PDDA machine learning competition 2021

Workflow description

Team: Iron486

Objective

The aim of this contest was to develop data-driven models to estimate reservoir properties, including shale volume, porosity, and fluid saturation, based on a common set of well logs like gamma ray, bulk density, neutron porosity, resistivity, and sonic. Log data from eight wells from the same field together with the corresponding reservoir properties estimated by petrophysicists were provided. The goal was building a data-driven model using the provided training dataset. Following that, the newly developed data-driven models was deployed on the test dataset to predict the reservoir properties based on the well-log data.

Imported Libraries

Firstly, libraries were imported. The used libraries were Pandas, Numpy, Scikit-Learn, Matplotlib, Seaborn, Keras (and Tensorflow) and Xgboost. Numpy and Pandas were used to wrangle and modify the structures of the data, both in the preprocessing steps and when the models were applied. Scikit-learn was used to train and test the different models as well as xgboost that is a library used to fit and predict XGboost algorithm on the data. Matplotlib and Seaborn were used, instead, for data visualization.

Preprocessing steps

The data were read using pandas and were displayed. Next, values corresponding to "-9999" were replaced with np.nan and all the missing values in the output columns (PHIF, SW, VSH) were dropped, passing from 318967 to 42309 rows.

The features were selected considering the most correlated features and, also, trying other combinations. The following are the selected features:
['DEPTH', 'DTC', 'DTS', 'CALI', 'DEN', 'DENC', 'GR', 'NEU', 'PEF','RDEP', 'RMED']

Later, to fill the missing values IterativeImputer algorithm was applied to the selected features. Bayesian Ridge was the estimator used in Iterative Imputer but also ExtraTreesRegressor with various number of estimators was applied, even though it gave worse results. In addition, the obtained dataframe was saved. Successively, outliers were removed using IsolationForest algorithm passing from 42309 to 38923 rows.

Some graphs showing correlations between variables were plotted. Particularly, in the notebook plots that involve gamma ray values, volume of shale and depth are shown.
Data were scaled using StandardScaler function and, finally, data were split into train and validation datasets, using the first 80 percent for the train dataset and the remaining for the validation one. Also, the features and the 3 petrophysical properties to predict were separated into 2 separated datasets.
Therefore, we have 4 different datasets: X_train, X_val, y_train, y_val.

Training and predicting the model

Lots of different models were experimented. Ensemble methods and neural networks were evaluated in particular. Moreover, hyperparameter tuning was performed for each model. The data were also divided into clusters and, for each cluster, different algorithms were applied. At the end, the best model was the model obtained not dividing into clusters the dataset. Thus, the number of clusters was set to 1 because there were no clusters considered.

Neural network was the best model for predicting VSH. It was built with 3 hidden layers and 3 batch normalization layers. The first hidden layer had 6820 neurons, the second 1200 and the third 44. These layers had an exponential linear unit as an activation function with Glorot normalization and the output layer had one neuron with a sigmoid activation function. Moreover, even other

regularizers were tested, like L1 and L2 regularizers and dropout was also applied but without any good result.
Some optimizers like Adam, Nadam, RMSprop and Nesterov Accelerated Gradient (NAG) were tested. In the best model, Adam optimizer was applied with the following hyperparameters: 'learning_rate': 5.5e-06, 'decay': 0.0, 'beta_1': 0.92, 'beta_2': 0.88, 'epsilon': 1e-08, 'amsgrad': 'False'.

GradientBoostingRegressor function was used to predict SW with the following hyperparameters: "max_depth" : [280], "min_samples_leaf": [5], "max_features": ['auto'], "max_leaf_nodes": [8] , 'n_iter_no_change': [3], 'n_estimators': [80]

Instead, for PHIF ExtraTreesRegressor with the following hyperparameters was performed:

'bootstrap': [False],"min_samples_leaf": [6], "min_weight_fraction_leaf": [0.0], "max_features": ["auto"], "max_leaf_nodes": [6000] ,'n_estimators': [80]

Also, other algorithms were tested and tuned (including neural networks) for the last two parameters, giving worse results.

The models were finally tested on validation dataset and used to predict the values of the test dataset. Again, the test dataset was read, the features were selected, the missing values were filled with IterativeImputer with BayesianRidge estimator and, finally, data were scaled.

The obtained predicted scores were saved in a .csv file called Iron486_1 and the model was submitted, yielding a score of 0.09816.

In the next two pages, R2 and RMSE metrics on train and validation datasets respectively, are shown and the metrics were calculated also for each of the three petrophysical parameters to predict. In addition, below the scores, plots with the sample number on the x axis and the petrophysical feature to estimate were plotted (Fig 1 and Fig 2). We have the predicted value in orange and the actual one in blue.

On the right, instead, plots with the actual value on the x axis and the predicted value in the y axis were plotted showing that the data, especially in the train dataset, tend to have similar values on the x and y axes.

SCORES ON TRAIN DATASET

```
RMSE:  0.02970
        PHIF   : 0.00501
        SW     : 0.04163
        VSH    : 0.02981
R^2:   0.986886
        PHIF   : 0.99560
        SW     : 0.98681
        VSH    : 0.97825
```

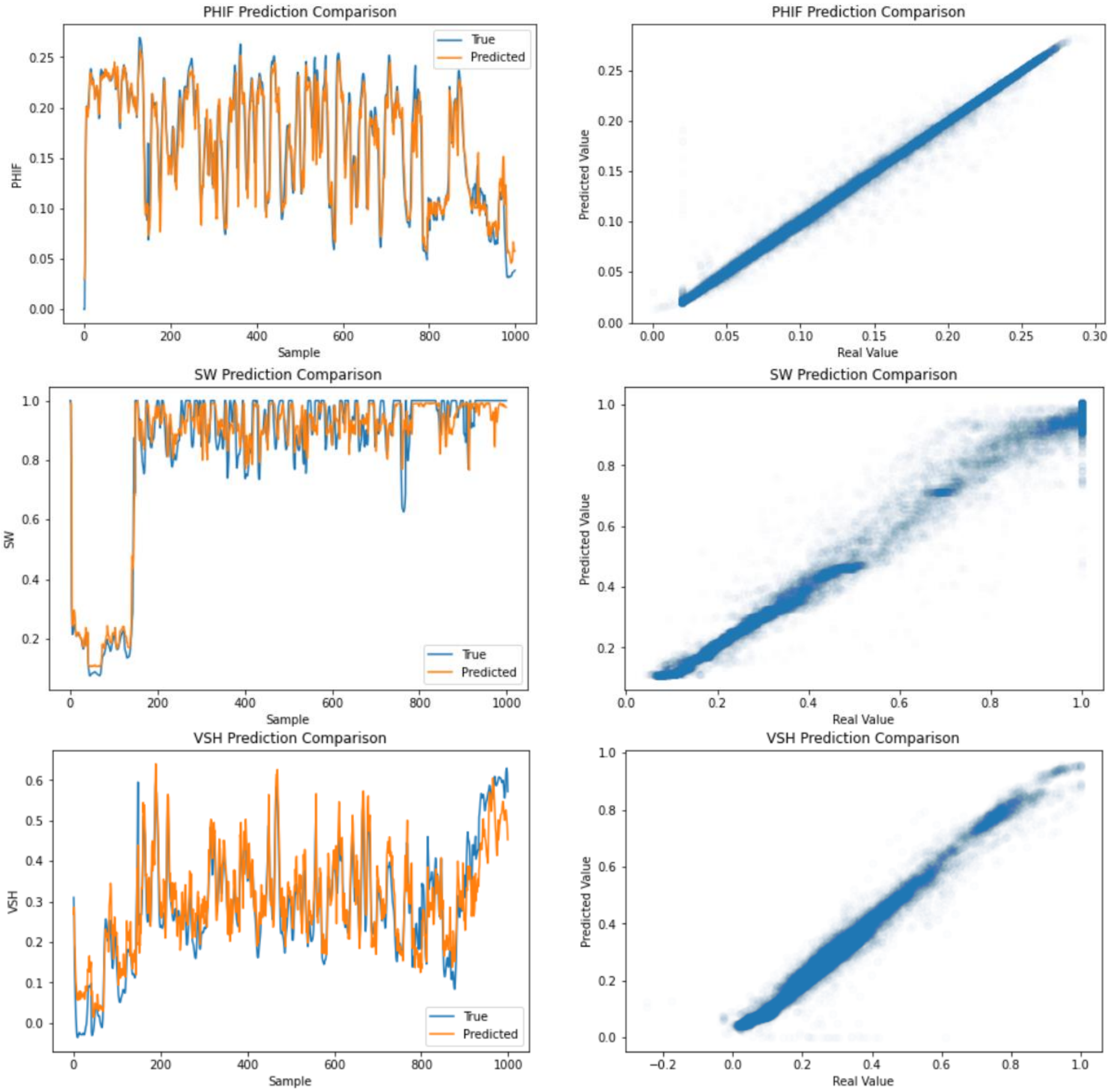

Fig 1: plots that depict the difference between predicted and actual values on train dataset

SCORES ON VALIDATION DATASET

```
RMSE: 0.05947
      PHIF  : 0.01589
      SW    : 0.09093
      VSH   : 0.04569
R^2:  0.835461
      PHIF  : 0.96641
      SW    : 0.63665
      VSH   : 0.90332
```

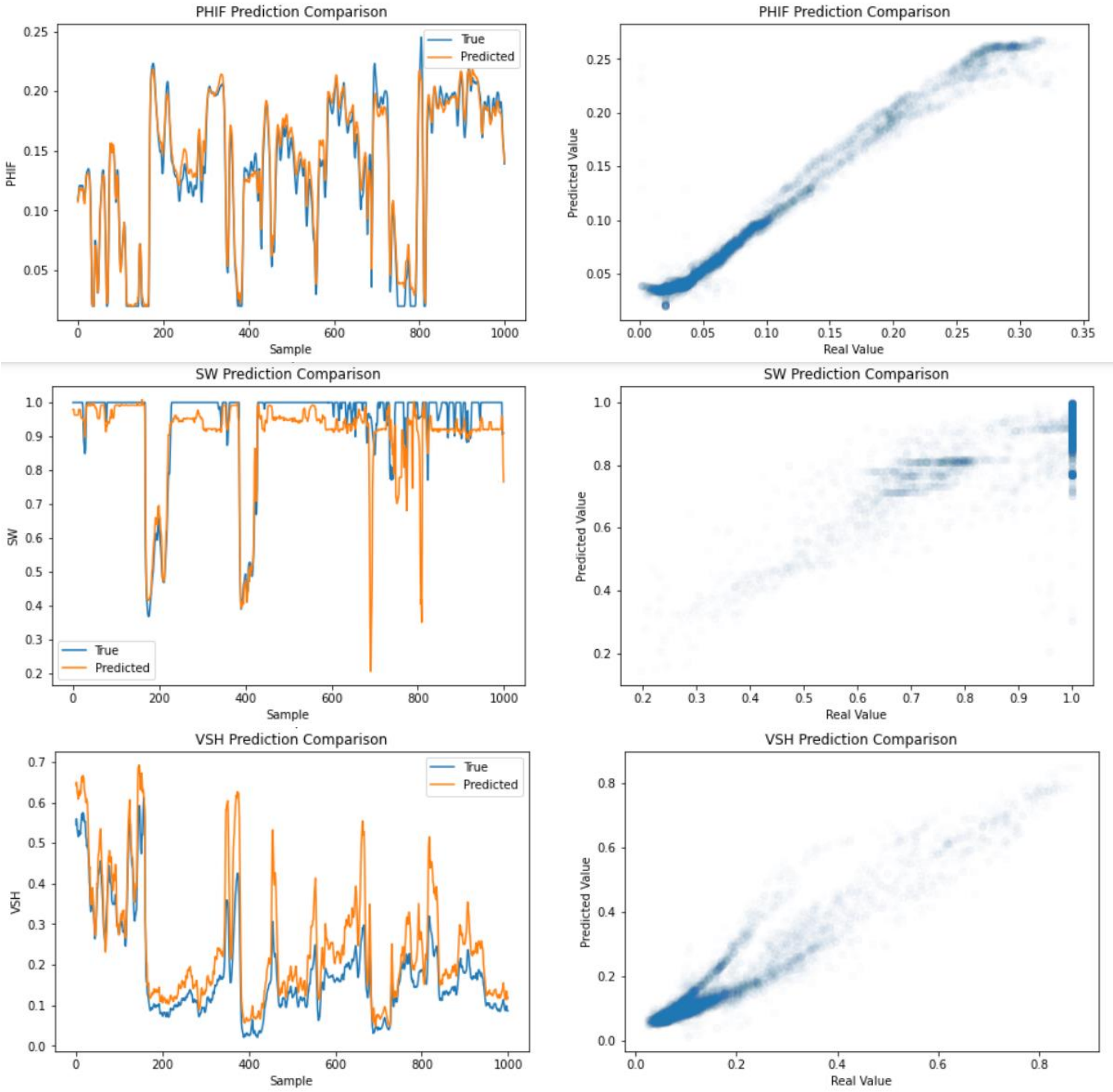

Fig 2: plots that depict the difference between predicted and actual values on validation dataset