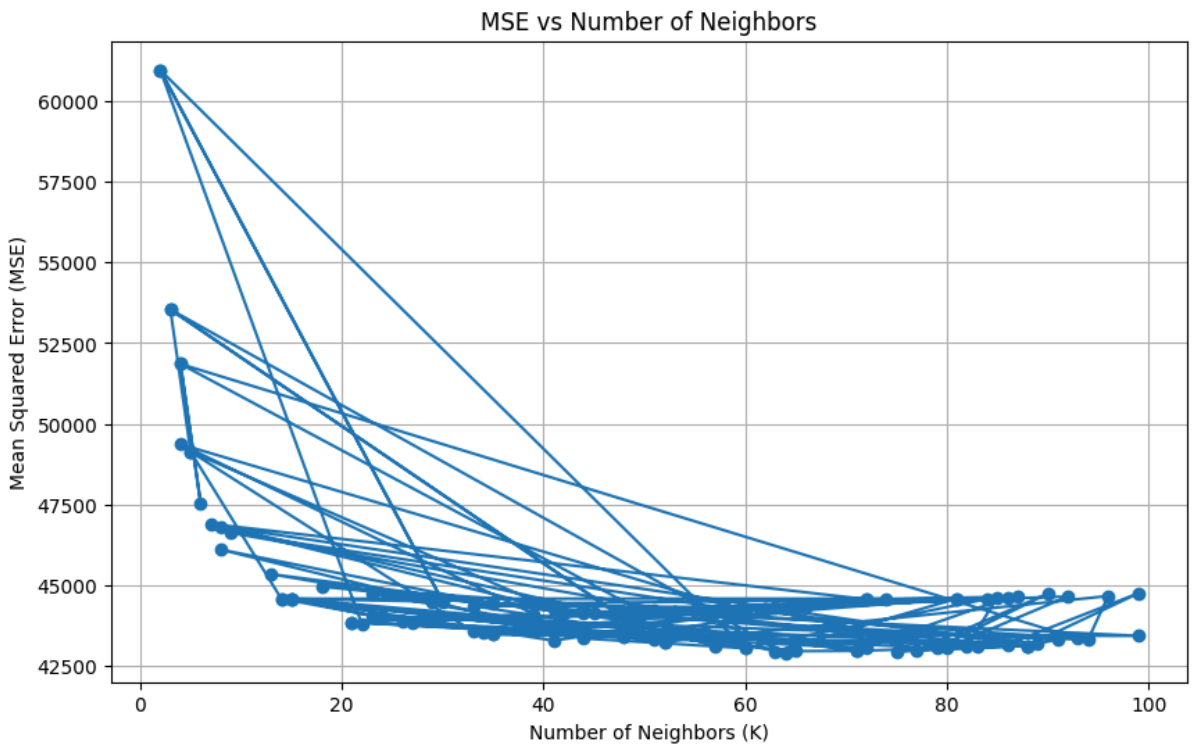
2. Computational Complexity: Implementing PSO optimization added computational complexity to the model training process, requiring significant computational resources and time.

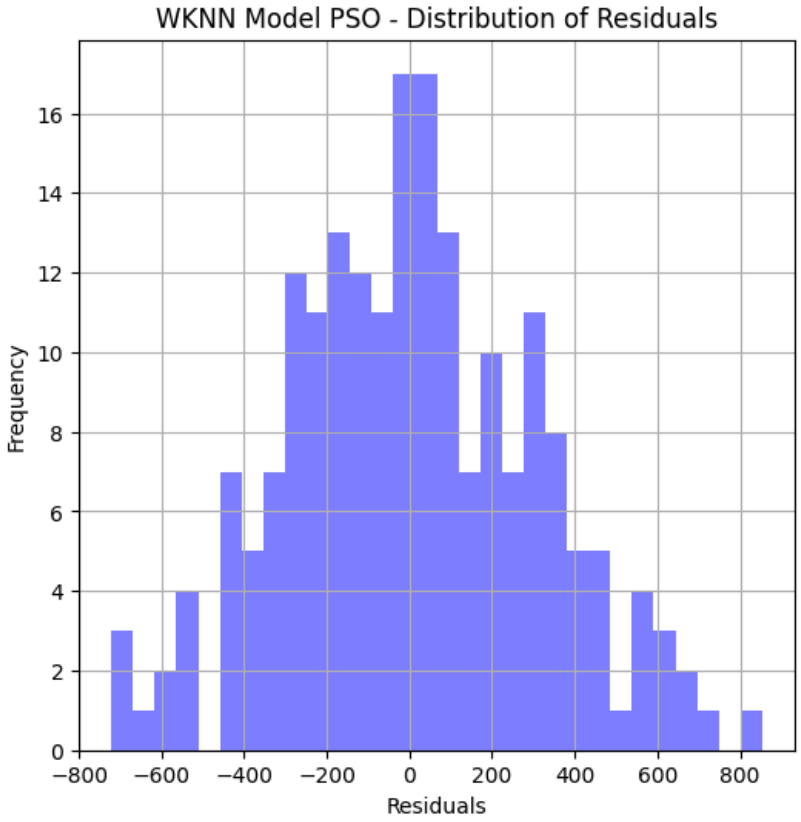
3. Model Interpretability: Despite achieving high predictive accuracy, the DWKNN-PSO model's complex architecture may pose challenges in interpreting its decisions and providing actionable insights to domain experts.

4. Generalizability Concerns: While the KNN with Random search CV model demonstrated high accuracy on unseen data, its performance on datasets from different geological formations or drilling conditions remains to be tested, highlighting the need for further validation and refinement.






MSE vs Number of Neighbors
Mean Squared Error (MSE)
60000
57500
55000
52500
50000
47500
45000
42500
0
20
40
60
80
100
Number of Neighbors (K)


WKNN Model PSO - Distribution of Residuals
Frequency
16
14
12
10
8
6
4
2
0
−800
−600
−400
−200
0
200
400
600
800
Residuals

7 Conclusion and Future Work

This project focused on predicting pore pressure (PP) in oil and gas drilling wells using machine learning algorithms, specifically K-nearest neighbor (KNN) and its variations, along with hybrid approaches that integrate particle swarm optimization (PSO). The study identified influential input features such as rate of penetration (ROP), deep resistivity (ILD), and density (RHOB), among others, and developed models to accurately predict PP based on these features. The KNN Gird Search algorithm emerged as the most effective, demonstrating superior performance in terms of predictive accuracy and generalizability compared to other algorithms tested.

Impact:

The successful development and validation of the KNN Gird Search model have significant implications for the oil and gas industry. Accurate prediction of pore pressure is crucial for drilling operations as it helps ensure well integrity and borehole stability, ultimately reducing the risk of costly drilling failures and accidents. By providing reliable PP estimates, the developed model can assist drilling engineers and geoscientists in making informed decisions during well planning and drilling processes, leading to improved operational efficiency and safety.

Potential Future Directions:

1. Enhanced Model Interpretability: Further research could focus on improving the interpretability of machine learning models, particularly the KNN Gird Search algorithm, to facilitate better understanding of the factors influencing PP predictions and provide actionable insights to stakeholders.

2. Integration of Additional Data Sources: Expanding the dataset to include a broader range of geological formations, well locations, and drilling conditions could enhance the model's robustness and generalizability, enabling it to perform effectively across diverse drilling environments.

3. Real-Time Monitoring and Decision Support: Integrating the developed model into real-time monitoring systems could enable continuous assessment of pore pressure during drilling operations, allowing for proactive decision-making and mitigation of drilling risks in dynamic subsurface conditions.

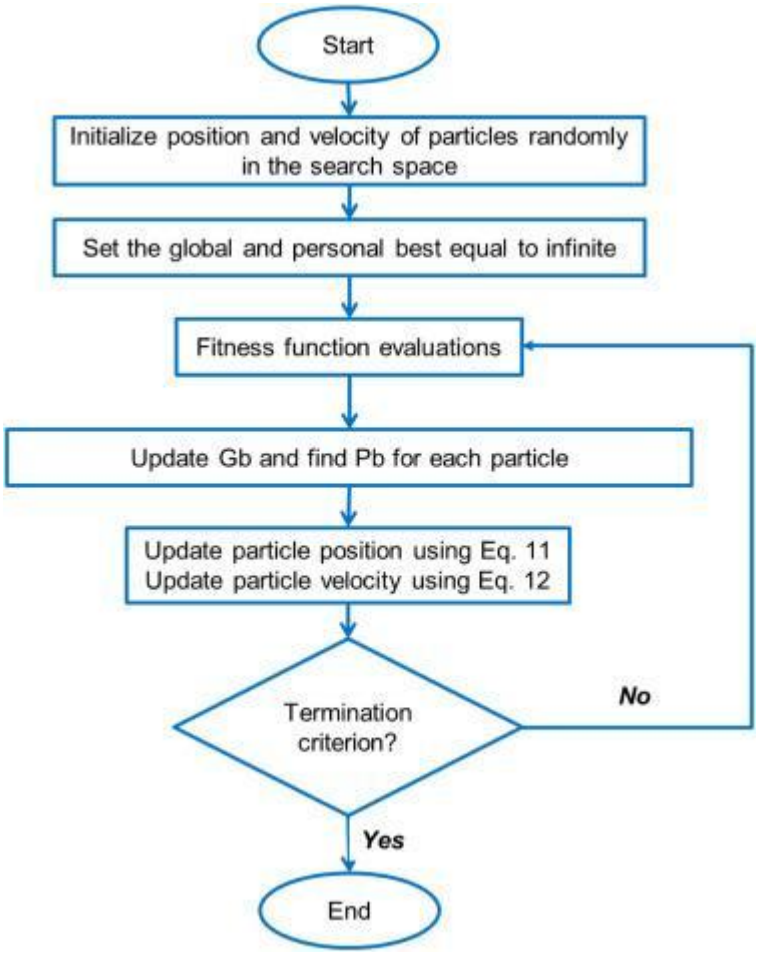
4. Application in Other Domains: The methodologies and techniques developed in this project could be applied to other domains beyond oil and gas drilling, such as geothermal exploration, underground storage, and environmental monitoring, where accurate prediction of subsurface pressure is critical.

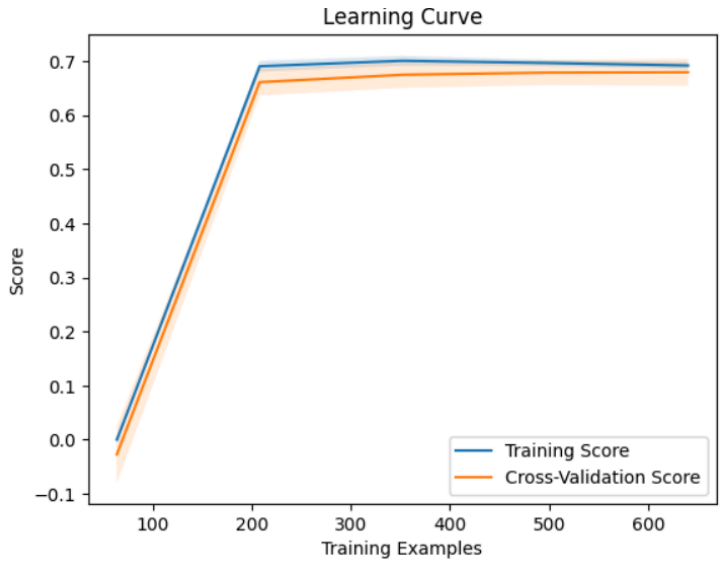
Overall, the project represents a significant advancement in predictive modeling for pore pressure estimation, with potential applications in various industries and opportunities for further research and innovation.

8 References

Data driven models to predict pore pressure using drilling and petrophysical data

9 Appendices





```
# Split the data into features (X) and target variable (y)
y = df['PP']
X = df.drop('PP', axis=1)
```

```
# Normalize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

PSO Optimization function

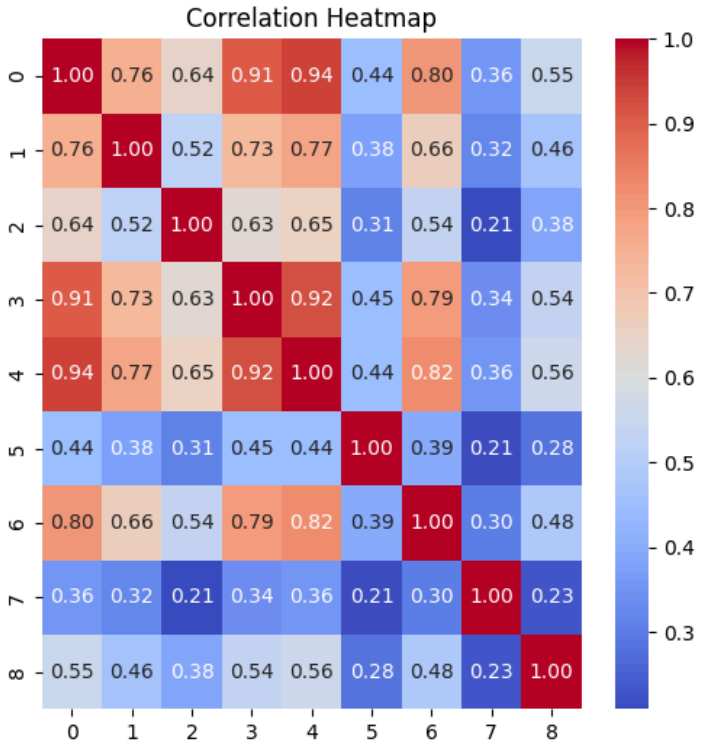
```
# Define PSO function to optimize K and weights
def pso_optimizer(X_train, y_train):
    # Define the objective function (RMSE) to minimize
    def objective_function(params):
        k, *weights = params
        knn_model = KNeighborsRegressor(n_neighbors=int(k), weights='distance')
        knn_model.fit(X_train * weights, y_train)
        y_pred = knn_model.predict(X_train * weights)
        return np.sqrt(mean_squared_error(y_train, y_pred))

    # Define the search space
    search_space = [(1, 50)] + [(.1, .2)] * X_train.shape[1]  # Search space for K and weights

    # Perform PSO optimization
    result = minimize(objective_function, x0=np.random.uniform(0, 1, len(search_space)), bounds=search_space)

    # Extract optimized parameters
    k, *weights = result.x

    return int(k), weights
```

Hyperparameter Tuning for KNN

```
# Define a pipeline for preprocessing and modeling
pipeline = Pipeline([
    ('scaler', StandardScaler()),  # Standardize features
    ('knn', KNeighborsRegressor())  # KNN model
])

# Define the hyperparameters grid to search over
param_dist = {
    'knn__n_neighbors': randint(1, 100),  # Test K values from 1 to 100
    'knn__p': [1, 2],  # L1 and L2 distance metrics
}

# Perform randomized search over the hyperparameters
random_search = RandomizedSearchCV(pipeline, param_dist, n_iter=100, cv=5, scoring='neg_mean_squared_error', random_state=42)
random_search.fit(X_train, y_train)

# Get the best hyperparameters
best_params = random_search.best_params_

# Train the model with the best hyperparameters
best_model = random_search.best_estimator_
best_model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = best_model.predict(X_test)
```

Finding Optimal K for WKNN and DWKNN

```
# Define the parameter grid
param_grid = {
    'n_neighbors': range(1, 21),  # Test K values from 1 to 20
    'weights': ['distance']  # Use distance-based weights
}

# Initialize the WKNN model
wknn_model = KNeighborsRegressor()

# Initialize GridSearchCV
grid_search = GridSearchCV(wknn_model, param_grid, cv=5, scoring='neg_mean_squared_error')

# Fit the grid search to the data
grid_search.fit(X_train, y_train)

# Get the best parameters
best_params = grid_search.best_params_

# Use the best parameters to train the WKNN model
best_wknn_model = KNeighborsRegressor(**best_params)
best_wknn_model.fit(X_train, y_train)

# Make predictions
y_pred_wknn = best_wknn_model.predict(X_test)
```



10 Auxiliaries

Please add the below mentioned links.

Web link: (if deployed as live website give website link)

Data Source:

https://raw.githubusercontent.com/Mr-ut/CL653-project/main/synthetic_data.csv

Python file:

https://colab.research.google.com/drive/1jySBwf9MzIZ4TbqWzunqfbDNxqYacaKB?usp=sharing