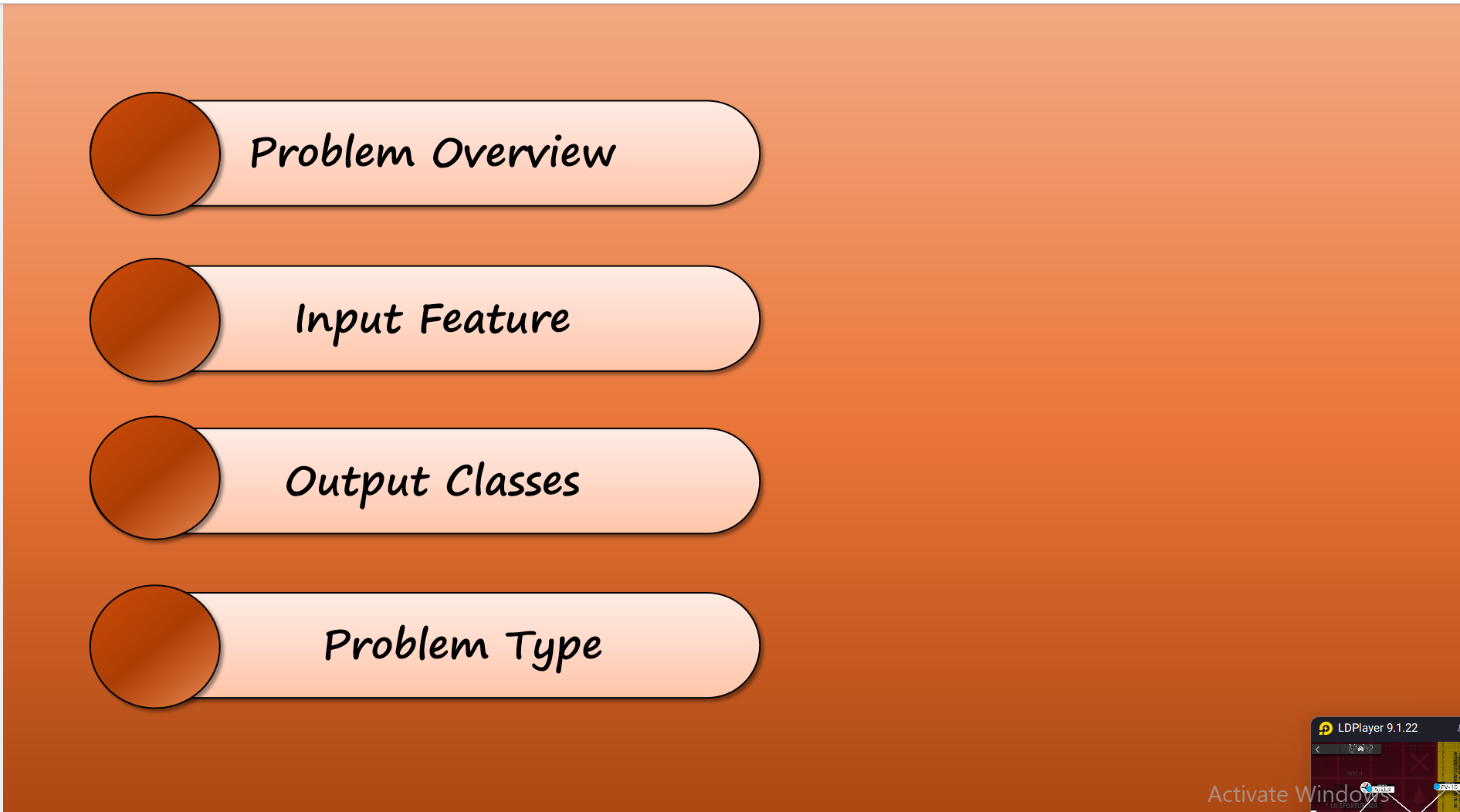
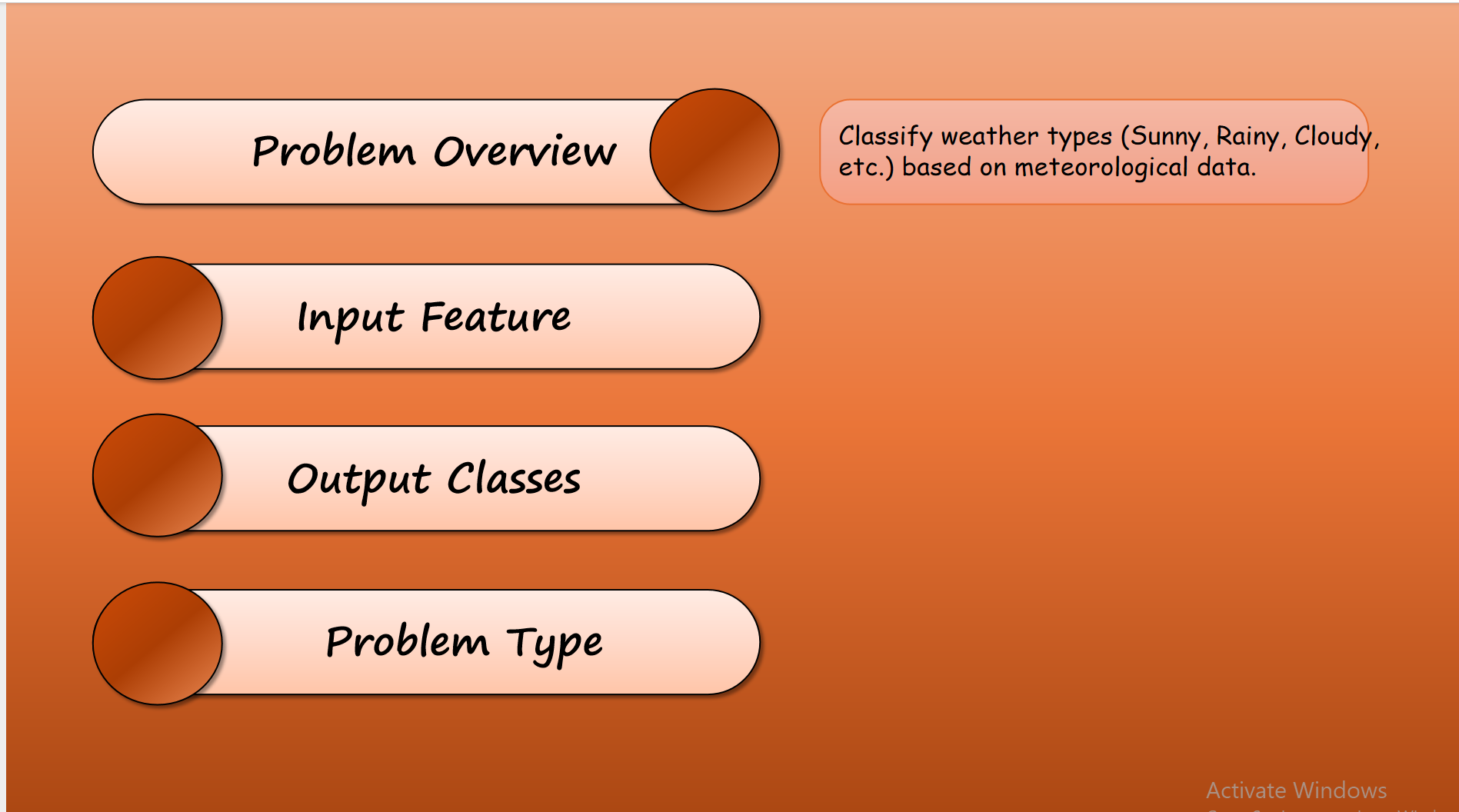
Giới thiệu nhóm, thành viên. Cái này tự múa nhé :)).

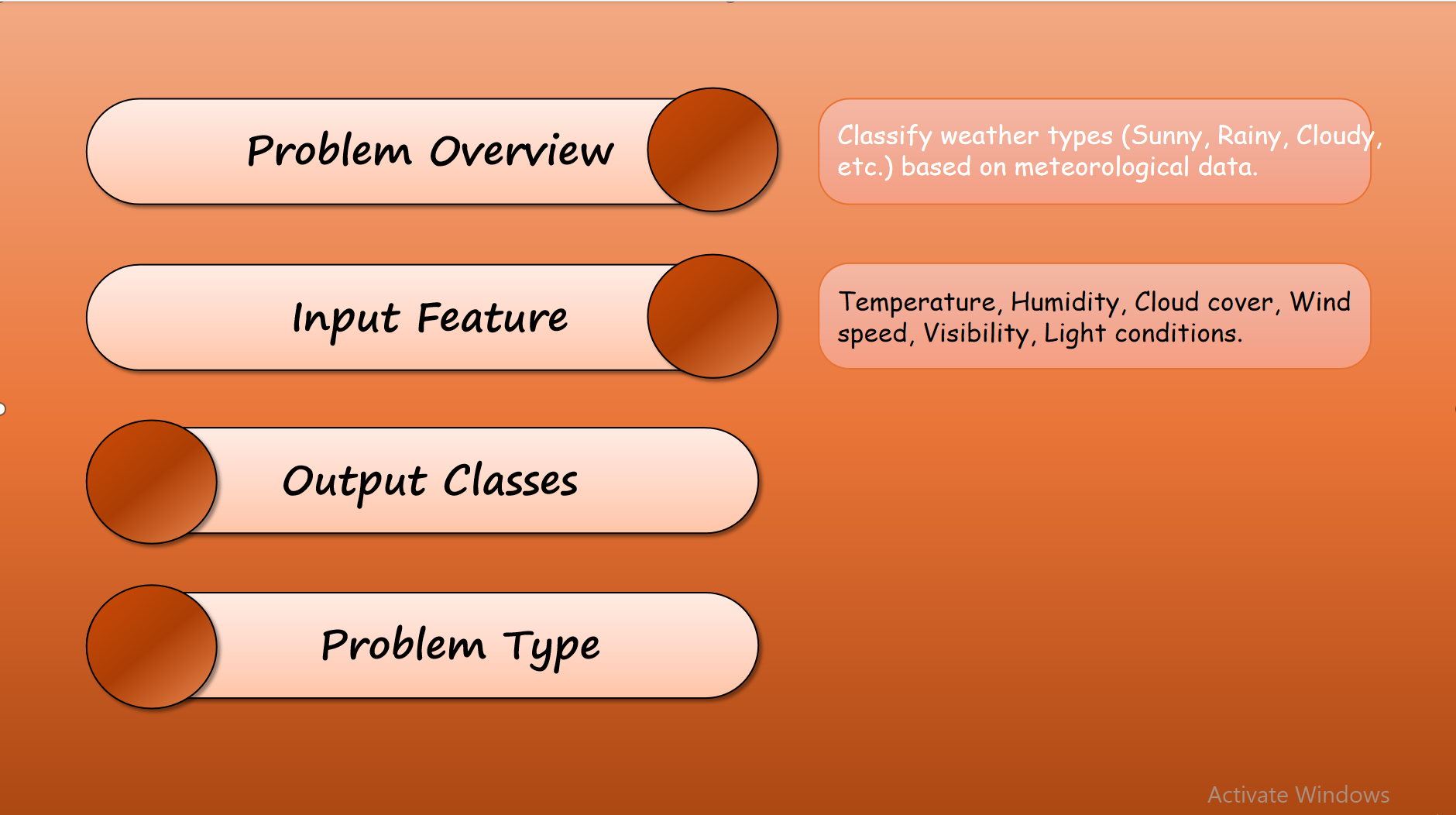
Dẫn vào bài:



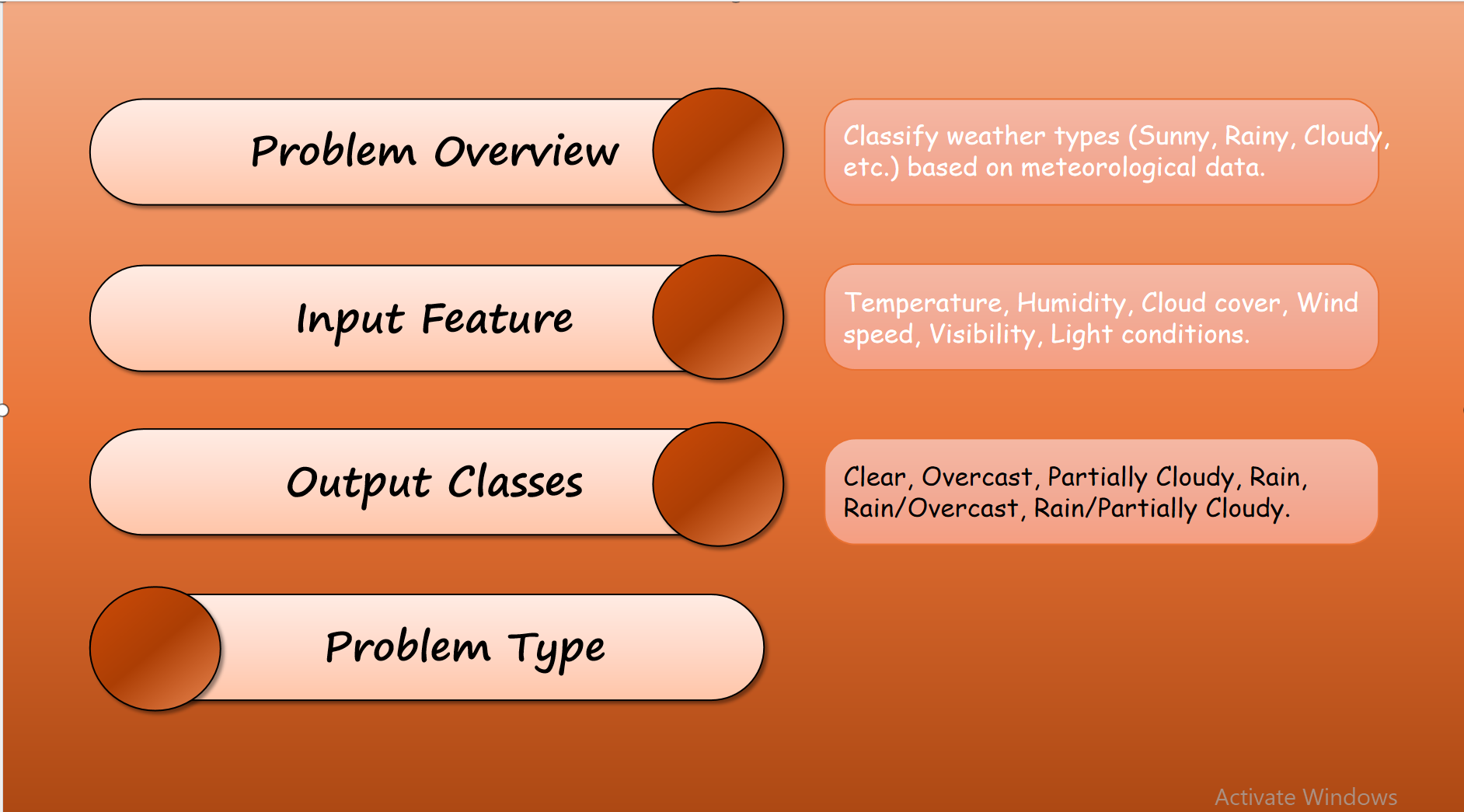
Understanding and predicting weather conditions is essential for various sectors including agriculture, transportation, and public safety. This project aims to develop a supervised machine learning model that classifies weather types based on real-time meteorological inputs.



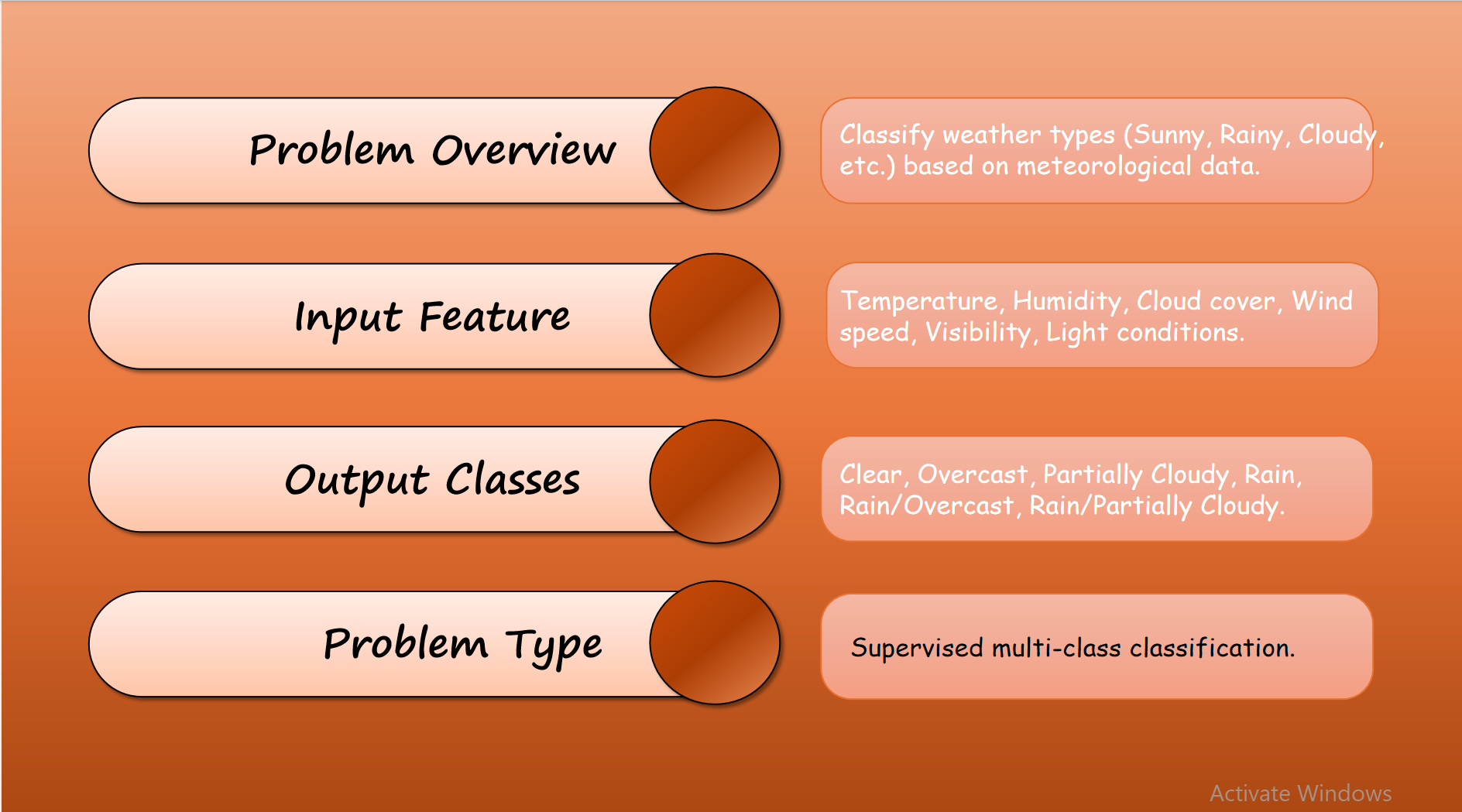
Using quantitative atmospheric data, the objective is to classify weather into distinct types such as *Clear*, *Overcast*, *Rain*, and their combinations (e.g., *Rain/Overcast*). This enables automated weather categorization for downstream applications such as automated alerts or environmental monitoring.



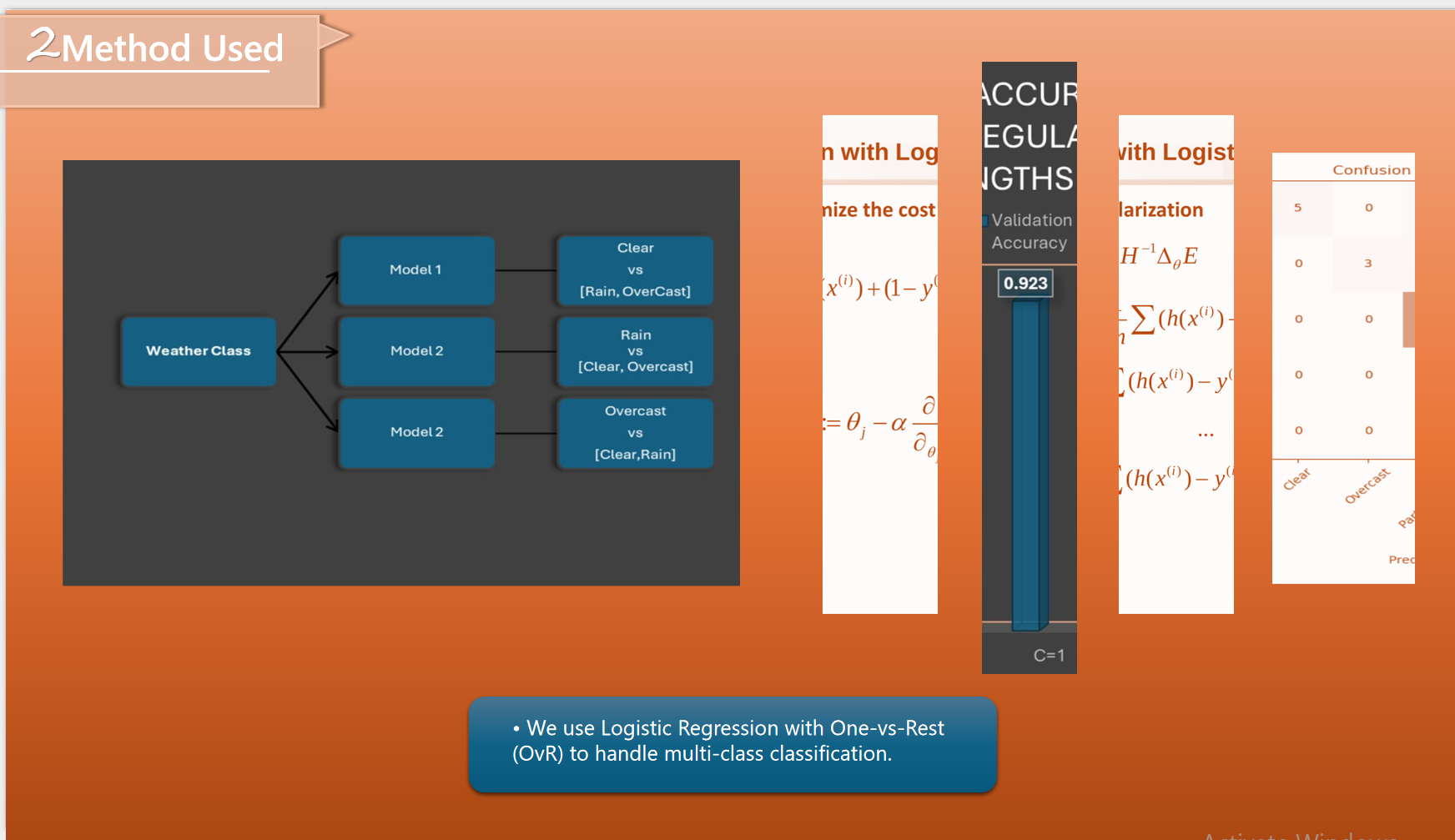
The input to the model is real-world meteorological data collected from weather stations. Each instance in the dataset includes measured features related to: Temperature, Humidity, Cloud cover, Wind speed, Visibility, and Light conditions.



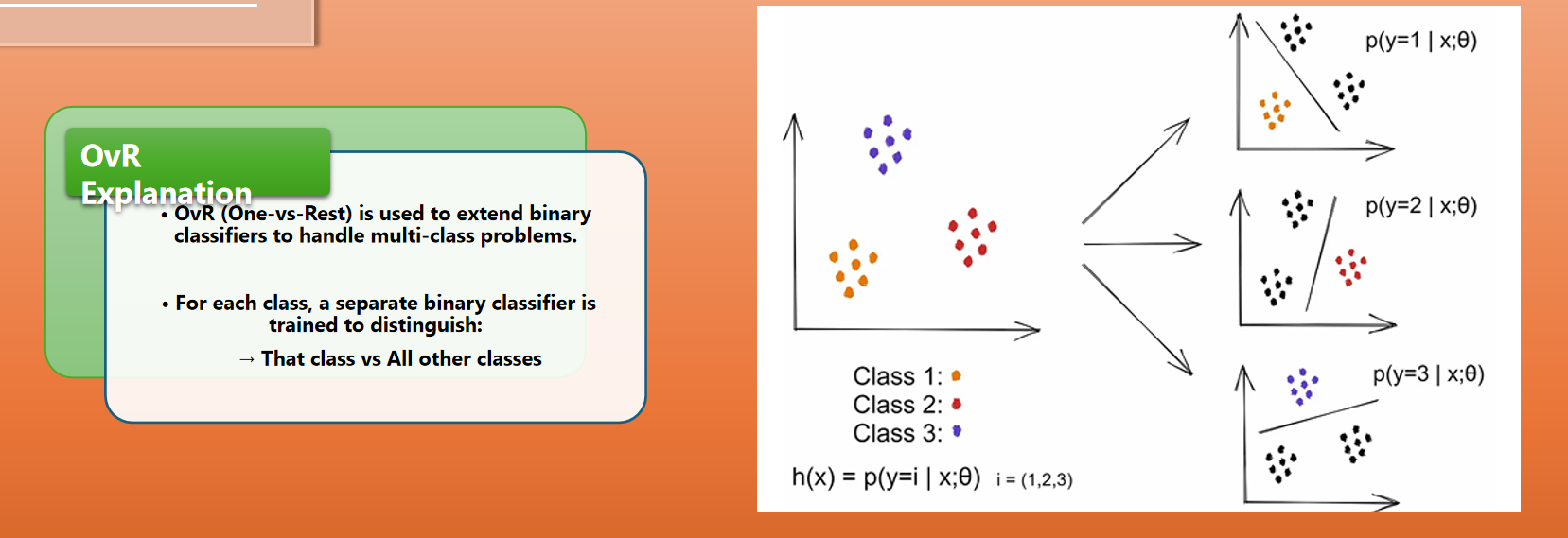
About the output of the model, it will categorize data into one of the following weather types: Clear, Overcast, Partially Cloudy, Rain, Rain & Overcast, Rain & Partially Cloudy.



Finally, this is a supervised multi-class classification problem, where each input instance is labeled with one of the defined weather categories. Given the labeled data, the model will learn to map input features to corresponding weather classes.

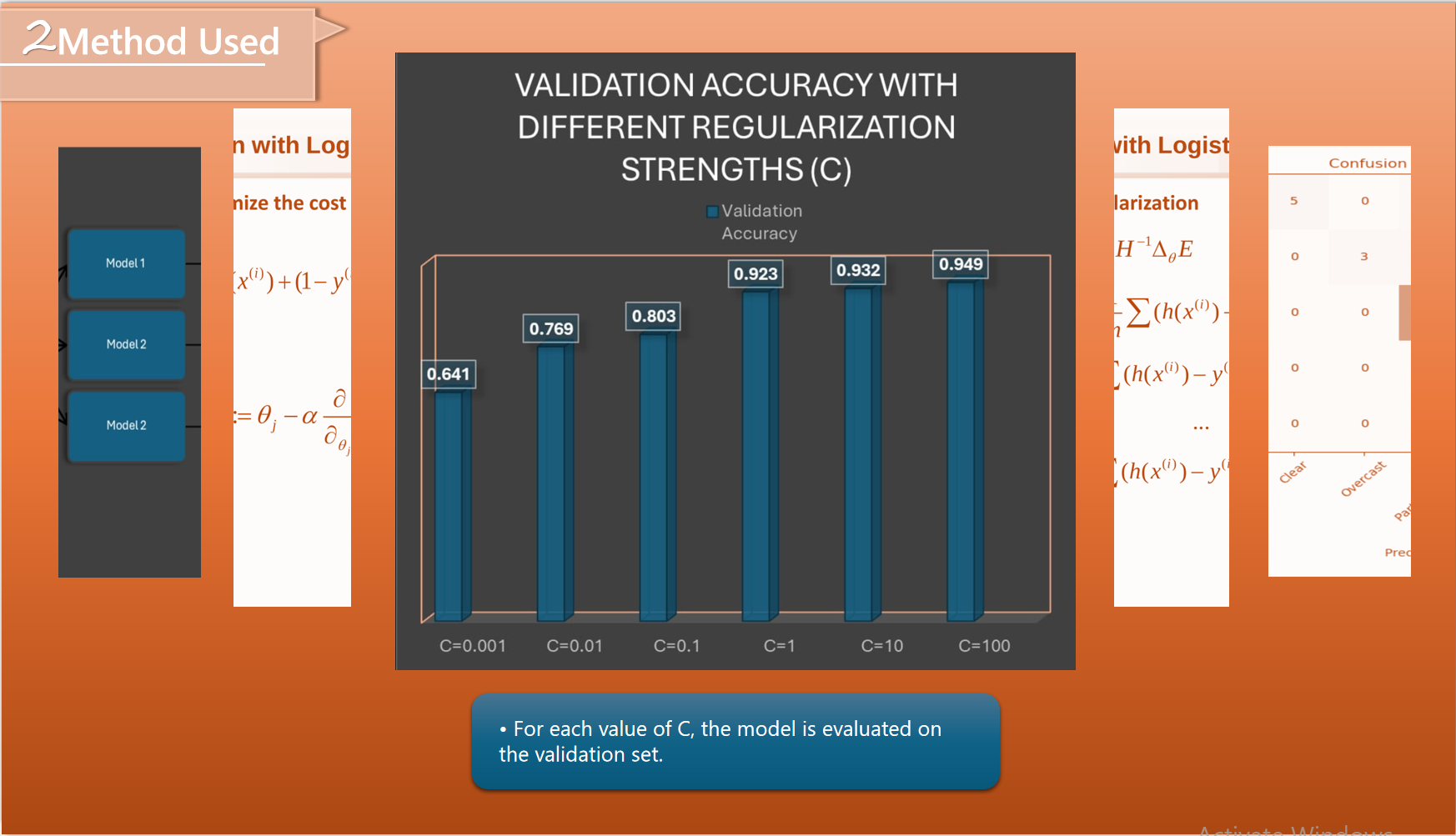
In the next section, we will discuss about the method that we use for this model.

We chose Logistic Regression with One-vs-Rest (OvR) for our weather classification model because it perfectly balances simplicity and effectiveness.

The idea of OvR is to divide the multiclass classification problem into binary classification problems, in which each binary classification problem will classify a certain class from all the remaining classes.

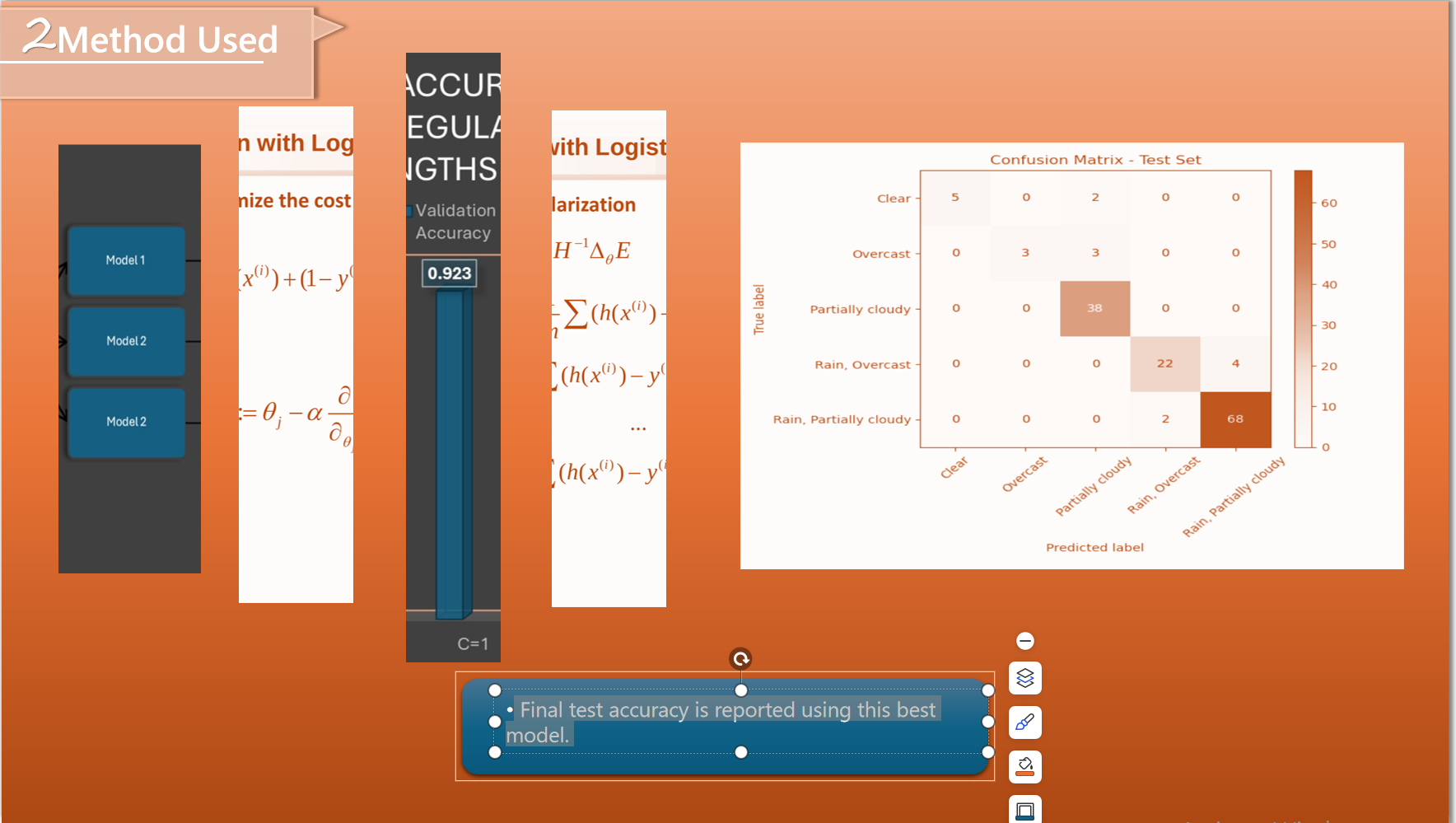


In Logistic Regression, we use the Sigmoid function to predict the output, and to help the model be more accurate, we minimize the cost function by changing theta. But sometimes, it may happen that some cases are overfitting, so we need to use regularization to avoid it. Our model will be trained with different regularization strengths to see what is the most efficient for the model.

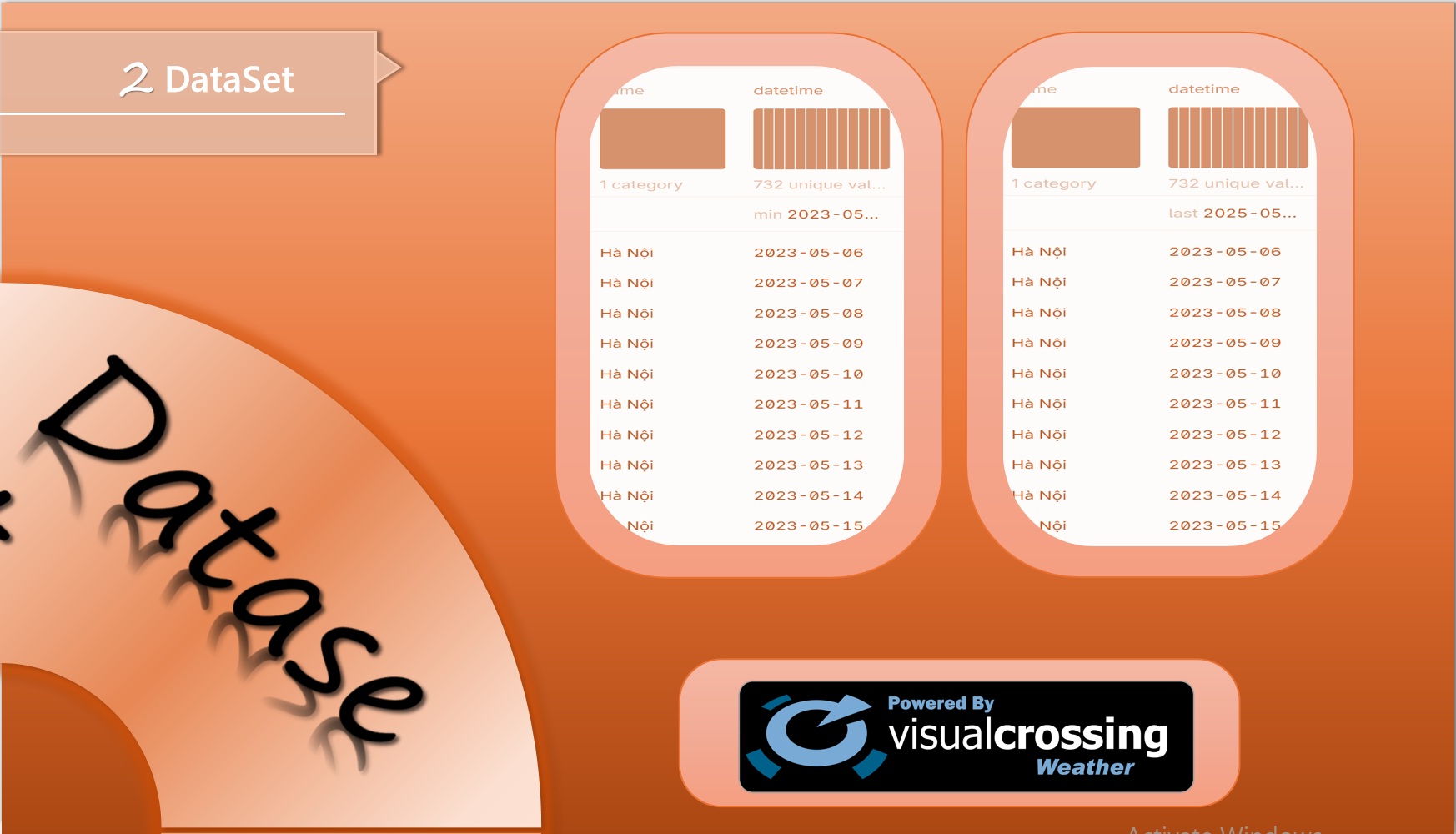


To do it, we divide the data set into 3 parts 

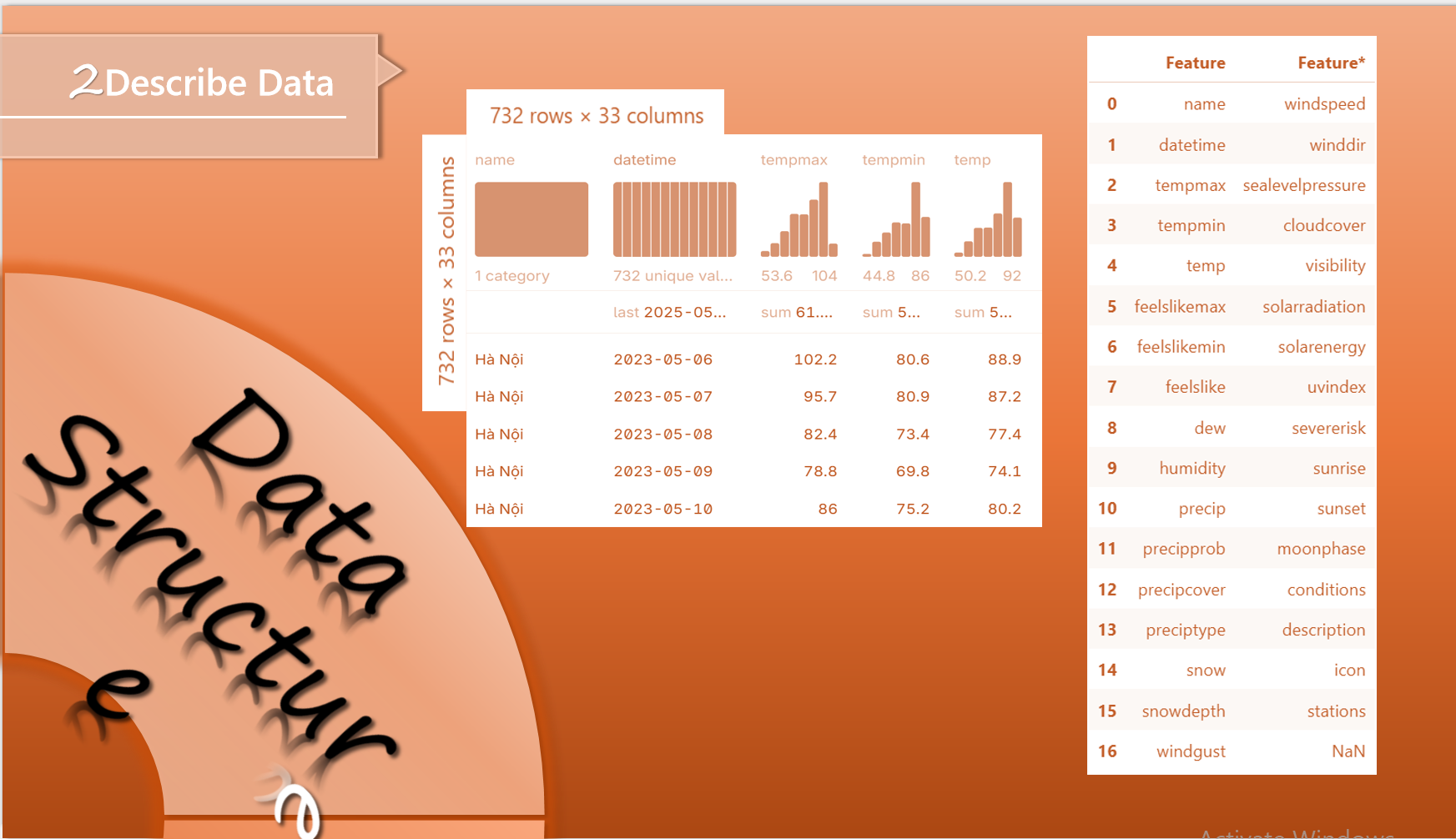
And for each value of c, the model is evaluated on the validation test. After that, the best model is chosen based on the highest validation accuracy.

 Final test accuracy is reported using this best model.

Let's look at the data that powers this model.

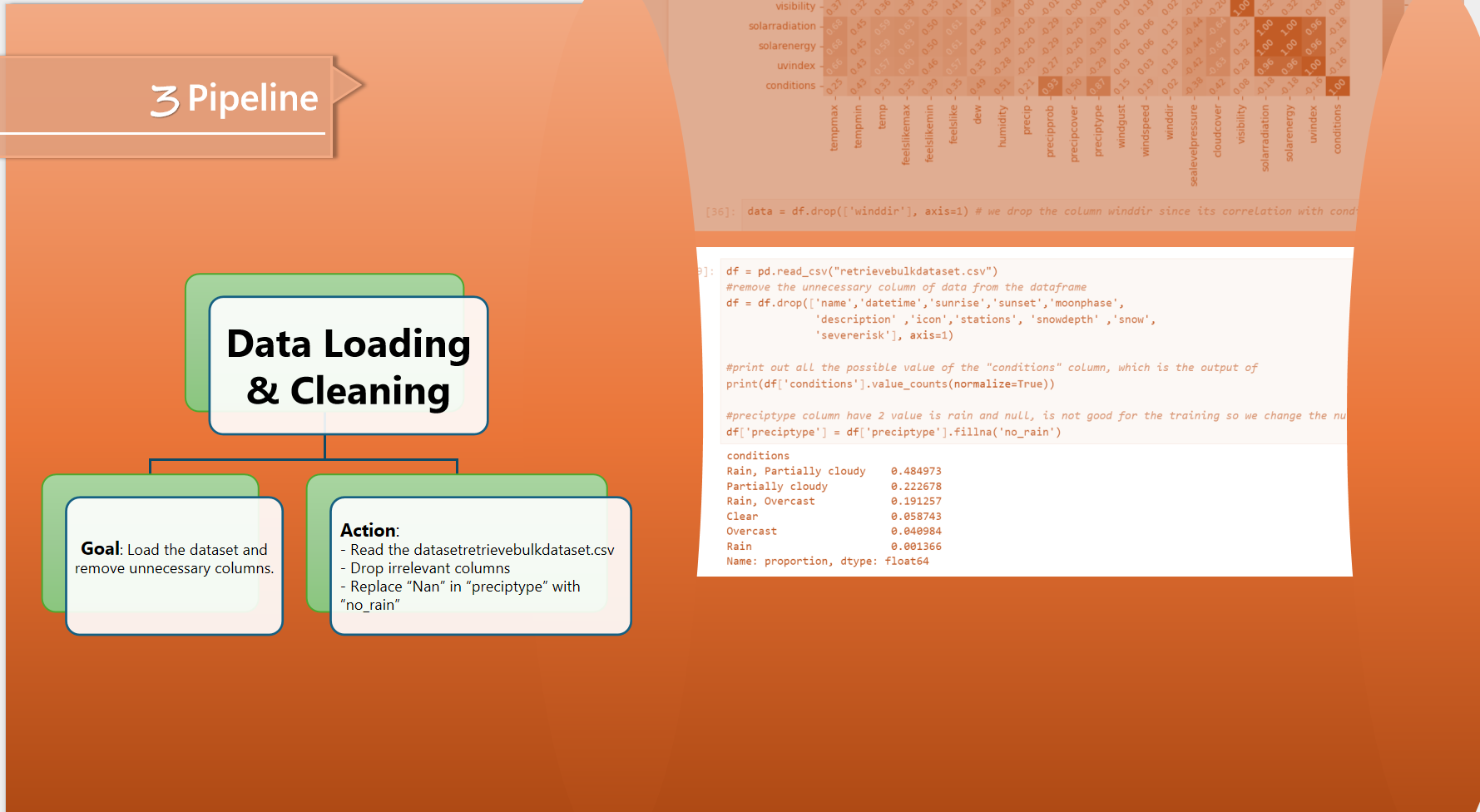


Our dataset about meteorological data collected from a weather station in Hanoi, Vietnam, covers the period from June 5, 2023, to June 5, 2025. The dataset includes daily measurements of key atmospheric variables, sourced from Visual Crossing Weather.



The raw data includes 732 samples and 33 data types for each sample.

In the following section, I will detail the data processing and modeling pipeline used to transform raw meteorological data into actionable weather predictions.

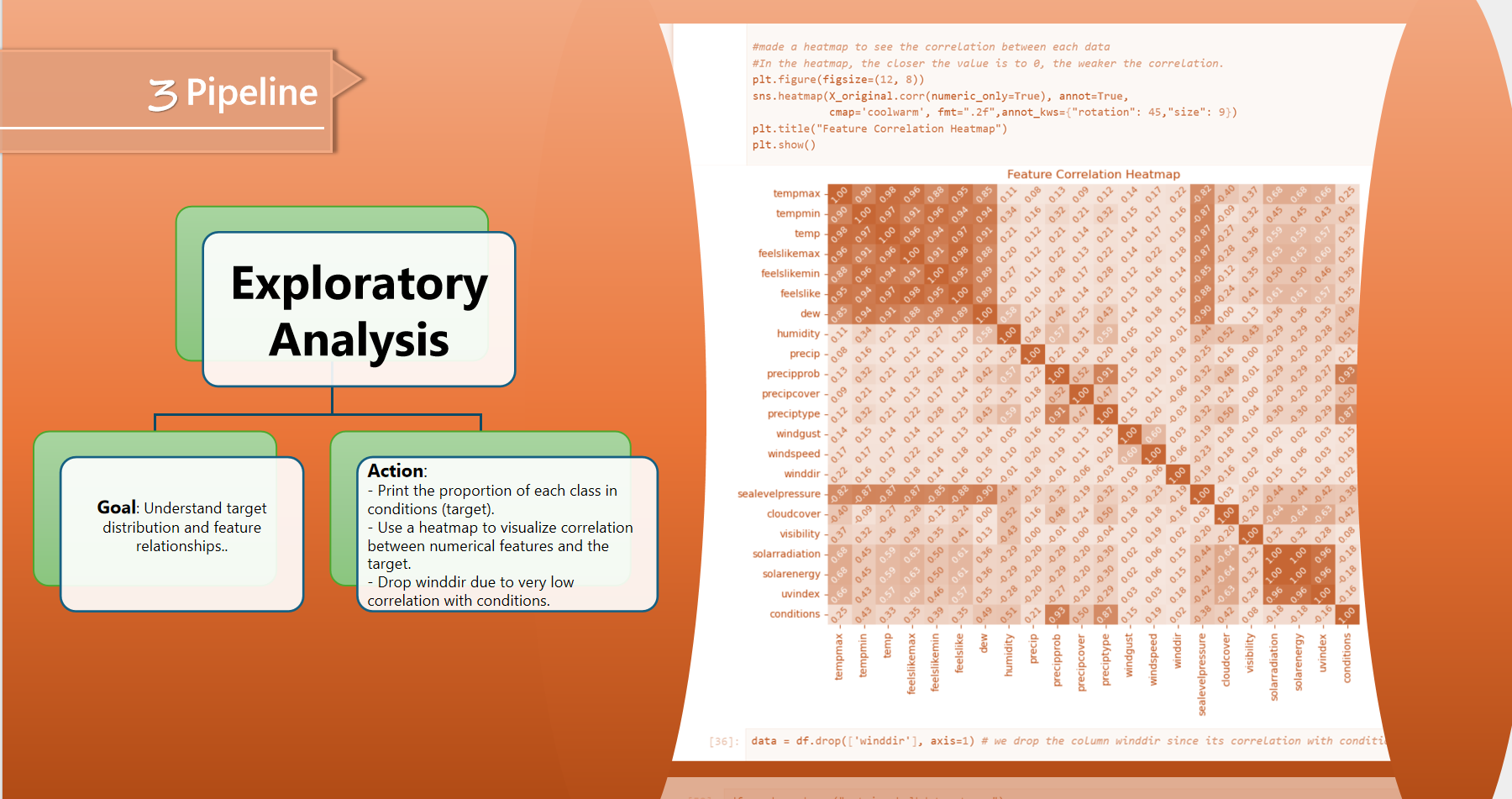


First, we read the dataset, and after that, we drop all the columns that do not affect our model. Like the data that is the same with every row, like name,datetime, station,... or the data that never appears, such as snow or snowdepth…

Since the preciptype column contains only 2 types of data is null and rain, we fill the null with no\_rain to train the model more easily.



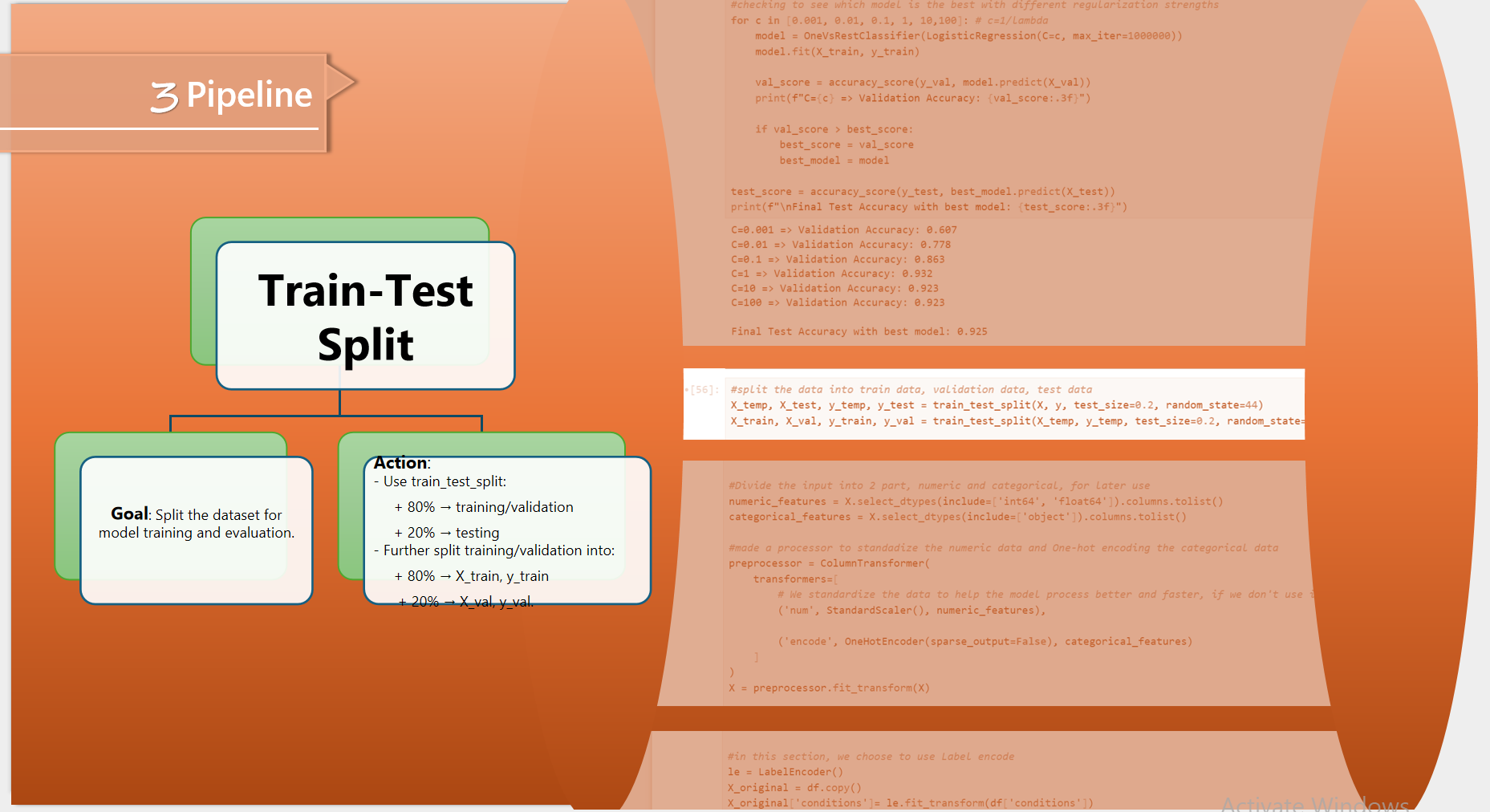
And to understand more about our data, we made a heatmap to visualize the correlation between input features and the output target. And to do this, since we have some data of a categorical type, we made a copy of the original dataframe and encoded the data with a categorical type into a numerical type with label encoding.



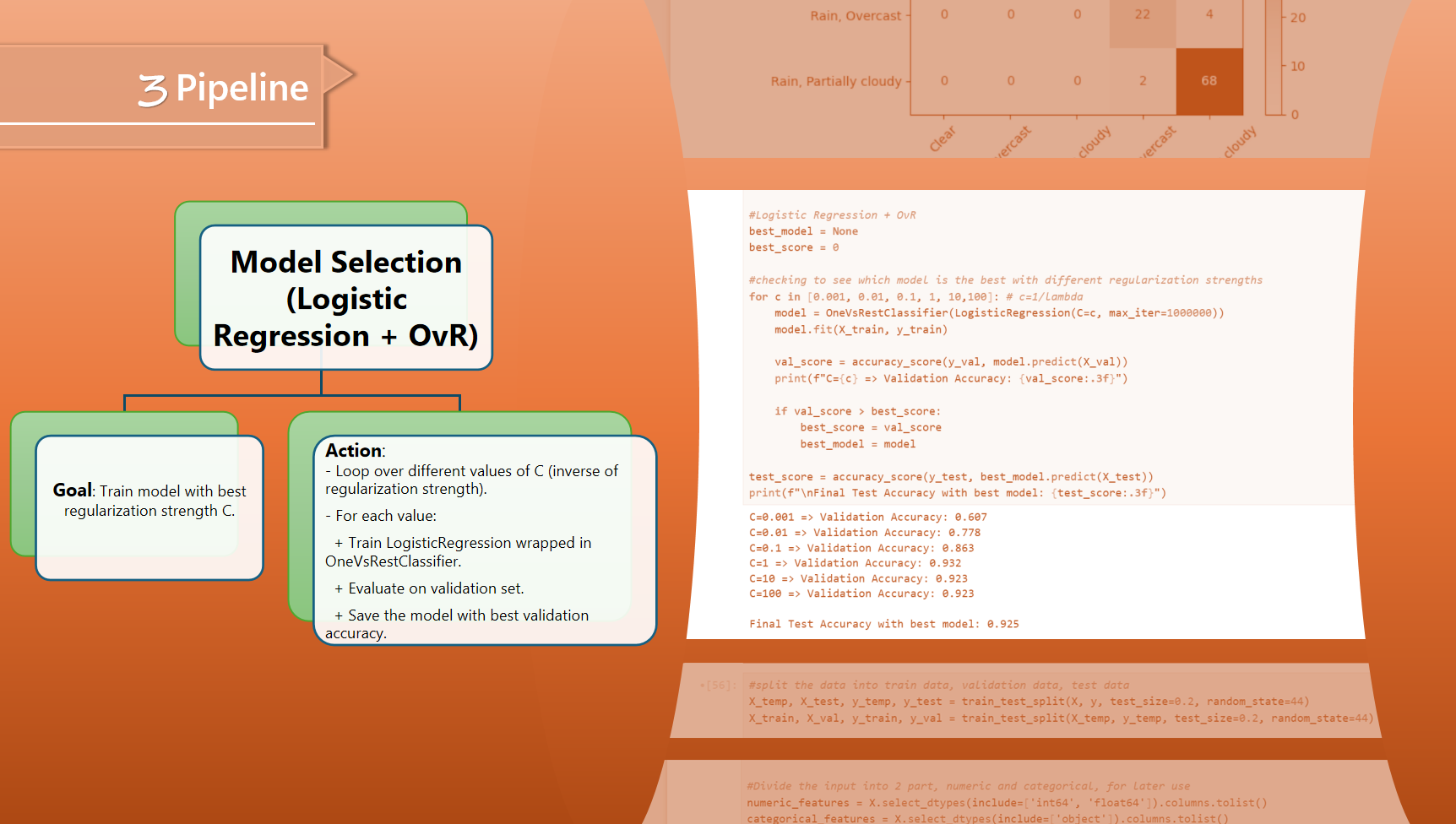
As we can see in the heatmap, the correlation between wind direction and weather conditions is very low, only 0.02, so we remove it for better performance.



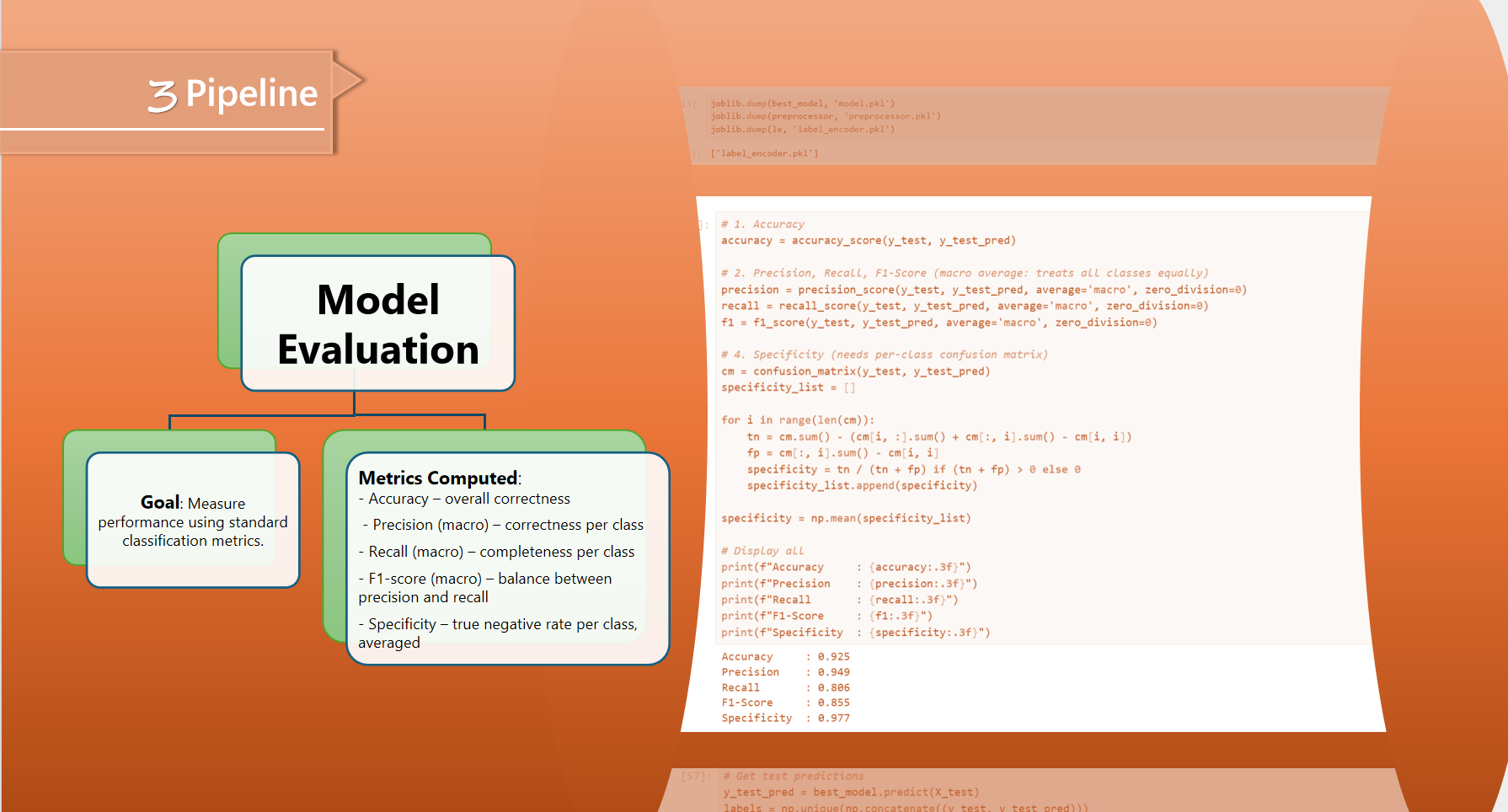
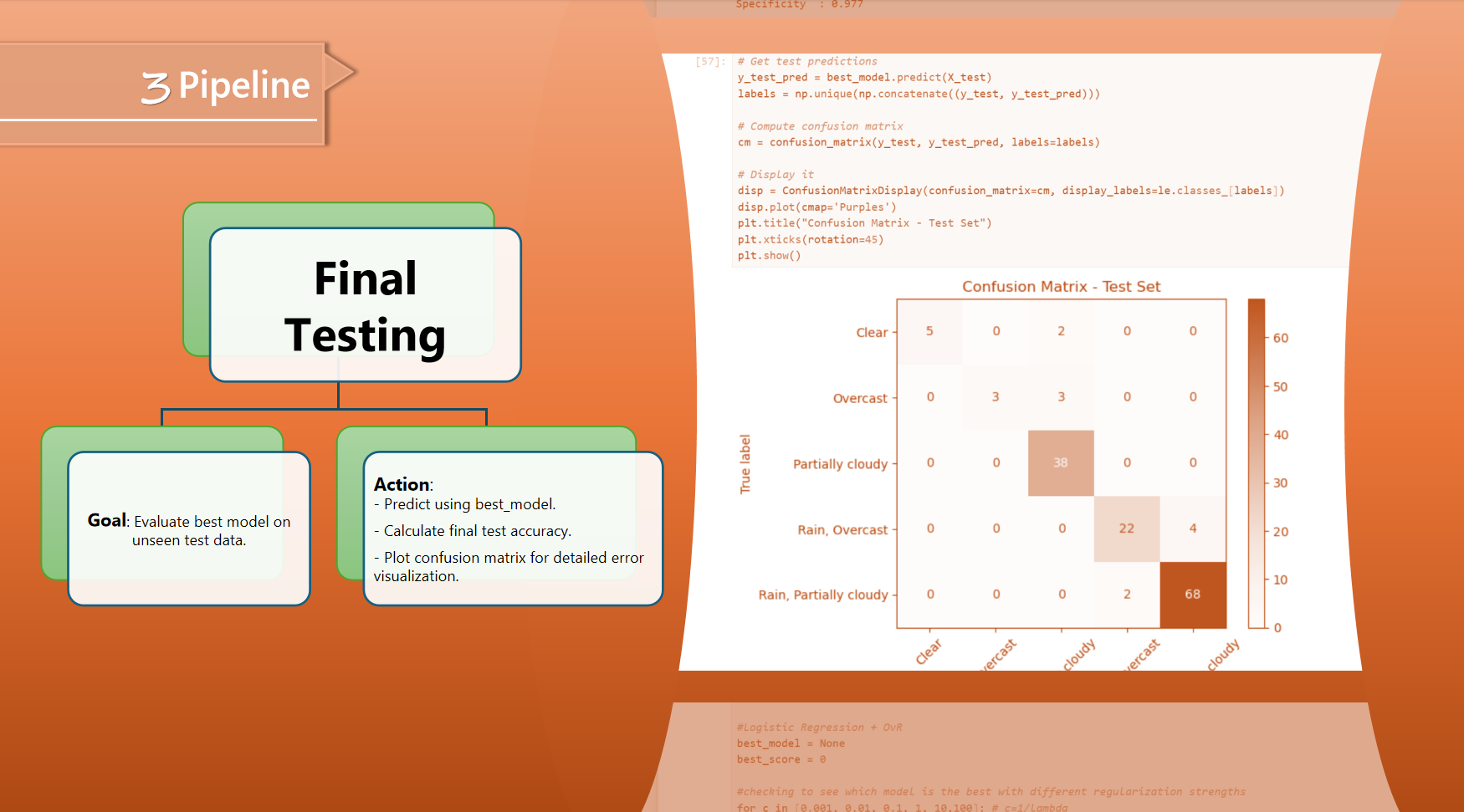
Next, we process the raw weather data into an optimized format. We split features into numerical and categorical types. Create a ColumnTransform to standardize the numerical feature and encode the categorical feature using a One-hot encoder. And apply that to get the input matrix X.

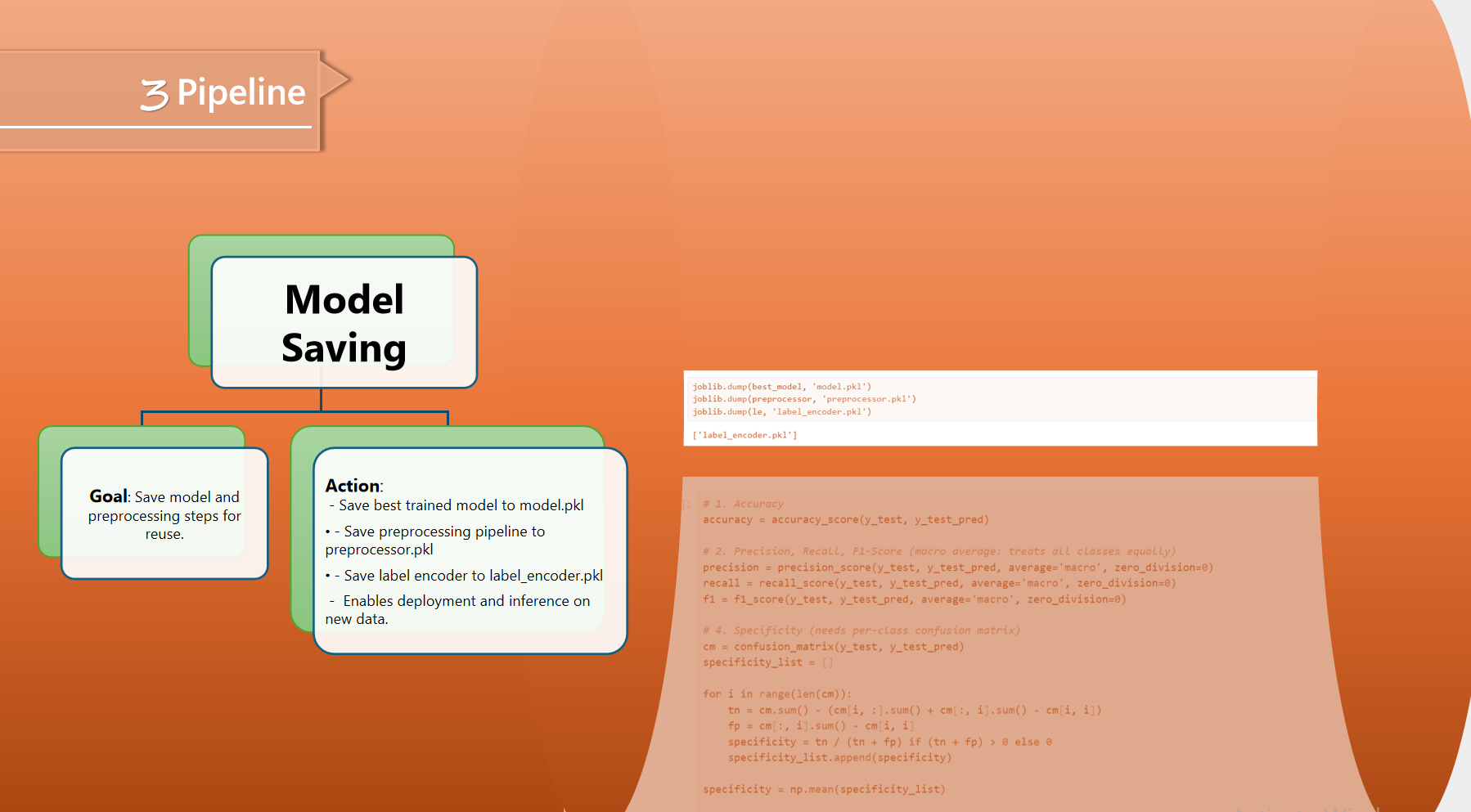


After processing the raw data, we split it into different sets of data with 64% for training, 16% for validating the model, and 20% for testing.



Just read everything in the slide (both are model evaluation)





same with above, read it…

then open the demo model to live test..