Observation and Processes

Strategic Importance:

- Accurate forecasting of gas prices is critical for strategic planning, budgeting, and risk management in sectors dependent on energy costs.
- Insights derived from the model can inform procurement strategies, pricing adjustments, and long-term investment decisions.

Variables Used:

- Natural Gas Prices
- Crude Oil Prices
- General Gasoline Prices
- RBOB Gasoline Prices
- Heating Oil Prices
- Diesel Prices
- Kerosene Prices
- Propane Prices
- Oil Stock Levels
- Drilling Activity
- Gas Production Volumes
- Gas Consumption Volumes
- Gas Storage Levels
- Gas Import Volumes
- Gas Import Prices

Derived Features:

1. Price Ratios

- **Ratios of different fuel prices** such as crude oil price to gas prices, or diesel price to heating oil price. These ratios can reveal relative pricing trends and substitutions that might occur depending on price movements.

- **Example**: **diesel_to_heating_oil_ratio = diesel price / heating oil price**
- **Reasoning**: Price ratios can help capture cross-dependencies between different fuel types which might be influenced by similar market forces or seasonal demand changes.

2. Historical Volatility

- **30-day or 60-day rolling volatility of prices** for commodities like crude oil, gas, and others.
- **Example**: **rolling_volatility_crude_oil = crude oil price.rolling(window=30).std()**
- **Reasoning**: Volatility measures how much the price of a commodity fluctuates over a period and can be indicative of market uncertainty or potential future price movements.

3. Price Differences

- **Differences between related commodity prices**, such as the spread between crude oil prices and various types of gas prices.
- **Example**: **crude_vs_gas_price_diff = crude oil price - conventional gas price**
- **Reasoning**: Price differences or spreads between similar commodities can indicate relative supply and demand dynamics, refining margins, or logistical costs.

4. Seasonal Indicators

- **Cyclical features derived from the week number**, such as sine and cosine transformations to capture seasonality effects throughout the year.
- **Example**: **sin_week = np.sin(2 * np.pi * week number / 52)**, **cos_week = np.cos(2 * np.pi * week number / 52)**
- **Reasoning**: Energy consumption patterns are highly seasonal; transforming week numbers into cyclical features allows the model to better learn and predict seasonal impacts on prices and demands.

5. Demand-Supply Balance

- **Ratios or differences between production, consumption, storage, and import volumes** to reflect the demand-supply balance in the market.
- **Example**: **supply_demand_ratio = (gas production + gas import volume) / gas consumption**
- **Reasoning**: A higher ratio or a significant change in the ratio can signal shifts in market equilibrium, potentially impacting prices and availability.

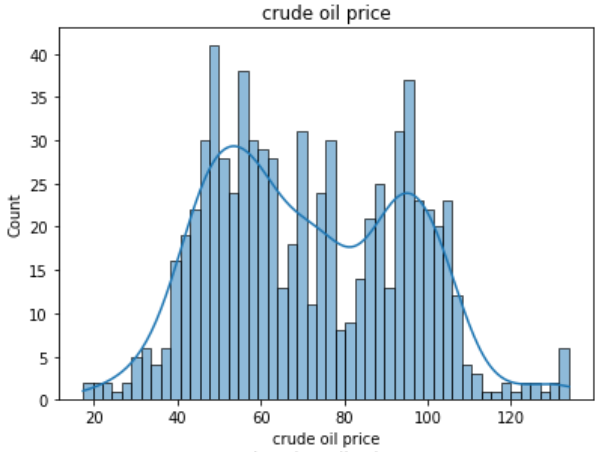
6. Import Cost

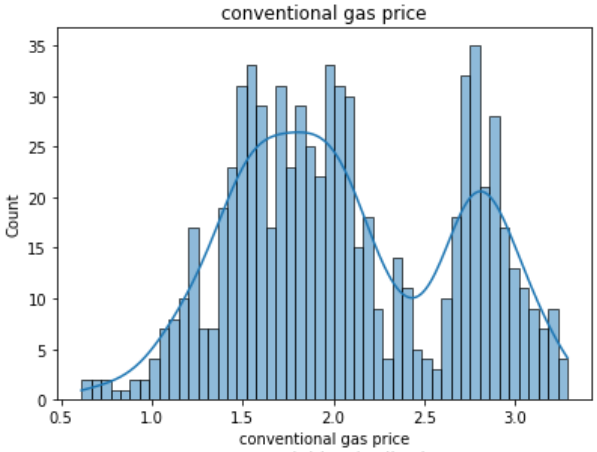
- **Total cost of imports** calculated by multiplying the volume by the price of imports.
- **Example**: **total_import_cost = gas import volume * gas import price**
- **Reasoning**: This feature can indicate the economic burden or benefit derived from importing versus local production, influencing national energy policies and market prices.

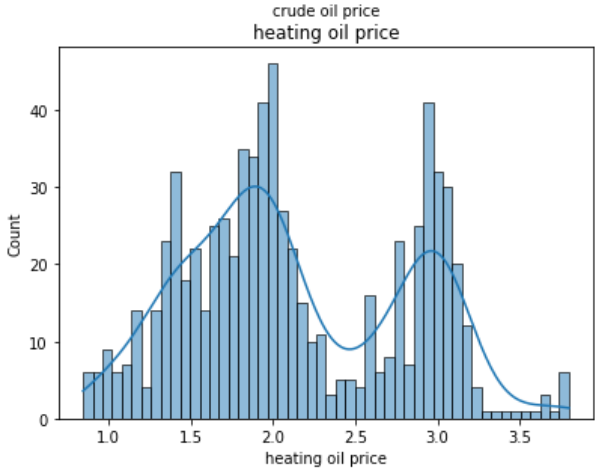
7. Inventory Turnover

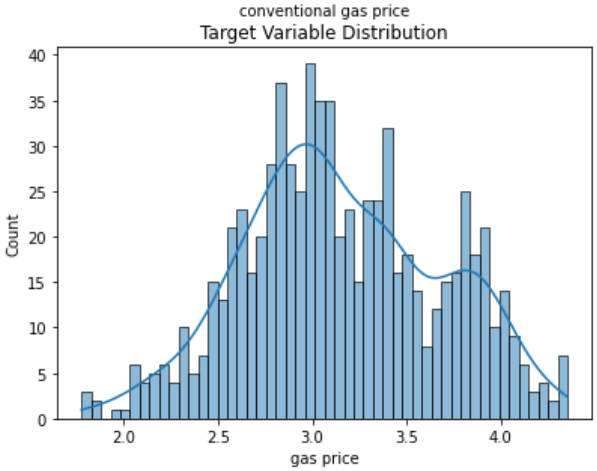
- **Ratio of consumption to storage** to gauge how quickly the stored gas is being used.
- **Example**: **inventory_turnover = gas consumption / gas storage**
- **Reasoning**: A lower ratio might indicate oversupply or decreased demand, while a higher ratio could signal tight market conditions or increased demand.

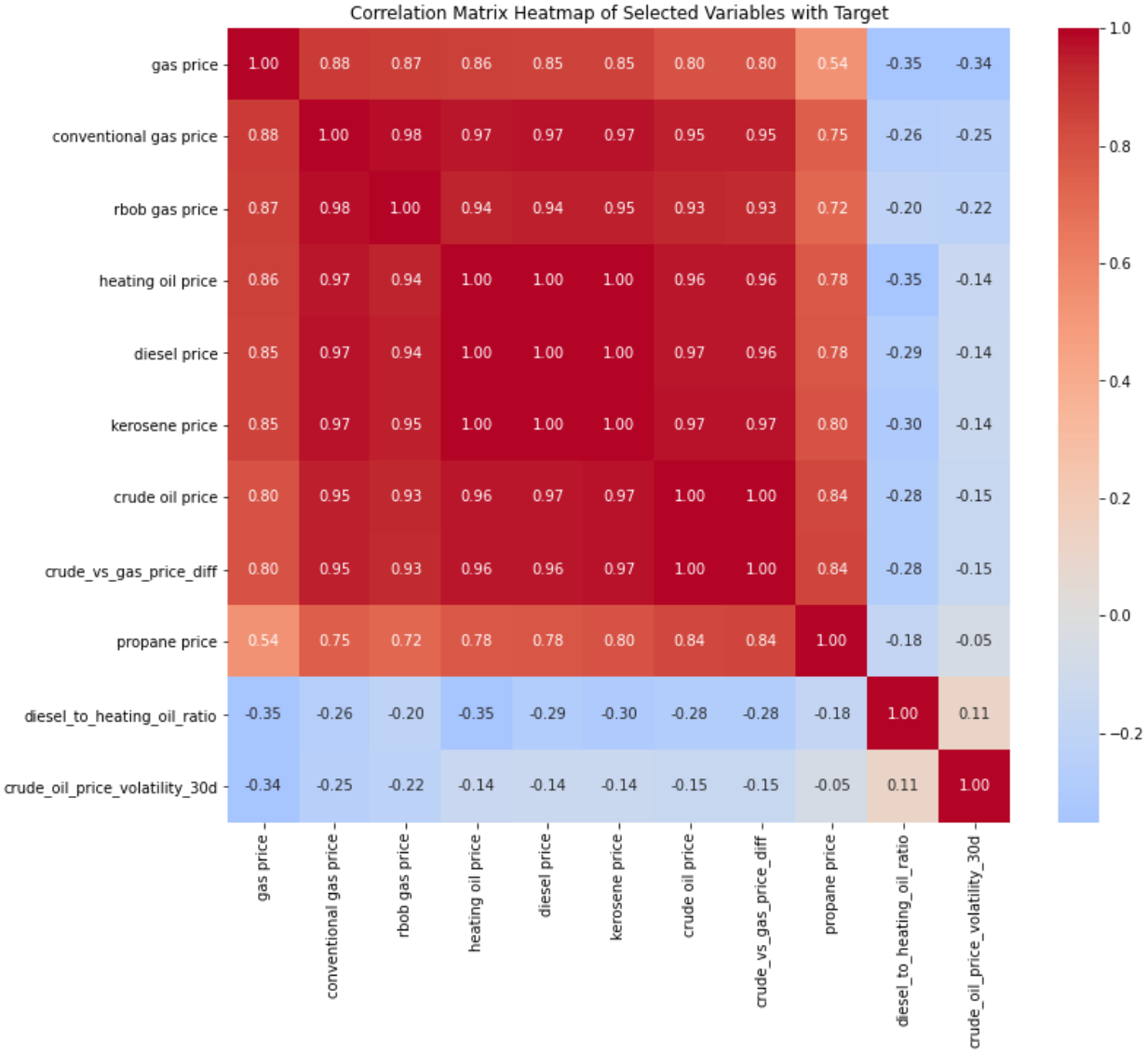
EDA and Correlational Analysis:


crude oil price
Count
crude oil price

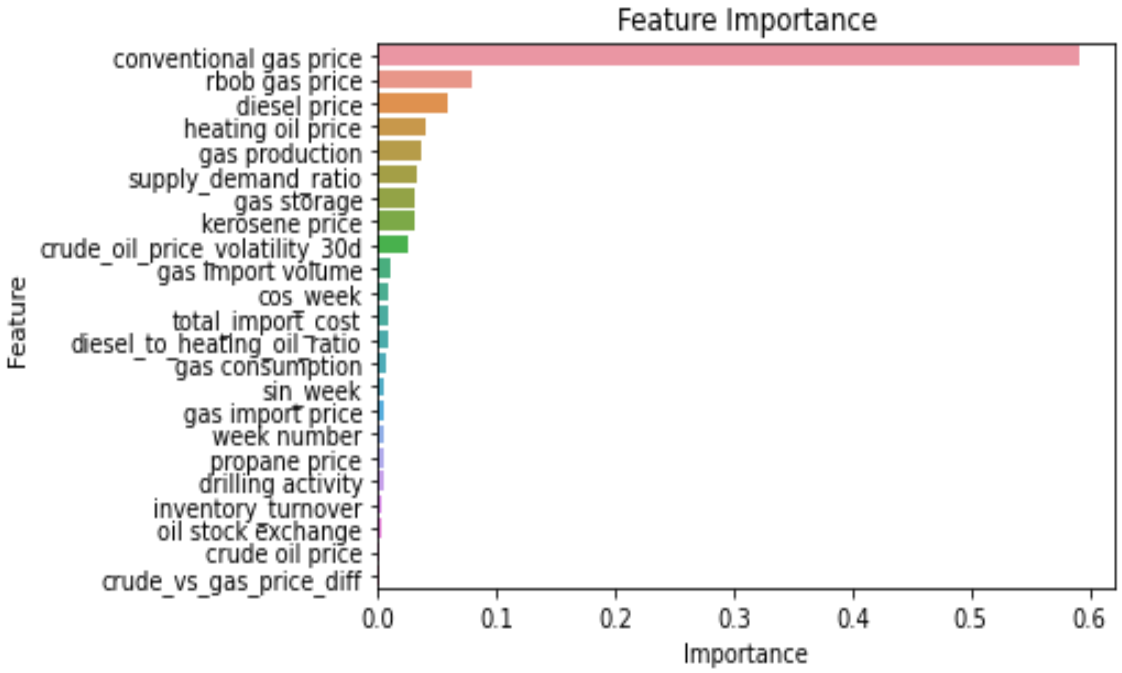

conventional gas price
Count
conventional gas price

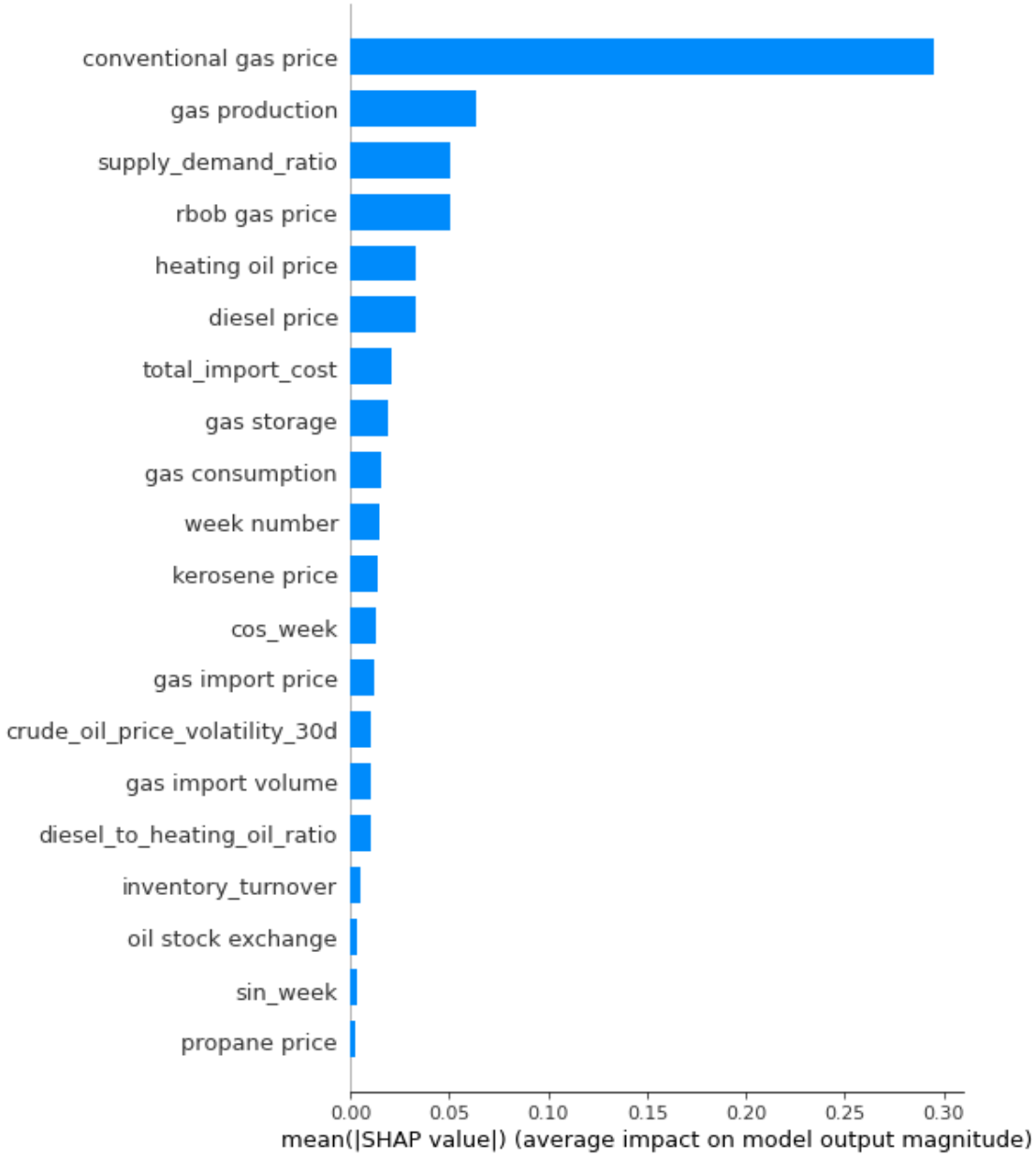

heating oil price
Count
heating oil price


Target Variable Distribution
Count
gas price


Correlation Matrix Heatmap of Selected Variables with Target
gas price 1.00 0.88 0.87 0.86 0.85 0.85 0.80 0.80 0.54 -0.35 -0.34
conventional gas price 0.88 1.00 0.98 0.97 0.97 0.97 0.95 0.95 0.75 -0.26 -0.25
rbob gas price 0.87 0.98 1.00 0.94 0.94 0.95 0.93 0.93 0.72 -0.20 -0.22
heating oil price 0.86 0.97 0.94 1.00 1.00 1.00 0.96 0.96 0.78 -0.35 -0.14
diesel price 0.85 0.97 0.94 1.00 1.00 1.00 0.97 0.96 0.78 -0.29 -0.14
kerosene price 0.85 0.97 0.95 1.00 1.00 1.00 0.97 0.97 0.80 -0.30 -0.14
crude oil price 0.80 0.95 0.93 0.96 0.97 0.97 1.00 1.00 0.84 -0.28 -0.15
crude_vs_gas_price_diff 0.80 0.95 0.93 0.96 0.96 0.97 1.00 1.00 0.84 -0.28 -0.15
propane price 0.54 0.75 0.72 0.78 0.78 0.80 0.84 0.84 1.00 -0.18 -0.05
diesel_to_heating_oil_ratio -0.35 -0.26 -0.20 -0.35 -0.29 -0.30 -0.28 -0.28 -0.18 1.00 0.11
crude_oil_price_volatility_30d -0.34 -0.25 -0.22 -0.14 -0.14 -0.14 -0.15 -0.15 -0.05 0.11 1.00
gas price
conventional gas price
rbob gas price
heating oil price
diesel price
kerosene price
crude oil price
crude_vs_gas_price_diff
propane price
diesel_to_heating_oil_ratio
crude_oil_price_volatility_30d
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

Feature Importance using RF and SHAP Analysis:





Outlier Detection:

The Isolation Forest algorithm detects outliers by isolating observations. It randomly selects a feature and splits the dataset between the maximum and minimum values of the selected feature; this process repeats recursively, and the logic is that anomalies are

easier to isolate and will have shorter paths in the tree structure, thus identifying them as outliers more efficiently than normal points.

Isolation Forest Setup: Configures an Isolation Forest with 100 trees, using the entire training dataset for each tree and expecting about 5% of the data to be outliers, with a fixed random state for consistency.

Model Training: Fits the Isolation Forest model on X_train_New to learn and identify outliers.

Outlier Detection and Marking: After fitting, assigns an outlier score and marks outliers directly in the dataset using the is_outlier column.

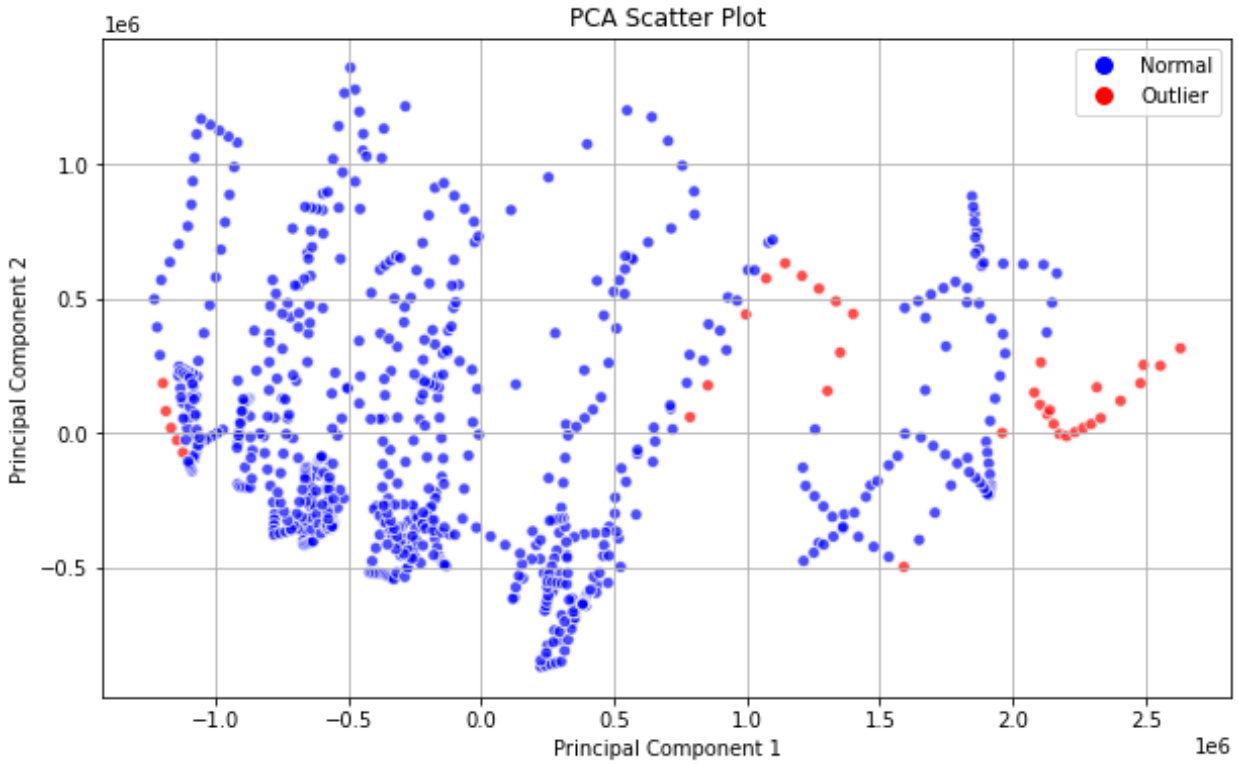
Outlier Removal: Filters out data points marked as outliers, retaining the rest in clean_data_train.

Results Display: Outputs the count of detected outliers and the shape of the dataset after outlier removal.

PCA for Outlier Visualization:

Principal Component Analysis (PCA) is a statistical technique that can be used for outlier detection among other applications like dimensionality reduction. In the context of outlier detection, PCA helps identify anomalies in a dataset by transforming the data into a set of linearly uncorrelated components (principal components) and analyzing the variance captured by these components.

- PCA transforms the original variables to new uncorrelated variables (principal components), ordered so that the first few retain most of the variation present in all of the original variables.
- The transformation to a lower-dimensional space (using the first few principal components) makes it easier to visualize and detect outliers.


PCA Scatter Plot
1e6
Normal
Outlier
1.0
0.5
0.0
−0.5
Principal Component 2
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
Principal Component 1
1e6

Final Prediction: