# How it Works

## Let's break down the code:

1. `import` statements:

- `streamlit`: Streamlit library for creating web applications.

- `numpy` (aliased as `np`): Library for numerical computations.

- `pandas` (aliased as `pd`): Library for data manipulation and analysis.

- `pickle`: Module for serializing and deserializing Python objects.

- `matplotlib.pyplot` (aliased as `plt`): Library for creating visualizations.

- `LabelEncoder` and `OneHotEncoder` from `sklearn.preprocessing`: These are used for encoding categorical variables.

2. `st.title('Customer Churn Model')`: Sets the title of the Streamlit app as "Customer Churn Model".

3. Markdown strings:

- `st.markdown("<h3></h3>", unsafe\_allow\_html=True)`: Inserts an empty heading.

- The multi-line markdown string provides an introduction to the Streamlit app, explaining its purpose and features.

4. `read\_data(data\_path)`: Function to read the dataset from the CSV file specified by `data\_path`. The data is then stored in the DataFrame `df\_original`.

5. Displaying Data:

- `df\_disp`: A copy of the original DataFrame for display.

- `table1`: Placeholder for displaying the DataFrame `df\_disp`.

6. Reading Clustering Centroids data:

- `read\_cent(data\_path)`: Function to read the clustering centroids from the CSV file specified by `data\_path`. The centroids are stored in the DataFrame `df\_cent`.

- `df\_cent\_disp`: A copy of the centroids DataFrame for display.

- `table2`: Placeholder for displaying the DataFrame `df\_cent\_disp`.

7. Loading the trained Random Forest model:

- The model is loaded from the pickle file "finalized\_model.sav" into the variable `rfc\_best`.

8. Filtering Data:

- Sidebar options for filtering data based on selected parameters.

- `grp`: List of selected parameters for filtering.

- Filtered DataFrame `df\_disp` based on user selections.

9. Customer Search:

- Allows the user to search for a specific customer record by entering the customer ID in the sidebar.

- The searched record is displayed using `searched\_df`.

10. Centroid Search:

- Allows the user to search for a cluster centroid by entering the cluster number in the sidebar.

- The searched centroid is displayed using `searched\_df`.

11. Customized Use Case:

- Sidebar options for customizing a use case by selecting parameters.

- The customized record is displayed using `custom\_df`.

12. `convert\_data(df\_churn)`: Function to preprocess the data for prediction. It handles missing values, encodes categorical variables, and prepares the data for the Random Forest model.

13. Prediction:

- Generates predictions for the churn probability using the trained Random Forest model.

- Displays the churn probability using a pie chart.