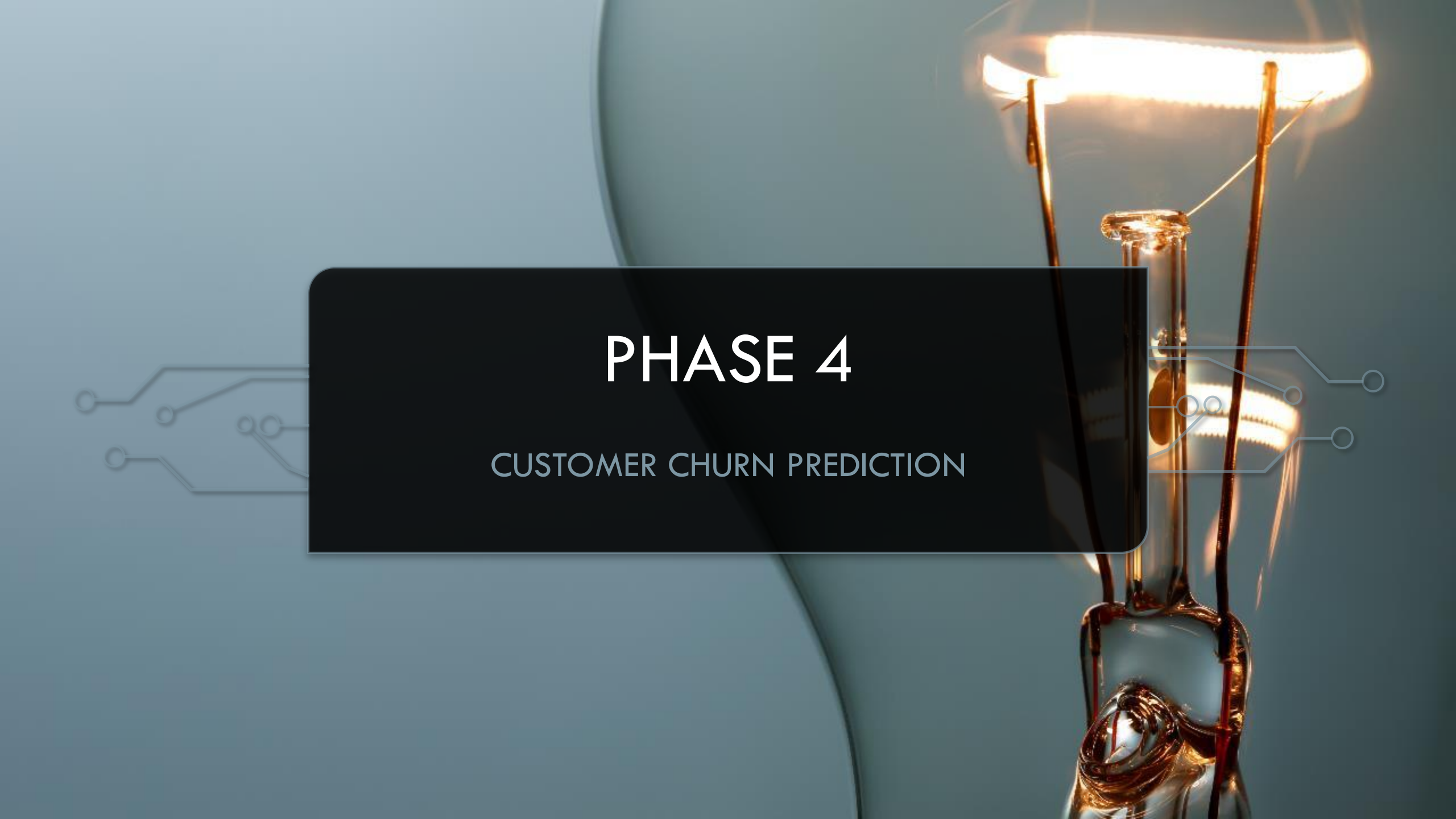
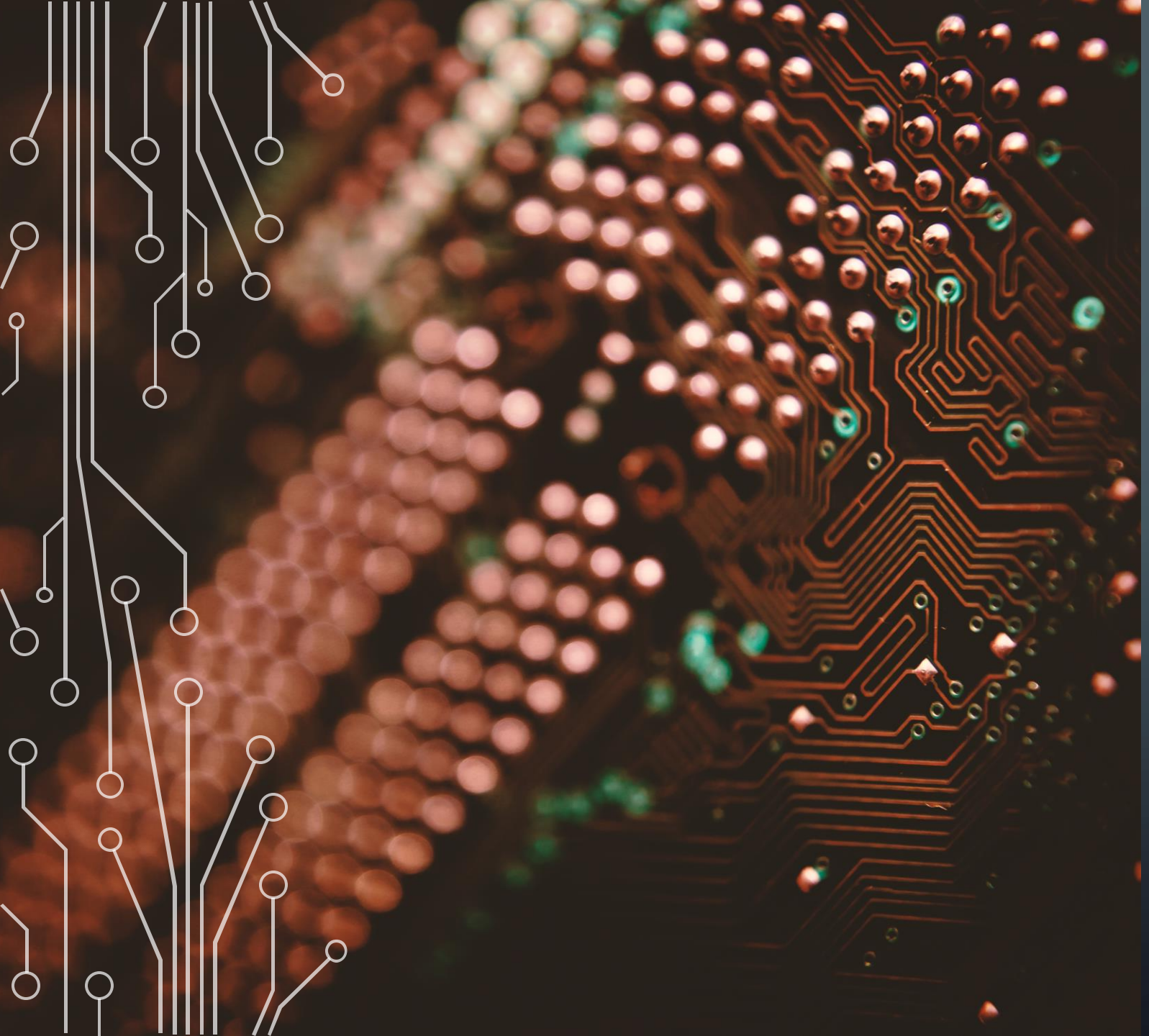


PHASE 4

CUSTOMER CHURN PREDICTION



EXPLANATION

THIS CODE IS FOR BUILDING A CLASSIFICATION MODEL TO PREDICT CUSTOMER CHURN BASED ON VARIOUS FEATURES. HERE'S A STEP-BY-STEP EXPLANATION OF WHAT THE CODE IS DOING:

1. It imports necessary libraries for data manipulation and machine learning: Pandas, NumPy, Matplotlib, and various modules from the scikit-learn library.

2. It reads a CSV file ('Telco_customer.csv') into a Pandas DataFrame called 'df' and displays the first few rows of the DataFrame using `df.head()`.

3. It drops the 'customerID' column from the DataFrame using `df = df.drop(['customerID'], axis=1)` and displays information about the DataFrame using `df.info()`.

4. It converts the 'TotalCharges' column to numeric values, handling any errors by coercing them using `pd.to_numeric(df.TotalCharges, errors='coerce')`.

5. It checks for and counts missing values in the DataFrame using `df.isnull().sum()`.

6. It removes rows where the 'tenure' column is equal to 0 using `df.drop(labels=df[df['tenure'] == 0].index, axis=0, inplace=True)`.

7. It fills missing values in the 'TotalCharges' column with the mean value of the column using `df.fillna(df["TotalCharges"].mean())`.

8. It maps the 'SeniorCitizen' column from binary values (0 and 1) to categorical values ('No' and 'Yes') using `df["SeniorCitizen"] = df["SeniorCitizen"].map({0: "No", 1: "Yes"})`.

9. It identifies numerical columns as 'tenure', 'MonthlyCharges', and 'TotalCharges'.

10. It defines a function 'object_to_int' to convert object-type columns to integers using label encoding, and it applies this function to the DataFrame using `df = df.apply(lambda x:object_to_int(x))`.

11. It creates a correlation matrix for the DataFrame and sorts the correlation values for the 'Churn' column in descending order.

12. It prepares the feature matrix 'X' by dropping the 'Churn' column, and the target vector 'y' is set as the 'Churn' column.

13. It splits the data into training and testing sets using `train_test_split`.

14. It standardizes numerical columns in the training and testing sets using the StandardScaler.

15. It defines the K-Nearest Neighbors (KNN) classifier, fits it to the training data, and makes predictions on the test data. It then calculates and prints the accuracy and classification report for KNN.

16. It defines and fits a Support Vector Machine (SVM) classifier and prints its accuracy and classification report.

17. It defines and fits a Random Forest classifier, makes predictions, and prints its accuracy and classification report.

18. It defines and fits a Logistic Regression model, calculates its accuracy, and prints the classification report.

19. Finally, it defines and fits a Decision Tree classifier, calculates its accuracy, and prints the classification report for the Decision Tree model.

In summary, this code loads and preprocesses customer data, encodes categorical features, scales numerical features, and trains various machine learning models (KNN, SVM, Random Forest, Logistic Regression, and Decision Tree) to predict customer churn, evaluating each model's performance.


THANK YOU