### Tank Ullage Prediction Model

```python

# Import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import matplotlib.pyplot as plt

```

- These lines import the necessary libraries: pandas for data manipulation, scikit-learn for machine learning, and matplotlib for visualization.

```python

# Assuming you have a dataset with relevant features and tank ullage values

# Replace 'dataset.csv' with the actual file path or URL

df = pd.read\_csv('dataset.csv')

```

- Reads a CSV file into a Pandas DataFrame. Replace 'your\_dataset.csv' with the actual path or URL of your dataset.

8

```python

# Assume 'feature1', 'feature2', ... are the relevant features in your dataset

features = df[['feature1', 'feature2', ...]]

ullage = df['ullage']

```

- Selects the relevant features and the target variable (ullage) from the dataset.

```python

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, ullage, test\_size=0.2, random\_state=42)

```

- Splits the dataset into training and testing sets for model evaluation.

```python

# Create a linear regression model

model = LinearRegression()

```

- Initializes a linear regression model.

```python

# Train the model

model.fit(X\_train, y\_train)

```

- Trains the linear regression model on the training data.

```python

# Make predictions on the test set

predictions = model.predict(X\_test)

```

- Uses the trained model to make predictions on the test set.

```python

# Evaluate the model

mse = mean\_squared\_error(y\_test, predictions)

print(f'Mean Squared Error: {mse}')

```

- Evaluates the model performance using mean squared error.

```python

# Visualize predictions vs. actual values

plt.scatter(y\_test, predictions)

plt.xlabel('Actual Ullage')

plt.ylabel('Predicted Ullage')

plt.title('Tank Ullage Prediction')

plt.show()

```

- Visualizes the predicted ullage values against the actual values using a scatter plot.

### Tank Refill Optimization Prediction Model

The second code follows a similar structure:

- Imports necessary libraries.

- Reads the dataset.

- Selects features and target variable.

- Splits the data into training and testing sets.

- Creates a linear regression model.

- Trains the model.

- Makes predictions on the test set.

- Evaluates the model performance.

- Visualizes predicted refill amounts against actual refill amounts.

These codes serve as basic templates, and you would need to customize them based on your specific dataset and requirements.