```
import random
import pandas as pd
import numpy as np
from datetime import datetime
```

```
def generate_live_data(n_samples=1):
    data = {
        "speed": [random.randint(20, 120) for _ in range(n_samples)],
        "weather_risk": [random.choice([0, 1]) for _ in range(n_samples)],
        "road_condition": [random.choice([0, 1]) for _ in range(n_samples)],
        "hour": [datetime.now().hour for _ in range(n_samples)],
        "latitude": [random.uniform(12.8, 13.1) for _ in range(n_samples)],
        "longitude": [random.uniform(77.4, 77.7) for _ in range(n_samples)]
    }

    df = pd.DataFrame(data)

    df["accident"] = (
        (df["speed"] > 80) &
        (df["weather_risk"] == 1) &
        (df["road_condition"] == 1)
    ).astype(int)

    return df
```

```
live_training_data = generate_live_data(1000)

X = live_training_data.drop("accident", axis=1)
y = live_training_data["accident"]
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

print("✅ Live ML Model Trained")
```

```
✅ Live ML Model Trained
```

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=3, random_state=42)
live_training_data["hotspot"] = kmeans.fit_predict(
    live_training_data[["latitude", "longitude"]]
)
```

```
from IPython.display import Audio, display

def siren_alert():
    sr = 44100
    t = np.linspace(0, 1, sr)
    sound = np.sin(2 * np.pi * 1000 * t)
    display(Audio(sound, rate=sr, autoplay=True))
```

```
import time

while True:
    live_data = generate_live_data(1)
    features = live_data.drop("accident", axis=1)

    risk_prob = model.predict_proba(features)[0][1]

    print("\n📡 LIVE VEHICLE DATA")
    print(features)
    print(f"⚠️ Accident Risk Probability: {risk_prob:.2f}")

    if risk_prob > 0.7:
        print("🚨 HIGH RISK – TAKE PREVENTIVE ACTION 🚨")
```

```
        siren_alert()

    elif risk_prob > 0.4:
        print("⚠ MEDIUM RISK – DRIVE CAREFULLY")

    else:
        print("✅ LOW RISK – SAFE DRIVE")

    time.sleep(3)
```