

# Industry 4.0

Implementation of AI for the Industry 4.0 in  
a fiber optic sensor production process





# Data Exploration





# Data Exploration

I explored the characteristics and patterns of the data, including its:

- size
- shape
- distribution
- missing values
- data types

Based on my exploration, I have identified potential issues or challenges in the data that required preprocessing steps.



# Data Visualization

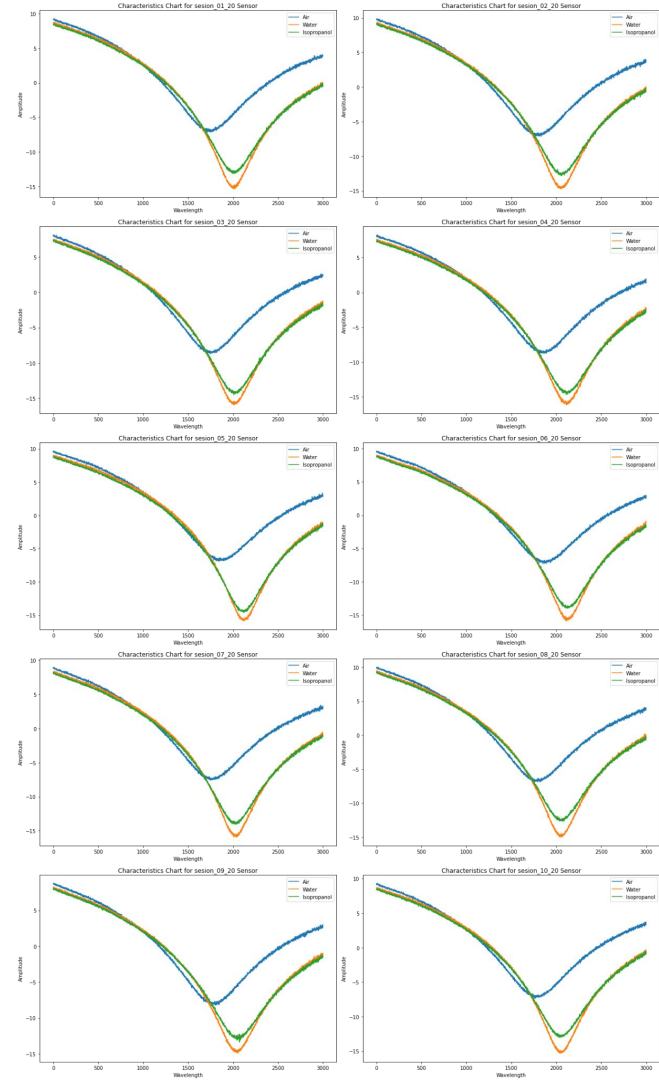
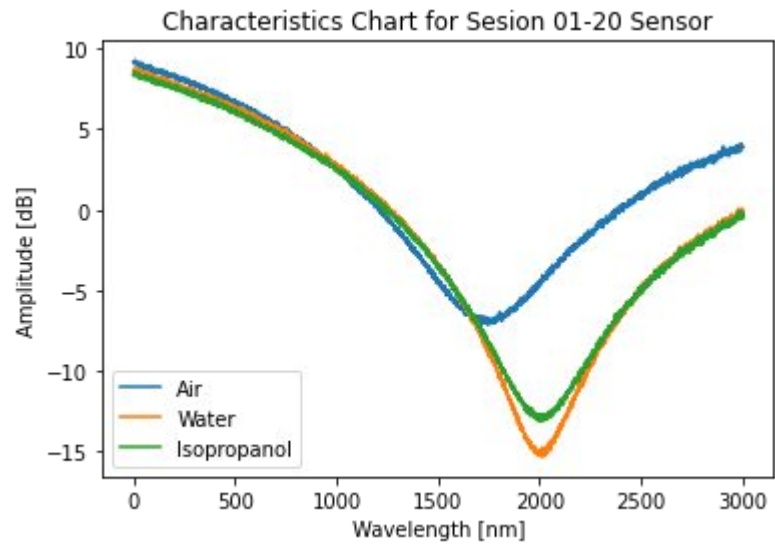


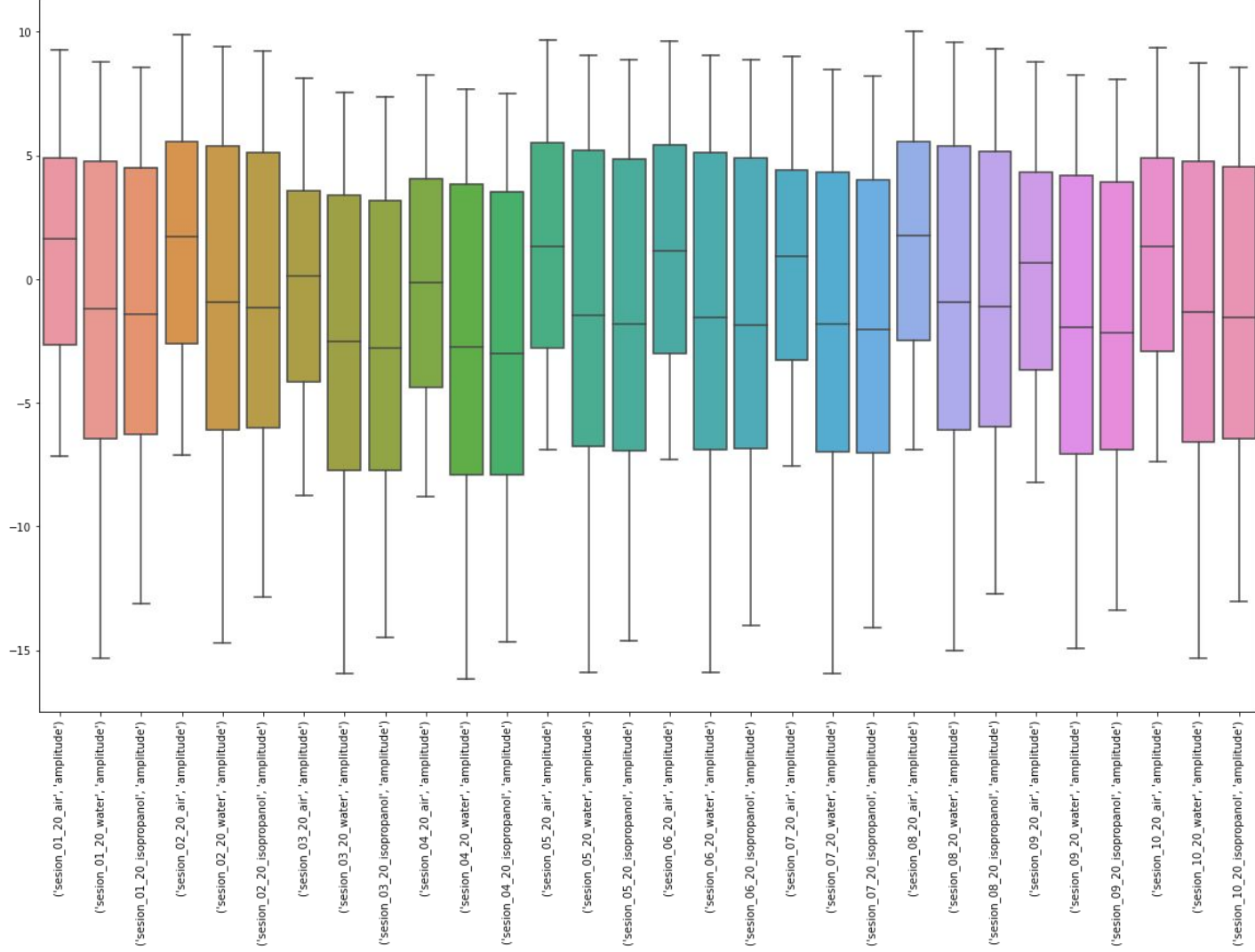


# Data Visualization

Data visualization is a powerful tool to explore and understand patterns, relationships, and trends in data.

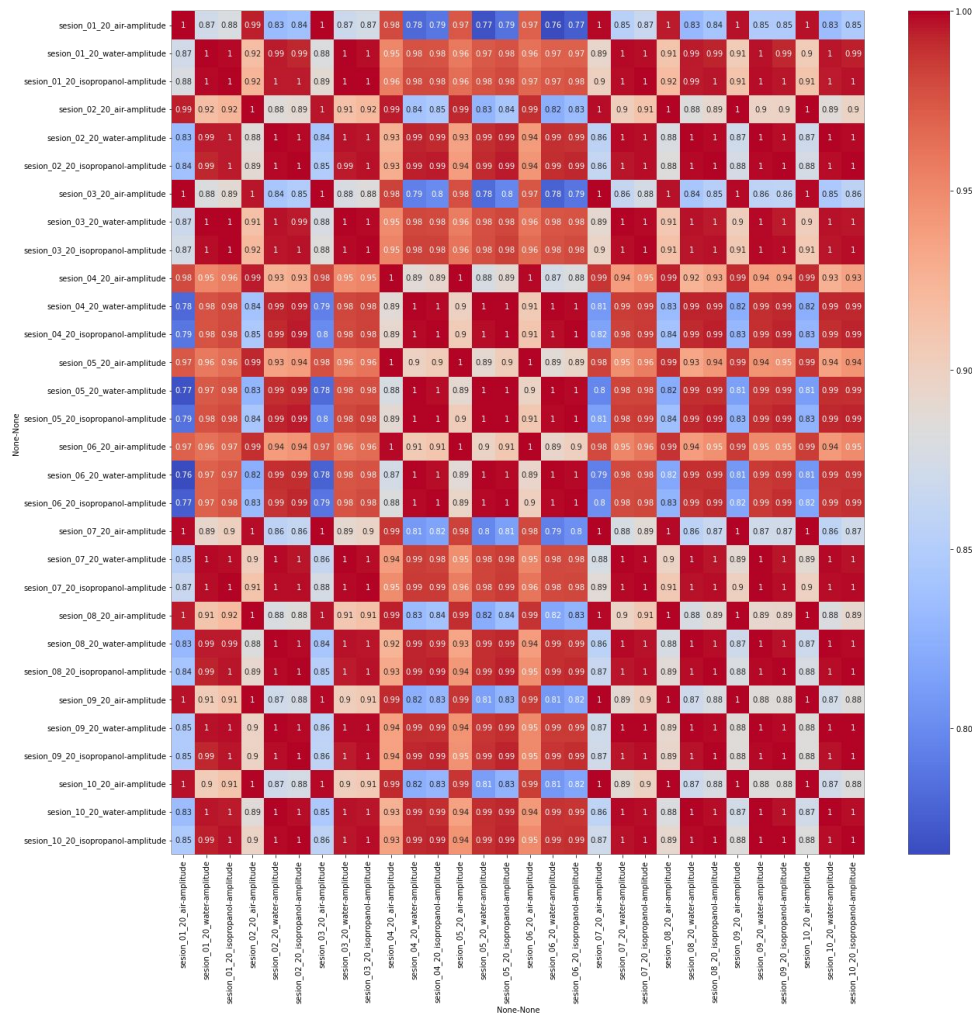
By visualizing data I gain insights into the distributions of variables, correlations between variables, and the overall structure of the data.



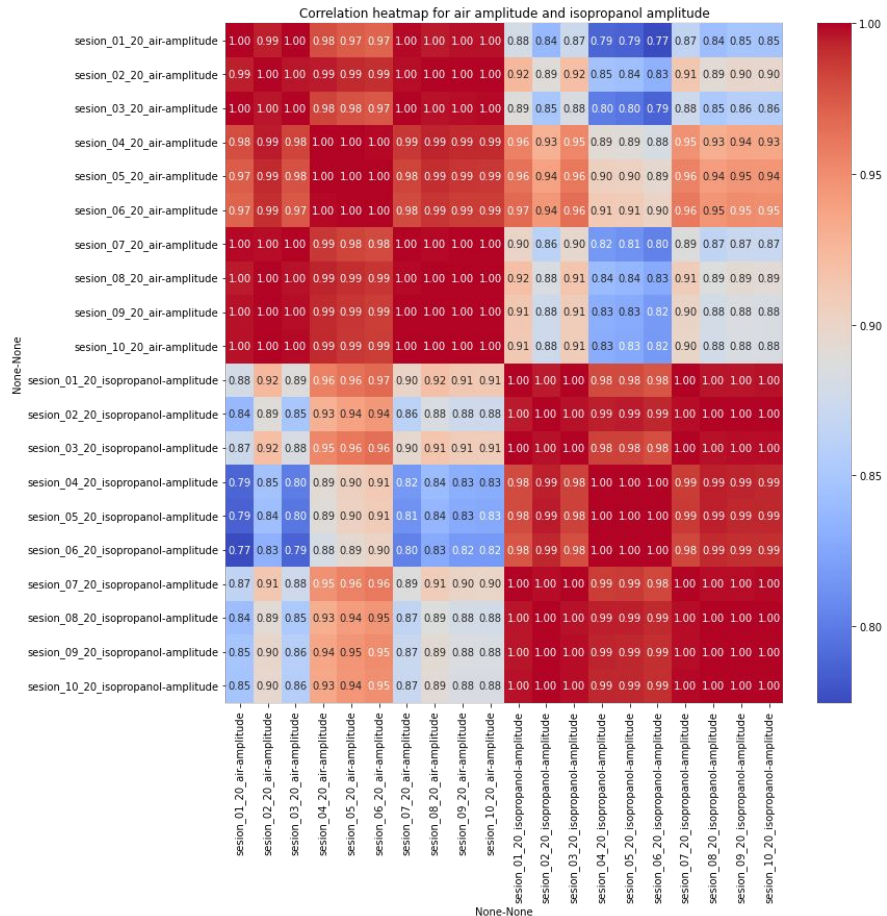
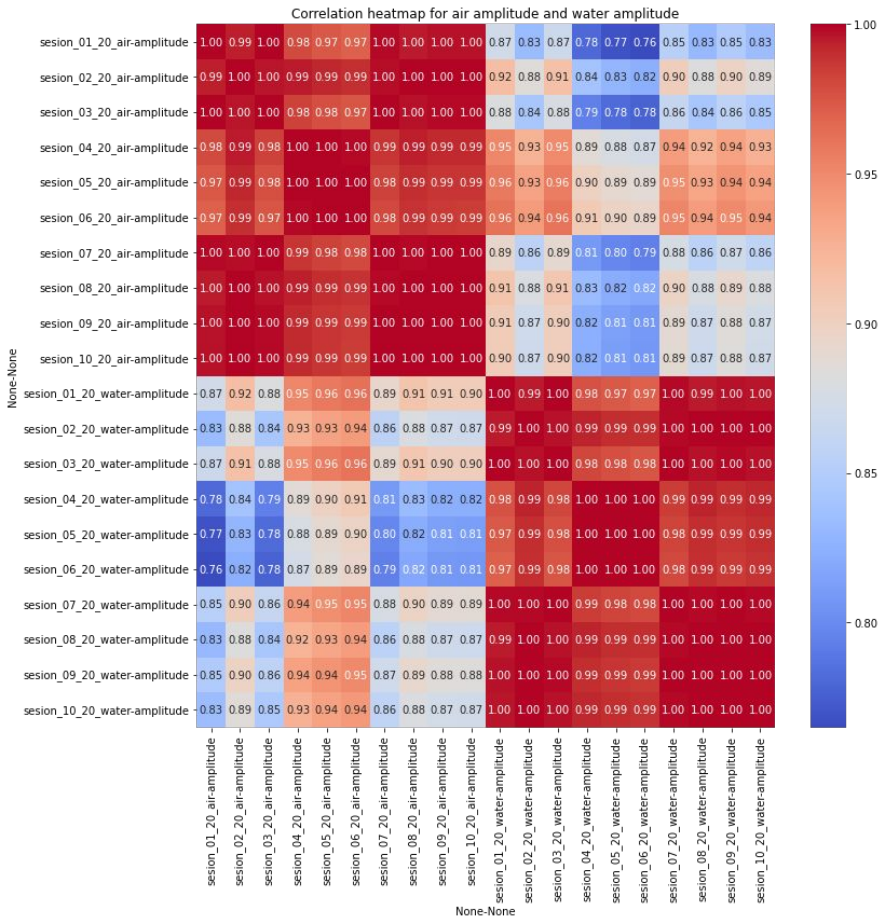




# Correlation









# Data Visualization


A characteristic chart of amplitude and wavelength between air, water, and isopropanol can provide insights into the behavior of these substances when subjected to a certain wavelength of electromagnetic radiation, such as light.

In general, the amplitude of the electromagnetic radiation can be thought of as the intensity of the signal, while the wavelength corresponds to the distance between consecutive peaks or troughs of the wave.

Based on the chart, we can observe that air generally has the lowest amplitude and a relatively shorter wavelength compared to water and isopropanol. This suggests that air may absorb less electromagnetic radiation at a given wavelength compared to water and isopropanol.

On the other hand, both water and isopropanol exhibit higher amplitudes and longer wavelengths compared to air, indicating that these substances may absorb more electromagnetic radiation at a given wavelength. The exact amplitude and wavelength values can depend on the specific wavelength range being considered and the concentration of the substances.

Overall, this type of chart can provide insights into the relative absorbance properties of different substances and can be useful in fields such as spectroscopy and materials science.



# **Data Preprocessing and Feature Engineering**





# Data Preprocessing and Feature Engineering

The data was scaled to ensure that all the features were on the same scale. This was done using standardization technique.

```
df_scaled_standard
array([[ -1.73147375,  1.69132901, -1.73147375, ...,  1.48729147,
        -1.73147375,  1.53990014],
       [ -1.73031943,  1.69579613, -1.73031943, ...,  1.49151516,
        -1.73031943,  1.56115709],
       [ -1.72916511,  1.69505514, -1.72916511, ...,  1.48321242,
        -1.72916511,  1.54177438],
       ...,
       [  1.72916511,  0.5601312 ,  1.72916511, ...,  0.11005189,
        1.72916511,  0.07636772],
       [  1.73031943,  0.58352537,  1.73031943, ...,  0.14568714,
        1.73031943,  0.12874322],
       [  1.73147375,  0.57310914,  1.73147375, ...,  0.16641651,
        1.73147375,  0.11099109]])
```

```
from sklearn.preprocessing import StandardScaler

# create a standard scaler object
standard_scaler = StandardScaler()

# fit and transform the data using the standard
scaler
df_scaled_standard =
standard_scaler.fit_transform(df)
```



# Data Preprocessing and Feature Engineering

Furthermore, feature engineering techniques were used to extract relevant features from the raw data. These techniques include dimensionality reduction using principal component analysis (PCA)

```
▶ df_scaled_standard.shape
```

```
↳ (3001, 60)
```

```
[41] X_pca.shape
```

```
(3001, 2)
```

```
from sklearn.decomposition import PCA

# create a PCA object with the desired
# number of components
pca = PCA(n_components=2)

# fit the PCA object to your data
pca.fit(df_scaled_standard)

# transform your data to the new PCA space
X_pca = pca.transform(df_scaled_standard)
```



# **Building Machine Learning Models**





# Building Machine Learning Models

e.g. Linear Regression

```
# Prepare the data for machine learning
X = df.filter(regex='_ain')
y = df.filter(regex='(water|isopropanol)')
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

pca = PCA(n_components=2)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

# Feature selection with RFE
estimator = LinearRegression()
selector = RFE(estimator, n_features_to_select=5)
selector.fit(X_train_pca, y_train)

X_train_selected = selector.transform(X_train_pca)
X_test_selected = selector.transform(X_test_pca)

# Train a linear regression model
model = LinearRegression()
model.fit(X_train_selected, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test_selected)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
print(f"R2 Score: {r2}")
```

Mean Squared Error: 2.671537845060998  
R2 Score: 0.9428791226589753



# Building Machine Learning Models

During the process of building machine learning models, several types of models were used, including linear regression, decision tree, random forest, and neural networks using the TensorFlow library. These models were used to analyze and make predictions based on the data set being used. Linear regression was used to predict a continuous variable, while decision trees and random forests were used for classification and regression problems. Neural networks were used for more complex problems and were implemented using TensorFlow. Overall, a variety of machine learning models were utilized to find the best fit for the specific problem and data set.





# Preliminary Results





# Preliminary Results

The evaluation of the machine learning models used in predicting the characteristics of sensors in water and isopropanol revealed that the models were able to predict the characteristics with an accuracy ranging from 93% to 99%, depending on the specific model used. For model evaluation, commonly used metrics such as Mean Squared Error (MSE), R-squared ( $R^2$ ), and Mean Absolute Error (MSAE) were used. Also loss and score were also used as evaluation metrics.

Loss is a metric used in machine learning models to measure how well the model is able to approximate the relationship between the input variables and the output variable during training. In TensorFlow, loss is commonly used as the objective function that the optimizer tries to minimize during training.

Score is a metric used to evaluate the performance of a machine learning model on a validation or test set. The specific score metric used depends on the type of problem being solved (classification, regression, etc.) and the evaluation criteria. For example, for classification problems, commonly used score metrics include accuracy, precision, recall, and F1 score, while for regression problems,  $R^2$ , MSE, and MSAE are commonly used as score metrics.

**if you are interested in  
seeing more about this  
project for industry 4.0,  
please feel free to check out  
my project!**

**<https://github.com/M1-KO/Industry-4.0>**