

```
# Time Series Forecasting using Linear  
Regression  
  
import pandas as pd  
from sklearn.linear_model import  
LinearRegression  
  
# Load data  
data = pd.read_csv("timeseries.csv")  
  
# Create lag feature  
data["lag1"] = data["value"].shift(1)  
data = data.dropna()  
  
# Features & target  
X = data[["lag1"]]  
y = data["value"]  
  
# Train model  
model = LinearRegression()  
model.fit(X, y)  
  
# Predict next values  
predictions = model.predict(X)  
  
print(predictions[:5])
```

```
import pandas as pd
from sklearn.linear_model import
LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import
train_test_split

# Load dataset (must contain columns: date,
value)
data = pd.read_csv("timeseries.csv")

# Convert date column
data['date'] = pd.to_datetime(data['date'])
data = data.sort_values('date')

# Create lag features
data['lag1'] = data['value'].shift(1)
data['lag2'] = data['value'].shift(2)

# Remove null rows created by lagging
data = data.dropna()

# Define features and target
X = data[['lag1', 'lag2']]
y = data['value']

# Train-test split (no shuffle for time
series)
X_train, X_test, y_train, y_test =
train_test_split(
    X, y, test_size=0.2, shuffle=False
)

# Train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict
predictions = model.predict(X_test)

# Evaluate
mse = mean_squared_error(y_test, predictions)
print("MSE:", mse)

# Show sample predictions
print(predictions[:10])
```