ME 5201: Computational Methods in Engineering

Machine Learning Project

N. Sai Teja
ME22S084

Problem Statement:

To measure the velocity of the sound wave propagation in a given fluid medium using ML approach.

Solution Procedure:

- The goal is to fit an approximate model to measure the velocity of the sound wave propagation which is dependent on density of the fluid and the bulk modulus of the fluid.
- For finding the approximate fit, the experimental data set provided is used for training, testing and validation of the model.
- To get an approximate idea of the variation of the velocity of the sound wave propagation with density and bulk modulus, firstly a 3D scatter plot is plotted for the given data set.
- Below figure shows the scatter plot for the velocity of sound wave propagation.

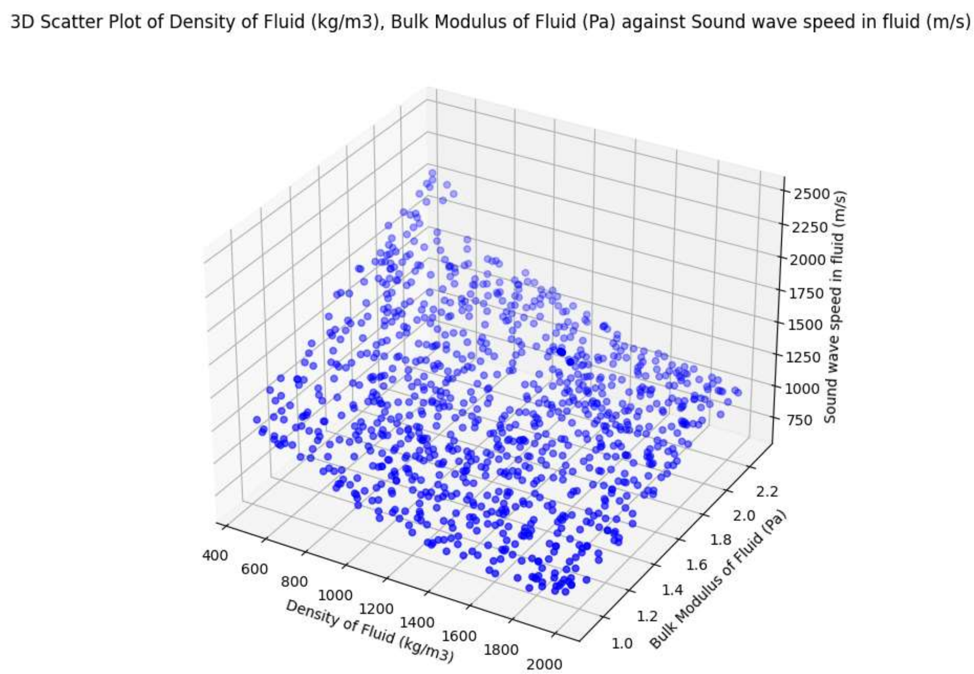

Figure 1: 3D scatter plot for sound wave velocity

- In the next step, an approximate model using linear regression is fit to the given data set. Model is trained considering the "Mean square error" as a loss function. The goal is to minimize the loss.
- The approximate model fitting is done using the in-built python libraries like "sklearn.preprocessing", "sklearn.linear_model", "sklearn.metrics", by importing features like "polynomialFeatures", "Linearregression", "mean_squared_error" and "r2_score".
- The model is trained for 70% of the data and remaining 30% data is used for testing and validation.

- Performance metrics like mean square error and r2 score are calculated for the dataset and equation of the fit is obtained.
- Next a function for predicting the velocity of the sound wave propagation is developed. It takes the input values of density and bulk modulus of the fluid from the user and predicts the velocity based on the approximate model.
- The actual values are read from the dataset and are printed as output along with the predicted values for comparison.
- This process is repeated thrice by increasing the degree of polynomial. The approximate model is fit is done using linear, quadratic and cubic regression.
- The approximate equation of fit for all the three cases is obtained. The performance metrics in each case are calculated and a 3D scatter plot for the approximate fit is plotted in each case.
- Next, a simple Feed Forward Neural Network [FNN] architecture is developed for training the model.
- The hyper parameters of the developed FNN are:
    - Number of layers = 1
    - Number of neurons = 300
    - Learning rate = 0.001
    - Epochs = 1000
- The model is trained using gradient descent, by considering Mean Square Error as loss.
- Next the model is validated, performance metrics are calculated and a 3D scatter plot for actual vs predicted velocities is plotted.
- A function for predicting the values from the FNN network is developed. It takes the inputs of density and bulk modulus from the user and predicts the value of the velocity.
- In the next step FNN architecture is modified by changing the hyper parameters for comparing with the previous architecture. The modified parameters are as follows:
    - Number of layers = 2
    - Number of neurons = [100 10]   {100 neurons in layer 1 and 10 in layer 2}
    - Learning rate = 0.001
    - Epochs = 1000
- In this case also a function for predicting the velocity is developed and the performance metrics are calculated.

Results:

For linear regression the following are the equation of the approximate fit, mean square error and r2 score:

- Equation:
  $z = 1192.070269671431 + 0.0 * x^{\wedge(0)} * y^{\wedge(0)} + -0.49111740228522727 * x^{\wedge(1)} * y^{\wedge(0)} + 3.3508663077608247e^{-07} * x^{\wedge(0)} * y^{\wedge(1)}$
- Mean Squared Error (MSE): 16736.027178290216
- Root Mean Squared Error (RMSE): 129.3677980731303
- R-squared (R2): 0.7897533203790228

3D scatter plot for linear regression:

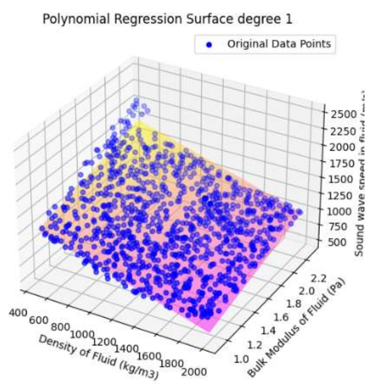

Figure 2: 3D scatter plot for linear regression

For quadratic regression:

- Equation: $z = 505.44771562316 + 0.0*x^{\wedge(0)}*y^{\wedge(0)} + 3.0802747904453863e^{-08}*x^{\wedge(1)}*y^{\wedge(0)} + 7.547372312200253e^{-07}*x^{\wedge(0)}*y^{\wedge(1)} + 0.00010848399590876049*x^{\wedge(2)}*y^{\wedge(0)} + -4.593776727543904e^{-10}*x^{\wedge(1)}*y^{\wedge(1)} + 4.8806289883699e^{-17}*x^{\wedge(0)}*y^{\wedge(2)}$
- Mean Squared Error (MSE): 13317.377906707707
- Root Mean Squared Error (RMSE): 115.40094413265304
- R-squared (R2): 0.8327001709357226

3D scatter plot for quadratic regression:

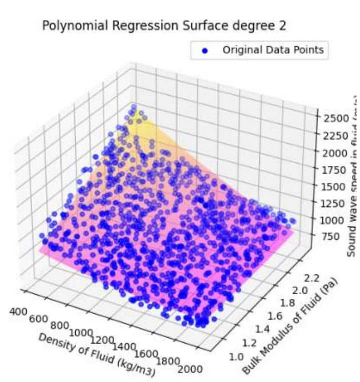

Figure 3: 3D scatter plot for quadratic regression

For cubic regression:

- Equation: $z = 771.5626915429623 + 0.0*x^{\wedge(0)}*y^{\wedge(0)} - 5.79131243017189e^{-20}*x^{\wedge(1)}*y^{\wedge(0)} - 4.691355841747578e^{-24}*x^{\wedge(0)}*y^{\wedge(1)} + 5.083745817808638e^{-24}*x^{\wedge(2)}*y^{\wedge(0)} + 1.1968422872528136e^{-18}*x^{\wedge(1)}*y^{\wedge(1)} + 3.810801769372643e^{-16}*x^{\wedge(0)}*y^{\wedge(2)} + 1.2172973701523516e^{-20}*x^{\wedge(3)}*y^{\wedge(0)} + 4.985481292567937e^{-15}*x^{\wedge(2)}*y^{\wedge(1)} - 1.6725209160695767e^{-19}*x^{\wedge(1)}*y^{\wedge(2)} + -2.7573145474082044e^{-26}*x^{\wedge(0)}*y^{\wedge(3)}$
- Mean Squared Error (MSE): 17316.897853876068
- Root Mean Squared Error (RMSE): 131.5936847036212
- R-squared (R2): 0.7824561207790163

3D scatter plot for cubic regression:

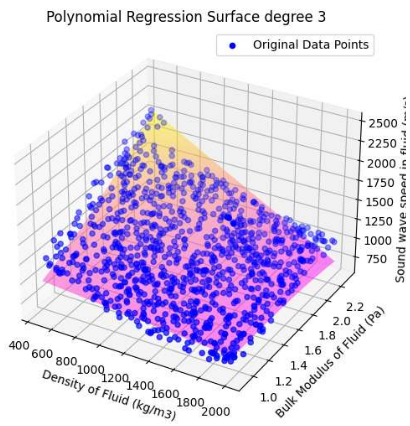

Figure 4: 3D scatter plot for cubic regression

Results using FNN:

- Mean Squared Error: 14485.633144362357
- R-squared (R2): 0.8634050226654129

3D scatter plot using FNN:

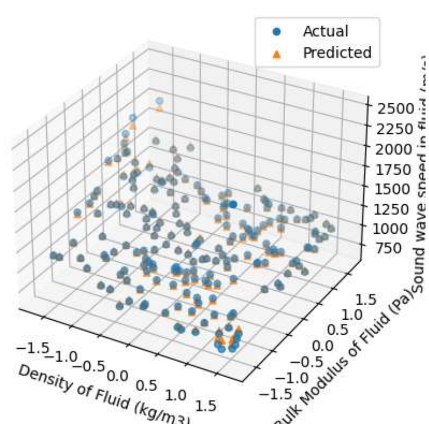

Figure 5: 3D scatter plot using FNN

Analysis and Conclusions:

- In model fitting using regression, the quadratic regression has the highest r2 score and least mean square error compared to linear and cubic.
- This may be due to the following reasons:
    - The linear regression model might have not captured most of the information as we can see from the scatter plot in figure 2.
    - Cubic regression might have led to overfitting as it is simple data set with only two independent variables and one dependent variable.
- In the model fitting using FNN with one layer and 300 neurons, the r2 score is higher than that of quadratic regression.
- The r2 score with single hidden layer with 300 neurons is higher than that compared to single layer with 200 and single layer with 500 neurons.
- The r2 score decreased significantly when the FNN with 2 layers is used, this may be due to overfitting of the data.

The following are the predictions values obtained using different methods:

- Density of fluid: $1.57e^3$
- Bulk modulus: $2.21e^9$
- Actual Velocity from the dataset: 1111.1100
    - Predicted value using linear regression: 1161.5574
    - Predicted value using quadratic regression: 1085.2913
    - Predicted value using cubic regression: 1079.8385
    - Predicted value using FNN: 1109.4904

Hence from the above data it can be observed that FNN with higher r2 score gives more accurate predictions compared to regression. And also, among the regression models the quadratic regression gives the better approximation.