PROJECT DOCUMENTATION

Predicting Energy Production in a Combined-Cycle Power Plant

Introduce the project as a machine learning solution to optimize energy production in power plants by forecasting output based on environmental conditions and operational factors. Emphasize the potential benefits for efficiency, cost reduction, and sustainability.

Project Overview

Motivation:

The demand for efficient, cost-effective, and sustainable energy solutions has been steadily increasing worldwide. Combined-cycle power plants, which utilize both gas and steam turbines to generate electricity, offer one of the most efficient ways to produce energy. However, the energy output of these plants is sensitive to various environmental and operational factors, such as ambient temperature, pressure, humidity, and turbine performance. By accurately predicting energy production based on these variables, power plant operators can optimize plant operations, reduce fuel consumption, and improve overall efficiency.

Objective:

The goal of this project is to develop a machine learning model that predicts hourly energy output in megawatts (MW) based on key environmental and operational parameters. Using data from a real-world combined-cycle power plant, this model aims to capture complex relationships between variables and provide reliable energy forecasts. With an effective predictive model, operators can make proactive adjustments to enhance plant performance and lower operating costs.

Business Impact:

Predicting energy production accurately has several strategic benefits for power plant management:

1. **Efficiency Optimization:** By understanding how variables affect energy output, operators can adjust settings to optimize efficiency.
2. **Cost Reduction:** Accurate predictions allow for optimized fuel consumption, lowering operational costs.
3. **Proactive Maintenance:** Monitoring predictive trends can identify performance deviations, guiding preventive maintenance and reducing downtime.
4. **Sustainability:** Improved efficiency reduces fuel usage and emissions, aligning with sustainability goals and environmental regulations.

Problem Statement

Combined-cycle power plants generate electricity using both gas and steam turbines, creating a more efficient energy production process compared to traditional single-cycle plants. However, the energy output of these plants is highly influenced by various environmental and operational factors, including temperature, exhaust vacuum, ambient pressure, and relative humidity. These factors are unpredictable and can significantly affect the plant's efficiency and cost-effectiveness.

Currently, there is no reliable, data-driven approach for predicting the plant's energy output in response to changing environmental conditions. This lack of predictive capability limits operators' ability to make informed adjustments to optimize performance, manage fuel consumption, and reduce emissions.

The objective of this project is to develop a machine learning model that accurately predicts the hourly energy production of a combined-cycle power plant based on these influencing variables. By forecasting energy output, the model will enable plant operators to:

1. Anticipate and adjust for fluctuations in energy production,
2. Optimize operational settings for improved efficiency, and
3. Support sustainable energy management by reducing unnecessary fuel use and emissions.

This project will leverage historical data from the plant, analyzing the relationships between key variables and energy output, to create a reliable prediction tool that supports proactive, data-driven decision-making in power plant operations.

Understanding Combined-Cycle Power Plants

-**Plant Design**: Combined-cycle plants use both gas and steam turbines in a closed cycle, capturing waste heat to power the steam turbine, significantly increasing efficiency.

- **Advantages**: These plants are more efficient and have lower emissions than single-cycle plants, making them a preferred choice for modern power generation.

Dataset Description

-Data Collection: 9,568 observations collected over six years under full-load operational conditions.

- Features and Target Variable:

- Features: Temperature (°C), exhaust vacuum (cm Hg), ambient pressure (mbar), and relative humidity (%).

-Target Variable: Energy production (MW), representing the net hourly electrical output.

-Visual: Table with feature descriptions and sample data points to give an overview of the dataset structure

	temperature	exhaust_vacuum	amb_pressure	r_humidity	energy_production
0	9.59	38.56	1017.01	60.10	481.30
1	12.04	42.34	1019.72	94.67	465.36
2	13.87	45.08	1024.42	81.69	465.48
3	13.72	54.30	1017.89	79.08	467.05
4	15.14	49.64	1023.78	75.00	463.58

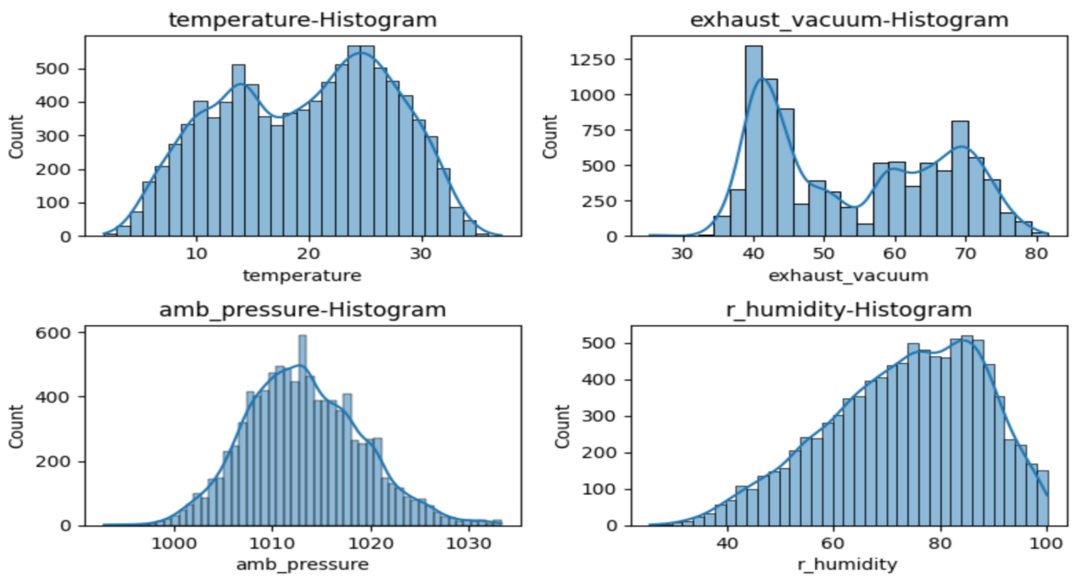
Exploratory Data Analysis (EDA)

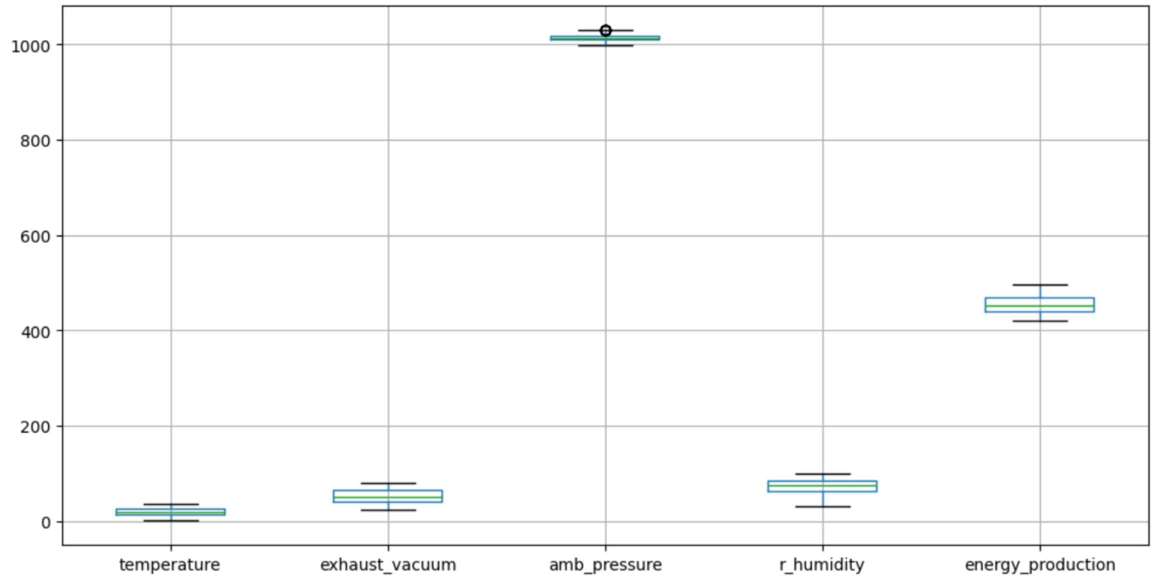
- Objective: Identify initial patterns, distributions, and correlations to guide model selection.

- Approach: Used univariate, bivariate, and multivariate analyses to understand the relationships between features and the target variable.

-Outcome: This analysis revealed strong correlations between energy production and key variables, guiding the feature selection for model training.

- Visual: Histograms and box plots to illustrate feature distributions and detect any anomalies.



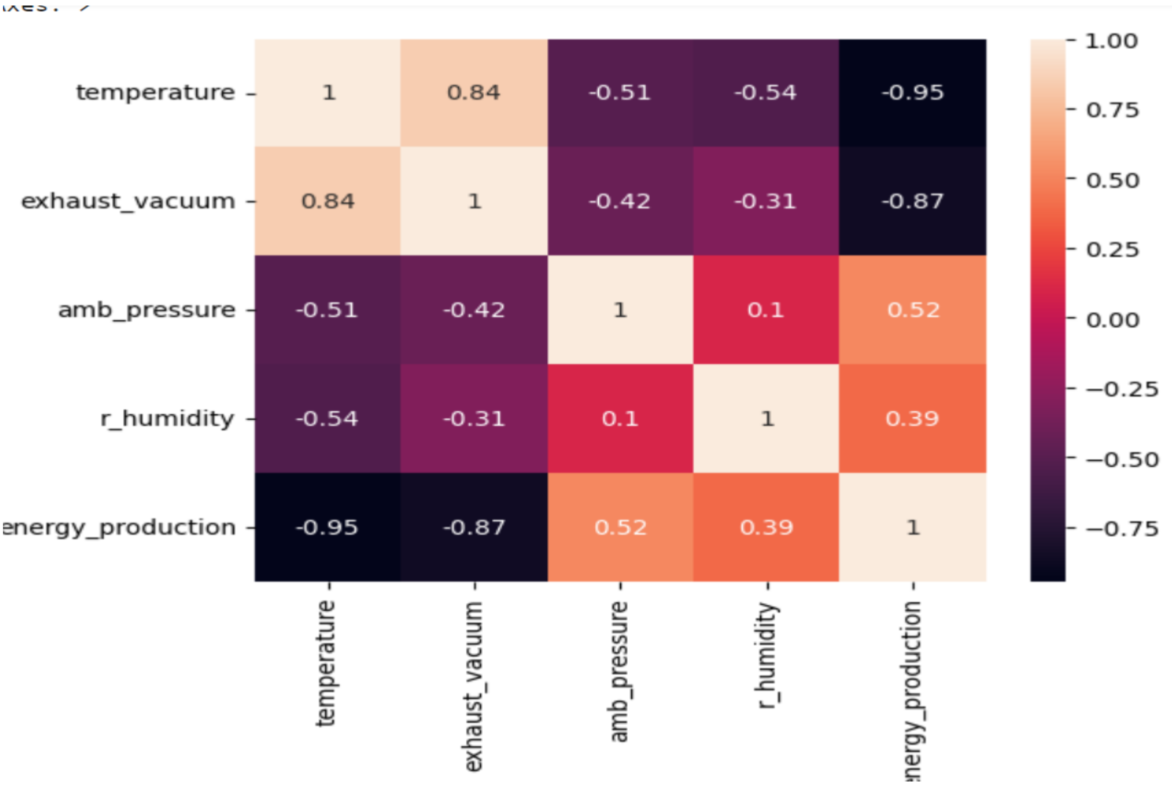


Feature Relationships and Correlation Analysis

-Bivariate Analysis: Examined the correlation between each feature and energy production, focusing on identifying which factors most influence the target variable.

- Insights: Temperature and exhaust vacuum were found to have the highest correlations with energy output, indicating they are critical features.

-Visual: Correlation heatmap highlighting the strength of relationships between variables, with temperature and exhaust vacuum standing out.



Data Preprocessing Overview

- **Objective**: Clean and prepare the data for model training to improve accuracy and reliability.

- **Preprocessing Steps**:

  - **Outlier Detection and Treatment**: Handled outliers to minimize their influence on model accuracy.

  - **Feature Scaling**: Standardized features using Min-Max Scaling to ensure that all variables contribute equally.

 - **Goal**: Create a consistent, high-quality dataset for optimal model performance.

Model Selection

- **Models Evaluated**: Linear Regression, Decision Tree Regressor , and Random Forest etc.

- **Why Random Forest**: Random Forest was chosen as the best model for its ability to handle non-linear relationships, robustness against overfitting, and provision of feature importance.

- **Model Hypothesis**: Random Forest, as an ensemble method, was expected to capture complex relationships more effectively, leading to better accuracy and generalization.

Training and Testing Split

- **Training and Testing Process:** Split the data into 80% for training and 20% for testing to validate the model's predictive power on unseen data.

- **Cross-Validation**: Used 5-fold cross-validation to ensure the model's robustness and avoid overfitting.

- **Goal**: Ensure the model's performance is generalizable and reliable for real-world deployment.

```
[ ] from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LinearRegression,Lasso
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.svm import SVR
    from sklearn.neighbors import KNeighborsRegressor
    from sklearn.metrics import mean_squared_error, r2_score,accuracy_score
```

```
[ ] x_train,x_test,y_train,y_test=train_test_split(features,target,train_size=0.80,random_state=42)
```

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(7621, 4)
(1906, 4)
(7621,)
(1906,)
```

Model Training and Hyperparameter Tuning

- **Training Process**: Trained each model using the training data, optimizing their settings to enhance performance.

- **Hyperparameters**: Tuned key Random Forest parameters, including the number of trees, maximum depth, and feature splits, using Grid Search to achieve the best configuration.

- **Purpose**: This tuning step allowed the model to balance accuracy and generalization, leading to an optimized predictive model.

Model Evaluation Metrics

- **Metrics Used**:

  - Mean Absolute Error (MAE): Measures the average error in predictions.

  - Mean Squared Error (MSE): Penalizes larger errors, indicating model variance.

  - R-squared: Reflects the proportion of variance explained by the model, showing its predictive strength.

- **Comparison**: Random Forest achieved the lowest error and highest R-squared, indicating it is the best choice for this project.

- **Visual**: Table comparing the performance metrics of Linear Regression, Gradient Boosting, and Random Forest.

```
[ ] models={
        'Linear Regression':LinearRegression(),
        'Decision Tree':DecisionTreeRegressor(),
        'Randomforest':RandomForestRegressor(),
        'Support vector':SVR()
    }
```

```
for model_name,model in models.items():
    model.fit(x_train, y_train)
    y_pred = model.predict(x_test)
    mse = mean_squared_error(y_test, y_pred)
    r2_sc=r2_score(y_test,y_pred)
    print(f"{model_name},Mean Squared Error: {mse:.4f},r2 score:{r2_sc:.4f}")
```

```
Linear Regression,Mean Squared Error: 20.8782,r2 score:0.9283
Decision Tree,Mean Squared Error: 19.1360,r2 score:0.9343
Randomforest,Mean Squared Error: 11.1223,r2 score:0.9618
Support vector,Mean Squared Error: 181.6420,r2 score:0.3762
```

Random Forest as Best Model

- **Performance Analysis**: Random Forest showed the best balance of accuracy and interpretability, excelling in both MAE and R-squared metrics.

- **Feature Importance**: Random Forest's feature importance analysis indicated that temperature and exhaust vacuum are the most influential factors for energy prediction.

Model Insights and Recommendations

- **Key Predictors**: Temperature and exhaust vacuum significantly impact energy production, making them essential for monitoring.

- **Actionable Insights**: The model suggests maintaining control over these factors can help optimize output.

- **Future Recommendations**: Regularly update the model with new data to maintain its predictive power and explore additional factors for deeper insights.

Introduction to Deployment

- **Deployment Goal**: Make the predictive model accessible via a user-friendly web application.

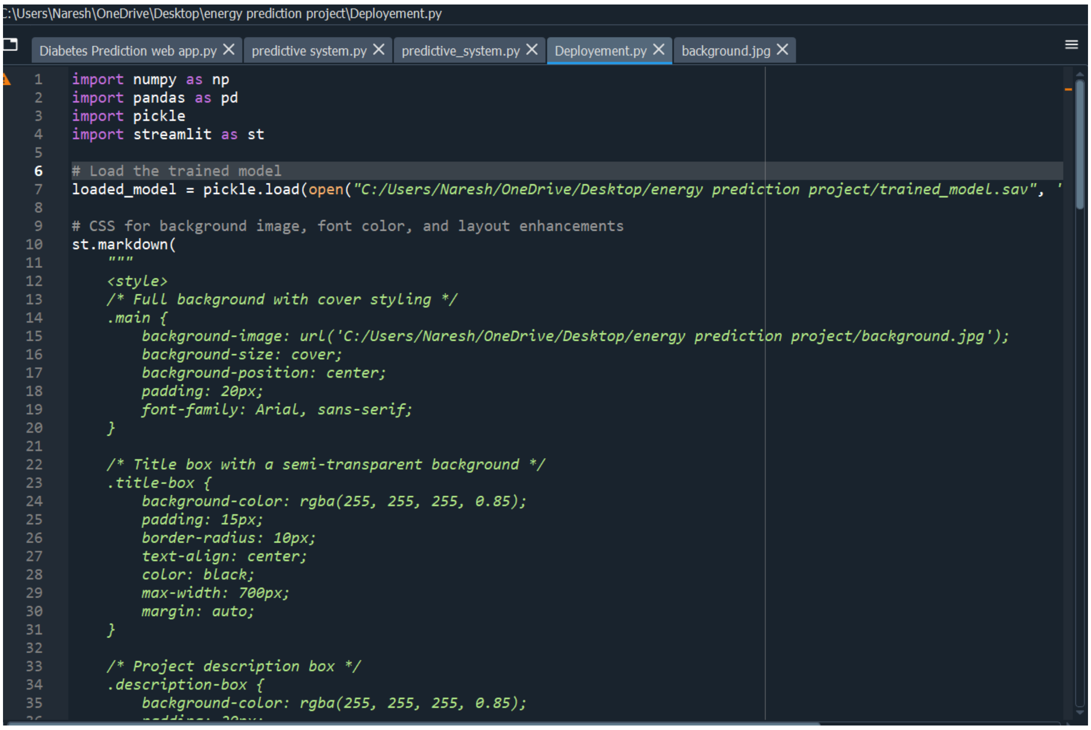
- **Deployment Tools**: Implemented with Streamlit to create a simple, interactive interface.

- **Benefit**: Enables real-time energy forecasting, allowing operators to make proactive decisions based on model predictions.

Deployment Architecture

- **Deployment Structure**: Overview of the application structure, from user input through to backend processing and result output.

- **Components**: Frontend for user input, backend for model processing, and output display for predictions.



```
    /* Project description box */
    .description-box {
        background-color: rgba(255, 255, 255, 0.85);
        padding: 20px;
        border-radius: 10px;
        max-width: 700px;
        margin: 20px auto;
        color: black;
        text-align: justify;
    }

    /* Content box for input fields */
    .content-box {
        background-color: rgba(255, 255, 255, 0.85);
        padding: 20px;
        border-radius: 10px;
        max-width: 700px;
        margin: 20px auto;
        color: black;
    }

    /* Prediction result box */
    .result-box {
        background-color: rgba(0, 204, 102, 0.8);
        padding: 15px;
        border-radius: 10px;
        text-align: center;
        font-size: 29px;
        color: white;
        max-width: 700px;
        margin: 20px auto;
    }
```

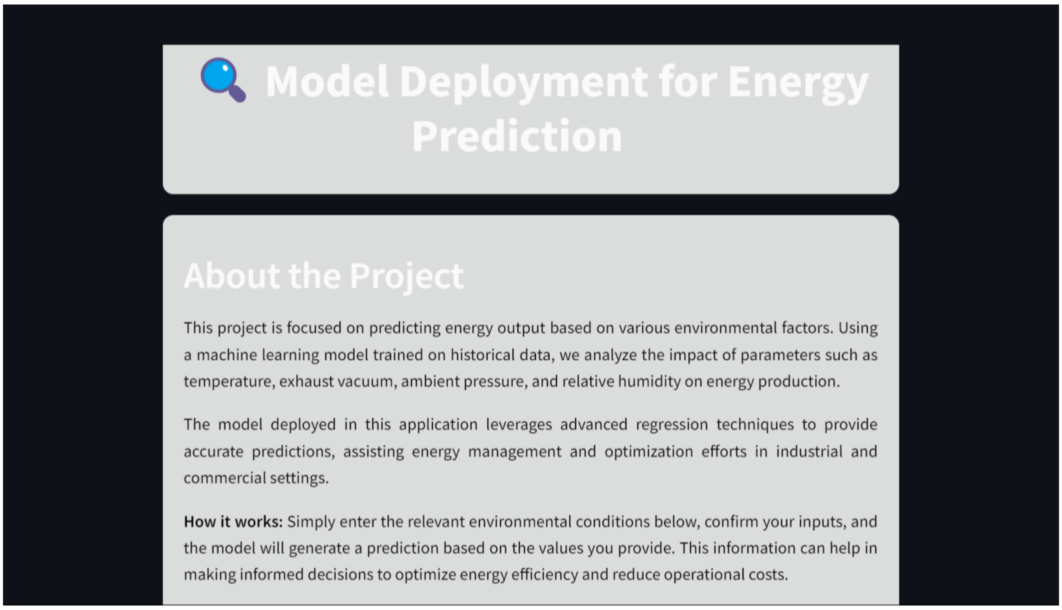
```
# Title of the app
st.markdown(
    """
    <div class="title-box">
        <h1>🔍 Model Deployment for Energy Prediction</h1>
    </div>
    """,
    unsafe_allow_html=True
)

# Project description
st.markdown(
    """
    <div class="description-box">
        <h2>About the Project</h2>
        <p>This project is focused on predicting energy output based on various environmental factors.
        Using a machine learning model trained on historical data, we analyze the impact of parameters
        such as temperature, exhaust vacuum, ambient pressure, and relative humidity on energy production.</p>
        <p>The model deployed in this application leverages advanced regression techniques to provide
        accurate predictions, assisting energy management and optimization efforts in industrial and commercial s
        <p><strong>How it works:</strong> Simply enter the relevant environmental conditions below,
        confirm your inputs, and the model will generate a prediction based on the values you provide. This infor
        help in making informed decisions to optimize energy efficiency and reduce operational costs.</p>
    </div>
    """,
    unsafe_allow_html=True
)

# Input parameters section
st.markdown(
    """
    <div class="content-box">
        <h2>Enter Input Parameters</h2>
        <p>Provide the following environmental conditions for an accurate prediction:</p>
```

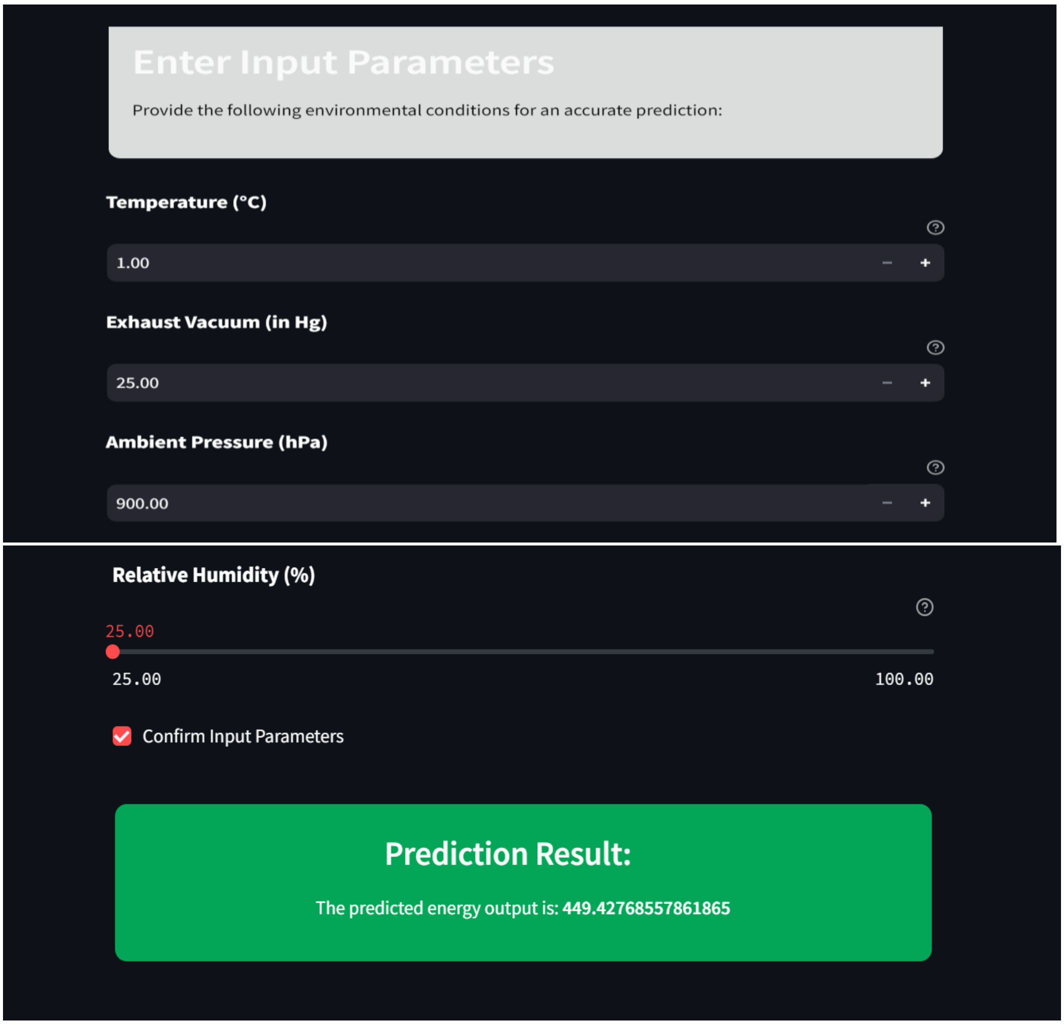
```python
st.markdown("<div class='input-header'>Exhaust Vacuum (in Hg)</div>", unsafe_allow_html=True)
exhaust_vacuum = st.number_input('', min_value=25.0, max_value=100.0, value=25.0, step=0.1, help="Enter the exhau

st.markdown("<div class='input-header'>Ambient Pressure (hPa)</div>", unsafe_allow_html=True)
amb_pressure = st.number_input('', min_value=900.0, max_value=1050.0, value=900.0, step=0.1, help="Enter the ambi

st.markdown("<div class='input-header'>Relative Humidity (%)</div>", unsafe_allow_html=True)
r_humidity = st.slider('', 25.0, 100.0, 25.0, step=0.1, help="Adjust the relative humidity percentage.")

# Confirmation checkbox
if st.checkbox("Confirm Input Parameters"):
    # Prepare data for prediction
    input_data = pd.DataFrame([[temperature, exhaust_vacuum, amb_pressure, r_humidity]],
                              columns=['temperature', 'exhaust_vacuum', 'amb_pressure', 'r_humidity'])

    # Make the prediction
    prediction = loaded_model.predict(input_data)

    # Display the prediction result
    st.markdown(
        f"""
        <div class="result-box">
            <h3>Prediction Result:</h3>
            <p>The predicted energy output is: <strong>{prediction[0]}</strong></p>
        </div>
        """,
        unsafe_allow_html=True
    )
else:
    st.markdown(
        """
        <div class="message-box">
            Please confirm your input parameters to see the prediction.
        </div>
        """
```

User Interface Design

- **Interface Elements**: User input fields for environmental variables, along with an output section for predicted energy production.

- **Visual**: Mock-up or screenshot of the web interface showing how users interact with the model by entering values and receiving immediate predictions.





Final Observations and Recommendations

- **Summary of Findings**: Random Forest was selected as the optimal model due to its high accuracy, interpretability, and robustness.

- **Future Improvements**: Suggest incorporating real-time data from sensors to update predictions dynamically and periodically retrain the model for continuous improvement.

- **Broader Impact**: The insights gained can support further research and development in optimizing energy production within the industry.