```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt


# Load your data
data = pd.read_csv('/content/sample_data/leadtime1 .csv')
data.head()
```

```
<ipython-input-19-8ed54982e00e>:2: DtypeWarning: Columns (12) have mixed types. Specify dtype option on import or set low_memory=False.
  data = pd.read_csv('/content/sample_data/leadtime1 .csv')
```

	SH_CD	CATALOG_NO	0	INDENT_DT	MATL_RECEIPT_DT	LEAD_TIME	INSUR_ITEM_IND	AR_ITEM_IND	SPC_ITEM_IND	QUALITY_IND	FTP_IND	FSN_IND	Unnamed: 12
0	4	209110603	BRG.NO.32211 A	01-21-02			N	N	Y	N	N	F	NaN
1	4	2585401002	WHITE CANVAS SHOES SIZE: 10.	06-10-02			N	N	N	N	N	S	NaN
2	4	910995101	;. .	06-10-02			N	N	N	N	N	S	NaN
3	4	1132805505	;V ,...	03-09-02			N	N	Y	N	N	F	NaN
4	4	209130407	/Q	05-03-02			N	N	Y	N	N	F	NaN

```python
# Convert date columns to da                              n
data['INDENT_DT'] = pd.to_da                              errors='coerce')
data['MATL_RECEIPT_DT'] = pc                              :CEIPT_DT'], errors='coerce')
```

```
<ipython-input-20-39339c88405d>:2: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent
  data['INDENT_DT'] = pd.to_datetime(data['INDENT_DT'], errors='coerce')
<ipython-input-20-39339c88405d>:3: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateutil`. To ensure parsing is consistent
  data['MATL_RECEIPT_DT'] = pd.to_datetime(data['MATL_RECEIPT_DT'], errors='coerce')
```

```python
# Drop rows with NaT in date columns
data.dropna(subset=['INDENT_DT', 'MATL_RECEIPT_DT'], inplace=True)
print(data.head())


# Encode categorical variables
label_encoders = {}
categorical_cols = data.select_dtypes(include=['object']).columns


for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col].astype(str))
    label_encoders[col] = le
```

```
# Define features and target
X = data.drop(columns=['LEAD_TIME', 'INDENT_DT', 'MATL_RECEIPT_DT'])
y = data['LEAD_TIME']
print(X.head())  # Check the features
print(y.head())  # Check the target
```

```
   SH_CD  CATALOG_NO      0  INSUR_ITEM_IND  AR_ITEM_IND  SPC_ITEM_IND  \
0      4   209110603   9072               0            0             1
1      4  2585401002  37182               0            0             0
2      4   910995101  27164               0            0             0
3      4  1132805505  35821               0            0             1
4      4   209130407   7441               0            0             1

   QUALITY_IND  FTP_IND  FSN_IND  Unnamed: 12
0            0        0        0            0
1            0        0        2            0
2            0        0        2            0
3            0        0        0            0
4            0        0        0            0
0    411
1     44
2     44
3    332
4    201
Name: LEAD_TIME, dtype:
```

```
# Split the data into traini                          :a (30%)
X_train, X_temp, y_train, y_                          , y, test_size=0.3, random_state=42)



# Split the remaining data i                          idation (10%)
X_test, X_val, y_test, y_val = train_test_split(X_temp, y_temp, test_size=1/3, random_state=42)


# Train a RandomForestRegressor model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

```
# Predict on test and validation sets
y_test_pred = model.predict(X_test)
y_val_pred = model.predict(X_val)


# Calculate errors
test_errors = abs(y_test - y_test_pred)
val_errors = abs(y_val - y_val_pred)
```

```
# Combine test and validation predictions and errors
predictions = pd.DataFrame({
    'Catalog No': pd.concat([X_test['CATALOG_NO'].reset_index(drop=True), X_val['CATALOG_NO'].reset_index(drop=True)]),
    'True Lead Time': pd.concat([y_test.reset_index(drop=True), y_val.reset_index(drop=True)]),
    'Predicted Lead Time': pd.concat([pd.Series(y_test_pred).reset_index(drop=True), pd.Series(y_val_pred).reset_index(drop=True)]),
    'Error': pd.concat([pd.Series(test_errors).reset_index(drop=True), pd.Series(val_errors).reset_index(drop=True)])
})

print(predictions.head())
```

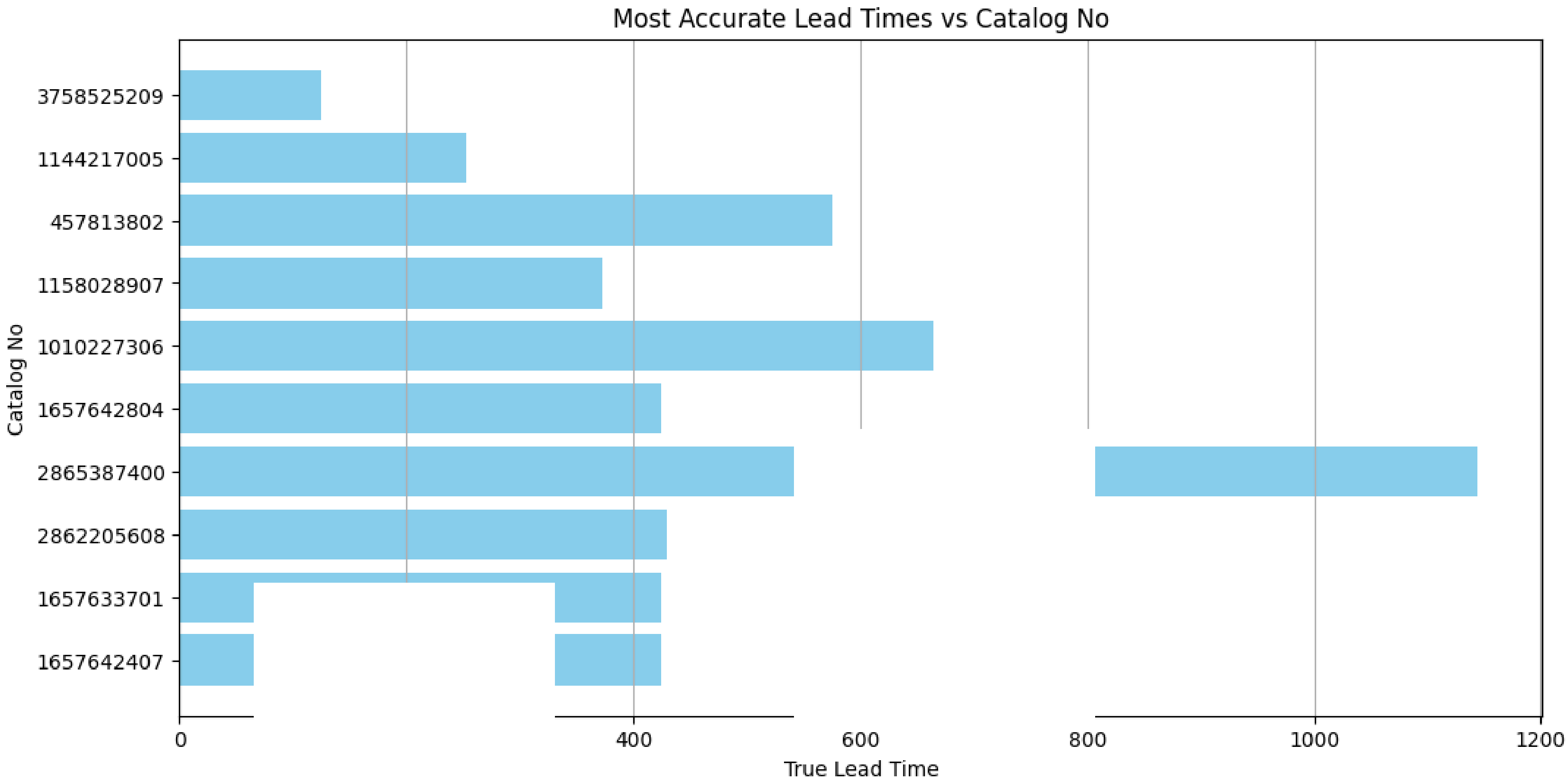
```
   Catalog No  True Lead Time  Predicted Lead Time       Error
0  3758318601             413           295.410500  117.589500
1   483225404             494           623.671786  129.671786
2   201121209            1073           712.860000  360.140000
3  1136838007             539           511.613333   27.386667
4   816139405             701           606.674987   94.325013
```

```
# Identify the most accurate items (smallest errors)
most_accurate_items = predictions.nsmallest(10, 'Error')
print(most_accurate_items)
```

```
       Catalog No  True                          d Time  Error
1204   3758525209                                 125.0    0.0
3321   1144217005                                 253.0    0.0
4479    457813802                                 576.0    0.0
9629   1158028907                                 373.0    0.0
10675  1010227306                                 665.0    0.0
11735  1657642804                                 424.0    0.0
12871  2865387400                                1144.0    0.0
570    2862205608                                 430.0    0.0
624    1657633701                                 424.0    0.0
3614   1657642407                                 424.0    0.0
```

```
# Select the most accurate items
most_accurate_items = predictions.nsmallest(10, 'Error')

# Plot the most accurate items
plt.figure(figsize=(12, 6))
plt.barh(most_accurate_items['Catalog No'].astype(str), most_accurate_items['True Lead Time'], color='skyblue')
plt.xlabel('True Lead Time')
plt.ylabel('Catalog No')
plt.title('Most Accurate Lead Times vs Catalog No')
plt.gca().invert_yaxis()  # To display the lowest values at the top
plt.grid(axis='x')
plt.show()
```


Most Accurate Lead Times vs Catalog No
Catalog No
3758525209
1144217005
457813802
1158028907
1010227306
1657642804
2865387400
2862205608
1657633701
1657642407
0
400
600
800
1000
1200
True Lead Time