In [2]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, lag
from pyspark.sql.window import Window
from pyspark.ml.feature import VectorAssembler, StandardScaler
from pyspark.ml.regression import RandomForestRegressor, RandomForestRegress
from pyspark.ml.evaluation import RegressionEvaluator
from pyspark.ml.tuning import ParamGridBuilder, TrainValidationSplit
from pyspark.sql.types import IntegerType, FloatType, DoubleType, LongType,
import findspark
```

In [3]:
```
# Initialize Spark session
spark = SparkSession.builder.master("local[*]").appName("FeatureEngineeringA
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLog
Level(newLevel).
24/11/06 14:02:35 WARN NativeCodeLoader: Unable to load native-hadoop librar
y for your platform... using builtin-java classes where applicable
24/11/06 14:02:35 WARN Utils: Service 'SparkUI' could not bind on port 4040.
Attempting port 4041.
24/11/06 14:02:35 WARN Utils: Service 'SparkUI' could not bind on port 4041.
Attempting port 4042.
24/11/06 14:02:35 WARN Utils: Service 'SparkUI' could not bind on port 4042.
Attempting port 4043.
```

In [4]:
```
# Load the data into a Spark DataFrame
df = spark.read.csv("/Users/sarabjotsingh/Downloads/Feature_engineering.csv"
df = df.select([col(c).alias(c.replace(" ", "_")) for c in df.columns])  # R
```

In [5]:
```
# Define a window specification based on the 'open' column as ordering
window = Window.orderBy("open")
```

In [6]:
```
# List of features for which we want to create lagged columns
selected_features = ['up', 'prev_diff', 'daydiff']
lags = [1, 2, 3]
```

In [7]:
```
# Generate Lagged Features
for feature in selected_features:
    for lag_value in lags:
        lagged_col_name = f"{feature}_lag{lag_value}"
        df = df.withColumn(lagged_col_name, lag(col(feature), lag_value).ove

df = df.dropna()  # Drop rows with null values introduced by lagging
```

In [8]:
```
# Identify numeric columns for VectorAssembler
numeric_columns = [field.name for field in df.schema.fields if isinstance(fi

# Check for unsupported columns
print("Numeric Columns:", numeric_columns)
```

```
Numeric Columns: ['open', 'high', 'low', 'close', 'volume', 'dividends', 'st
ock_splits', 'capital_gains', 'previous_day_close', 'rolling_dividends', 'ro
lling_splits', 'adj_close', 'prev_close', 'maxdiff', 'daydiff', 'RSI', 'up',
'dn', 'EMA12', 'EMA26', 'MACD', 'SignalLine', 'prev_diff', 'change_tomorro
w', 'up_lag1', 'up_lag2', 'up_lag3', 'prev_diff_lag1', 'prev_diff_lag2', 'pr
ev_diff_lag3', 'daydiff_lag1', 'daydiff_lag2', 'daydiff_lag3']
```

In [9]:
```
# Assemble features using only numeric columns
assembler = VectorAssembler(inputCols=numeric_columns, outputCol="features")
df = assembler.transform(df)
```

In [10]:
```
# Check if the 'features' column is created successfully
print("DataFrame schema after VectorAssembler:")
df.printSchema()
print("DataFrame columns after VectorAssembler:", df.columns)
```

```
DataFrame schema after VectorAssembler:
root
 |-- date: timestamp (nullable = true)
 |-- open: double (nullable = true)
 |-- high: double (nullable = true)
 |-- low: double (nullable = true)
 |-- close: double (nullable = true)
 |-- volume: integer (nullable = true)
 |-- dividends: double (nullable = true)
 |-- stock_splits: double (nullable = true)
 |-- symbol: string (nullable = true)
 |-- capital_gains: double (nullable = true)
 |-- previous_day_close: double (nullable = true)
 |-- rolling_dividends: double (nullable = true)
 |-- rolling_splits: double (nullable = true)
 |-- adj_close: double (nullable = true)
 |-- prev_close: double (nullable = true)
 |-- maxdiff: double (nullable = true)
 |-- daydiff: double (nullable = true)
 |-- RSI: double (nullable = true)
 |-- up: double (nullable = true)
 |-- dn: double (nullable = true)
 |-- EMA12: double (nullable = true)
 |-- EMA26: double (nullable = true)
 |-- MACD: double (nullable = true)
 |-- SignalLine: double (nullable = true)
 |-- prev_diff: double (nullable = true)
 |-- change_tomorrow: integer (nullable = true)
 |-- up_lag1: double (nullable = true)
 |-- up_lag2: double (nullable = true)
 |-- up_lag3: double (nullable = true)
 |-- prev_diff_lag1: double (nullable = true)
 |-- prev_diff_lag2: double (nullable = true)
 |-- prev_diff_lag3: double (nullable = true)
 |-- daydiff_lag1: double (nullable = true)
 |-- daydiff_lag2: double (nullable = true)
 |-- daydiff_lag3: double (nullable = true)
 |-- features: vector (nullable = true)

DataFrame columns after VectorAssembler: ['date', 'open', 'high', 'low', 'cl
ose', 'volume', 'dividends', 'stock_splits', 'symbol', 'capital_gains', 'pre
vious_day_close', 'rolling_dividends', 'rolling_splits', 'adj_close', 'prev_
close', 'maxdiff', 'daydiff', 'RSI', 'up', 'dn', 'EMA12', 'EMA26', 'MACD', '
SignalLine', 'prev_diff', 'change_tomorrow', 'up_lag1', 'up_lag2', 'up_lag
3', 'prev_diff_lag1', 'prev_diff_lag2', 'prev_diff_lag3', 'daydiff_lag1', 'd
aydiff_lag2', 'daydiff_lag3', 'features']
```

In [11]:
```
# Select the relevant columns for the model
df = df.select("features", "RSI")
```

In [12]:
```
# Split the data into training and testing sets
train_df, test_df = df.randomSplit([0.7, 0.3], seed=42)
```

In [13]:
```
# Standardize the feature vector
scaler = StandardScaler(inputCol="features", outputCol="scaled_features")
scaler_model = scaler.fit(train_df)
```

```
24/11/06 14:02:41 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:41 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:41 WARN SparkStringUtils: Truncated the string representation
of a plan since it was too large. This behavior can be adjusted by setting '
spark.sql.debug.maxToStringFields'.
24/11/06 14:02:42 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:42 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
```

In [14]:
```
# Transform the training and testing sets
train_df = scaler_model.transform(train_df).select("scaled_features", "RSI")
test_df = scaler_model.transform(test_df).select("scaled_features", "RSI").w
```

In [15]:
```
# Check the final DataFrame after scaling
print("Train DataFrame schema after scaling:")
train_df.printSchema()
print("Test DataFrame schema after scaling:")
test_df.printSchema()
```

```
Train DataFrame schema after scaling:
root
 |-- features: vector (nullable = true)
 |-- RSI: double (nullable = true)

Test DataFrame schema after scaling:
root
 |-- features: vector (nullable = true)
 |-- RSI: double (nullable = true)
```

In [16]:
```
# Load the model if saved previously
model_save_path = "/Users/sarabjotsingh/Downloads/rf_model"
try:
    loaded_model = RandomForestRegressionModel.load(model_save_path)
    print("Model loaded successfully.")
except Exception as e:
    print("Model loading failed:", e)
```

```
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.spark.util.SizeEstimator$ (
file:/opt/homebrew/Cellar/apache-spark/3.5.3/libexec/jars/spark-core_2.12-3.
5.3.jar) to field java.math.BigInteger.mag
WARNING: Please consider reporting this to the maintainers of org.apache.spa
rk.util.SizeEstimator$
WARNING: Use --illegal-access=warn to enable warnings of further illegal ref
lective access operations
WARNING: All illegal access operations will be denied in a future release
Model loaded successfully.
```

In [17]:
```
# Initialize Random Forest Regressor
rf = RandomForestRegressor(labelCol="RSI", featuresCol="features")
```

In [18]:
```
# Simplified parameter grid for quick testing
paramGrid = (ParamGridBuilder()
             .addGrid(rf.numTrees, [50])          # Only one setting for num
             .addGrid(rf.maxDepth, [5])           # Only one setting for max
             .build())
```

In [19]:
```
# Define the evaluator using Mean Squared Error
evaluator = RegressionEvaluator(labelCol="RSI", predictionCol="prediction",
```

In [20]:
```
# Use TrainValidationSplit for grid search validation with reduced training
tvs = TrainValidationSplit(estimator=rf,
                           estimatorParamMaps=paramGrid,
                           evaluator=evaluator,
                           trainRatio=0.8)  # 80% for training, 20% for vali
```

In [21]:
```
# Fit the TrainValidationSplit model to the training data
tvs_model = tvs.fit(train_df)
```

```
24/11/06 14:02:48 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:48 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:48 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:48 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:48 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:48 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:50 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:02:50 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:15 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:15 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:18 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:18 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:18 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:19 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:19 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
```

In [22]:

```
# Evaluate the best model from TrainValidationSplit
best_model = tvs_model.bestModel
predictions = best_model.transform(test_df)
mse = evaluator.evaluate(predictions)
print(f"Best Model Mean Squared Error (MSE) after Hyperparameter Tuning: {ms
```

```
24/11/06 14:03:50 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:50 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:50 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:51 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:51 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
[Stage 68:>                                                         (0 + 1)
/ 1]
Best Model Mean Squared Error (MSE) after Hyperparameter Tuning: 31.58595796
7459073
```

In [23]:

```
# Assuming tvs_model is your fitted TrainValidationSplit model
best_model = tvs_model.bestModel

# Print the best parameters found
print("Best Params:")
print(f"Num Trees: {best_model.getNumTrees}")  # Corrected access
print(f"Max Depth: {best_model.getMaxDepth}")  # Corrected access
print(f"Min Instances Per Node: {best_model.getMinInstancesPerNode}")  # Cor
```

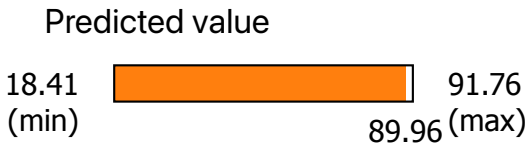
```
Best Params:
Num Trees: 50
Max Depth: <bound method _DecisionTreeParams.getMaxDepth of RandomForestRegr
essionModel: uid=RandomForestRegressor_71b86c1dbb24, numTrees=50, numFeature
s=33>
Min Instances Per Node: <bound method _DecisionTreeParams.getMinInstancesPer
Node of RandomForestRegressionModel: uid=RandomForestRegressor_71b86c1dbb24,
numTrees=50, numFeatures=33>
```

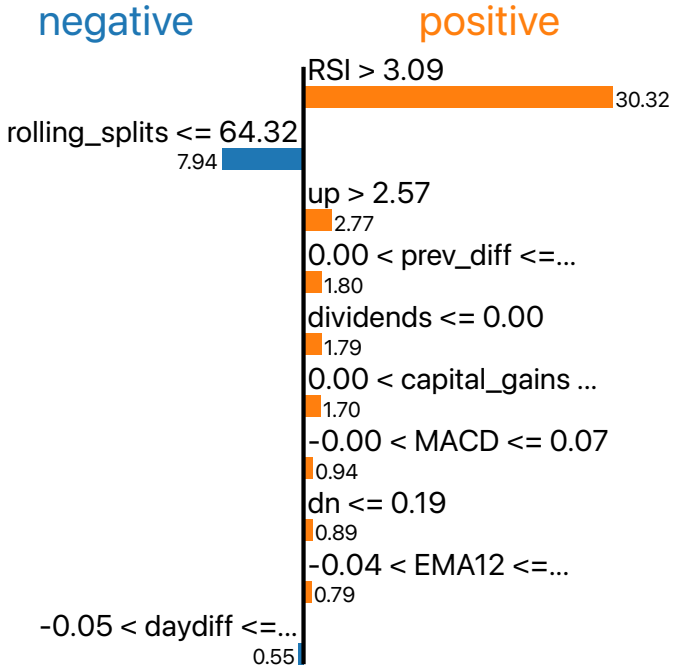
In [24]:

```
# Save the model to a specified path
#model_save_path = "/Users/sarabjotsingh/Downloads/best_rf_model"  # Specify
#best_model.save(model_save_path)
#
#print(f"Model saved at: {model_save_path}")
```

In [25]:
```
import numpy as np
from lime.lime_tabular import LimeTabularExplainer
from pyspark.ml.linalg import Vectors
from pyspark.ml import PipelineModel

# Prepare the data
# Use a random sample or a specific instance from the test set to explain
sample_data = test_df.select("features").first()
sample_data = sample_data["features"]

# Convert the data into a format that LIME understands (NumPy array)
sample_data_np = np.array(sample_data.toArray()).reshape(1, -1)

# Create a function to predict using the model (used by LIME)
def predict_fn(X):
    # Convert X into a DataFrame with a "features" column
    features_df = spark.createDataFrame([(Vectors.dense(x),) for x in X], ["
    predictions = best_model.transform(features_df)
    return np.array(predictions.select("prediction").collect()).flatten()

# Initialize the LIME explainer
explainer = LimeTabularExplainer(
    training_data=np.array(train_df.select("features").rdd.map(lambda row: r
    mode='regression',
    feature_names=numeric_columns,  # Assuming these are your feature names
    class_names=["RSI"],  # The name of the target variable (label)
    discretize_continuous=True
)

# Explain a specific instance (this example uses a random instance)
explanation = explainer.explain_instance(sample_data_np[0], predict_fn, num_

# Visualize the explanation
explanation.show_in_notebook()
```

```
24/11/06 14:03:56 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:56 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:56 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:57 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:57 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:58 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:58 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:58 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:59 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
24/11/06 14:03:59 WARN WindowExec: No Partition Defined for Window operatio
n! Moving all data to a single partition, this can cause serious performance
degradation.
```





Feature	Value
RSI	4.23
rolling_splits	64.32
up	2.76
prev_diff	0.02
dividends	0.00
capital_gains	0.02
MACD	0.01
dn	0.00
EMA12	-0.04
daydiff	0.00