Q1:

```python
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import LabelEncoder

data = {
    'Weather': ['Sunny', 'Sunny', 'Overcast', 'Rainy', 'Rainy', 'Overcast', 'Sunny', 'Sunny', 'Rainy', 'Sunny'],
    'Temperature': ['Hot', 'Hot', 'Mild', 'Mild', 'Cool', 'Mild', 'Mild', 'Hot', 'Mild', 'Mild'],
    'Play': ['No', 'No', 'Yes', 'Yes', 'Yes', 'Yes', 'No', 'Yes', 'Yes', 'No']
}

df = pd.DataFrame(data)

le_weather = LabelEncoder()
le_temperature = LabelEncoder()
le_play = LabelEncoder()

df['Weather'] = le_weather.fit_transform(df['Weather'])
df['Temperature'] = le_temperature.fit_transform(df['Temperature'])
df['Play'] = le_play.fit_transform(df['Play'])

X = df[['Weather', 'Temperature']]
y = df['Play']

nb = GaussianNB()

nb.fit(X, y)

prediction = nb.predict([[le_weather.transform(['Overcast'])[0], le_temperature.transform(['Mild'])[0]]])

play_result = le_play.inverse_transform(prediction)
print(f"Prediction: The player {'can' if play_result[0] == 'Yes' else 'cannot'} play when the weather is overcast and the temperature is mild.")
```

```
Prediction: The player can play when the weather is overcast and the temperature is mild.
```

Q2:

HomeTask: