**One issue related to environment and energy that can be solved using Python is the prediction of solar energy output. Solar power is a promising source of renewable energy, but its output is dependent on weather conditions and other factors. Accurate predictions of solar energy output can help power grid operators better manage their systems and make more informed decisions about when to use other sources of energy.**

**Using Python, you can analyze historical weather data, solar panel performance data, and other relevant data sources to build models that predict future solar energy output. These models can take into account variables such as cloud cover, temperature, humidity, and solar panel orientation and tilt. Python has powerful libraries for data analysis and machine learning, such as Pandas, NumPy, and Scikit-Learn, which can be used to build and train these models.**

**Once the model is built, it can be deployed as a web application or integrated into other systems using Python frameworks such as Flask or Django. This can provide real-time predictions of solar energy output, which can be used to inform energy management decisions and improve the efficiency and reliability of renewable energy systems.**

**In this example, I'm assuming that the solar panel and weather data is stored in a CSV file called 'solar\_data.csv'. The data contains columns for temperature, humidity, cloud cover, and solar output. We use the Pandas library to load the data into a DataFrame, and then split the data into training and testing sets using the train\_test\_split function from Scikit-Learn.**

**Next, we train a linear regression model on the training data using the LinearRegression class from Scikit-Learn. We use the model to make predictions on the test data, and then calculate the mean squared error of the predictions using the mean\_squared\_error function from Scikit-Learn.**

**Of course, this is just a simple example, and there are many ways you could improve this code and model depending on your specific needs and data.**