Anomaly Detection & Time Series Assignment

Question 1: What is Anomaly Detection? Explain its types (point, contextual, and collective anomalies) with examples.

Answer: Anomaly Detection refers to identifying unusual patterns or data points that do not conform to expected behavior. It's commonly used in fraud detection, network security, fault detection, and more.

Types of Anomalies:

1. **Point Anomalies:** A single data instance is anomalous.
   - *Example:* A temperature of 60°C in a winter season.
2. **Contextual Anomalies:** A data point is anomalous in a specific context.
   - *Example:* 30°C in winter may be normal in tropical areas but anomalous in Canada.
3. **Collective Anomalies:** A group of data points is anomalous when considered together.
   - *Example:* Sudden high traffic from a single IP address over time.

Question 2: Compare Isolation Forest, DBSCAN, and Local Outlier Factor in terms of their approach and suitable use cases.

Answer:

Algorithm	Approach	Suitable Use Cases	Strengths	Weaknesses
Isolation Forest	Randomly isolates observations to detect anomalies	High-dimensional or large numerical datasets	Fast, scalable	Doesn't handle clusters of anomalies well
DBSCAN	Groups dense clusters; detects outliers as low-density points	Spatial or location-based data	Detects arbitrary-shaped clusters	Sensitive to parameters
LOF	Measures local deviation relative to neighbors	Local anomaly detection	Detects subtle local anomalies	Not ideal for high-dimensional data

Question 3: What are the key components of a Time Series? Explain each with one example.

Answer:

1. **Trend:** Long-term increase or decrease in data.
   - *Example:* Monthly sales increasing steadily over years.
2. **Seasonality:** Regular pattern based on time (e.g., months).
   - *Example:* High ice cream sales in summer.
3. **Noise/Residual:** Random variation not explained by trend or seasonality.
   - *Example:* One-off spikes due to promotions.
4. **Cyclic:** Non-fixed periodic fluctuations.
   - *Example:* Business cycles in the economy.

Question 4: Define Stationary in time series. How can you test and transform a non-stationary series into a stationary one?

Answer: A time series is **stationary** if its statistical properties (mean, variance) do not change over time.

Testing:

- **Visual Inspection**: Look for constant mean/variance.
- **Augmented Dickey-Fuller (ADF) Test**

Transformation Methods:

- **Differencing:** Subtract current value from previous.
- **Log Transformation:** Stabilizes variance.
- **De-trending:** Remove trend component.

Question 5: Differentiate between AR, MA, ARIMA, SARIMA, and SARIMAX models in terms of structure and application.

Answer:

Model	Description	Use Case
AR (AutoRegressive)	Predicts using past values.	Stock prices.
MA (Moving Average)	Uses past forecast errors.	Noise smoothing.
ARIMA	Combines AR and MA with differencing.	Non-stationary time series.
SARIMA	ARIMA + seasonality.	Monthly temperature data.
SARIMAX	SARIMA + exogenous variables.	Energy consumption + weather.

Question 6: Load a time series dataset (AirPassengers), plot the original series, and decompose it into trend, seasonality, and residual components.

Answer (Python Code):

```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

# Load data
df = pd.read_csv("AirPassengers.csv", parse_dates=['Month'],
index_col='Month')

# Plot original series
df.plot(title='Monthly Air Passengers')
plt.show()

# Decomposition
result = seasonal_decompose(df['#Passengers'], model='multiplicative')
result.plot()
plt.show()
```

Question 7: Apply Isolation Forest on a numerical dataset (NYC Taxi Fare) to detect anomalies. Visualize the anomalies on a 2D scatter plot.

Answer (Python Code):

```
import pandas as pd
from sklearn.ensemble import IsolationForest
import matplotlib.pyplot as plt


# Load dataset
df = pd.read_csv('/mnt/data/nyc_taxi/NYC_taxi_fare_data.csv')


# Select two numerical columns
features = df[['fare_amount', 'trip_distance']].dropna()


# Apply Isolation Forest
model = IsolationForest(contamination=0.05)
features['anomaly'] = model.fit_predict(features)


# Plot
plt.scatter(features['trip_distance'], features['fare_amount'],
c=features['anomaly'], cmap='coolwarm')
plt.xlabel('Trip Distance')
plt.ylabel('Fare Amount')
plt.title('Isolation Forest - NYC Taxi Fare Anomaly Detection')
plt.show()
```

Question 8: Train a SARIMA model on the monthly airline passengers dataset. Forecast the next 12 months and visualize the results.

Answer (Python Code):

```
from statsmodels.tsa.statespace.sarimax import SARIMAX
import matplotlib.pyplot as plt
import pandas as pd

# Load dataset
df = pd.read_csv('/mnt/data/AirPassengers.csv')
df['Month'] = pd.to_datetime(df['Month'])
df.set_index('Month', inplace=True)

# Fit SARIMA model
model = SARIMAX(df['#Passengers'], order=(1,1,1), seasonal_order=(1,1,1,12))
results = model.fit()

# Forecast
forecast = results.get_forecast(steps=12)
forecast_index = pd.date_range(start=df.index[-1] + pd.DateOffset(months=1),
periods=12, freq='MS')
forecast_series = forecast.predicted_mean

# Plot
plt.plot(df['#Passengers'], label='Original')
plt.plot(forecast_index, forecast_series, color='red', label='Forecast')
plt.title('SARIMA Forecast for AirPassengers')
plt.xlabel('Date')
plt.ylabel('Passengers')
plt.legend()
plt.show()
```

Question 9: Apply Local Outlier Factor (LOF) on any numerical dataset to detect anomalies and visualize them using matplotlib.

Answer (Python Code):

```
from sklearn.neighbors import LocalOutlierFactor
import matplotlib.pyplot as plt
import pandas as pd

# Load dataset
df = pd.read_csv('/mnt/data/nyc_taxi/NYC_taxi_fare_data.csv')
data = df[['fare_amount', 'trip_distance']].dropna()

# LOF
lof = LocalOutlierFactor(n_neighbors=20)
data['anomaly'] = lof.fit_predict(data)

# Visualization
plt.scatter(data['trip_distance'], data['fare_amount'], c=data['anomaly'],
cmap='plasma')
plt.xlabel('Trip Distance')
plt.ylabel('Fare Amount')
plt.title('Local Outlier Factor - NYC Taxi Data')
plt.show()
```

Question 10:

You are working as a data scientist for a power grid monitoring company. Your goal is to forecast energy demand and also detect abnormal spikes or drops in real-time consumption data collected every 15 minutes. The dataset includes features like timestamp, region, weather conditions, and energy usage.

Answer:

Workflow:

1. **Data Ingestion**: Use streaming tools like Apache Kafka or Spark Streaming.
2. **Anomaly Detection**:
   - Use **Isolation Forest** for detecting sudden spikes/drops in real-time.
   - Use **LOF** for localized anomaly detection.
   - **DBSCAN** if the data has natural clustering.
3. **Forecasting Model:**
   - Use **SARIMAX** to capture seasonality and include weather as an exogenous variable.
4. **Validation:**
   - Use metrics like RMSE/MAE.

- Backtesting on historical data.
- Online monitoring dashboard for drift detection.

5. **Business Impact:**
   - Helps in grid stability.
   - Early alerts reduce downtime.
   - Optimizes energy distribution planning.

End of Assignment