IMDB SCORE PREDICITION

TEAM MEMBER

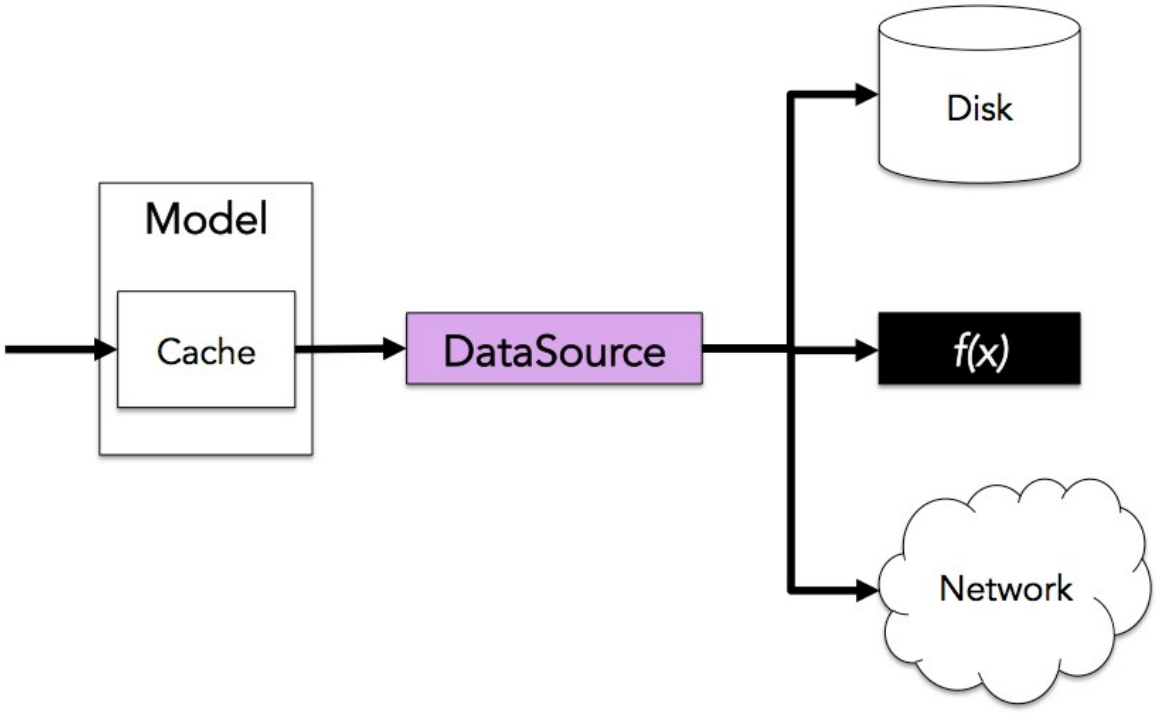
PHASE – 2 Submission Document

INTRODUCTION:

Problem definition is a critical step in the design thinking process. Design thinking is a human-centered approach to solving complex problems and creating innovative solutions. It involves a series of steps, and problem definition is one of the initial phases where you identify and understand the problem you're trying to address. In the first stage of design thinking, you seek to understand the needs, emotions, and experiences of the people you are designing for. This involves conducting research, interviews, and observations to gain insights into their perspective. This is where problem definition comes into play. In this phase, you synthesize the information gathered during the empathize stage and define the core problem or challenge. It's crucial to frame the problem from the user's perspective and express it in a way that is both specific and actionable. Once you have a well-defined problem statement, you move on to brainstorming and generating creative ideas for solving the problem. The goal is to think broadly and consider a wide range of potential solutions.

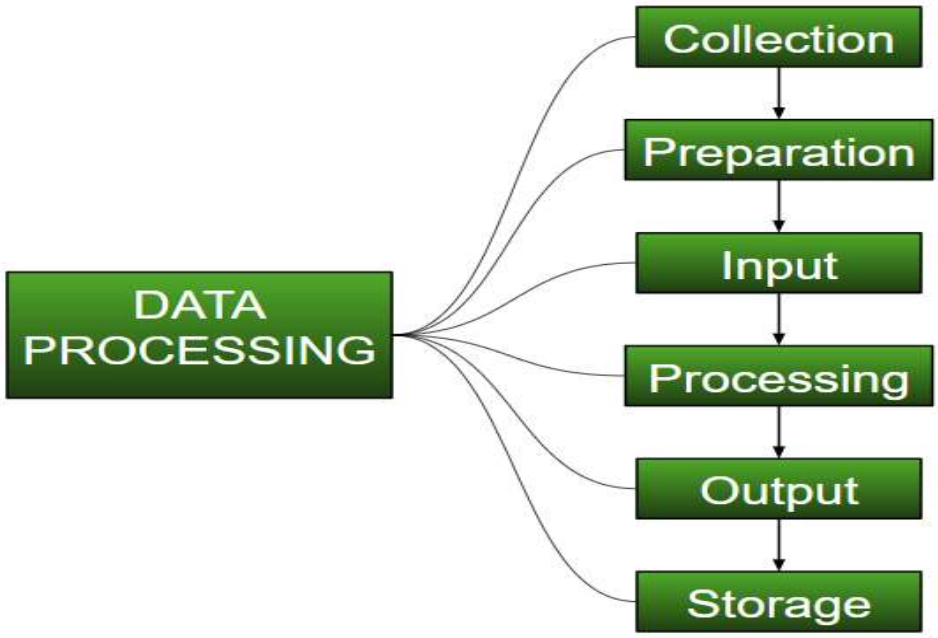
DATA SOURCE:

Websites like data.gov, the World Bank's data portal, and Kaggle provide access to a wide range of public datasets on topics such as demographics, economics, climate, and more. Many websites and online services offer APIs that allow you to programmatically access their data. For example, you can use APIs from social media platforms, weather services, financial data providers, and more. Government agencies often publish data on their websites, covering areas like population statistics, healthcare, education, and transportation.



DATA PREPROCESSING:

Data preprocessing involves various activities to transform raw data into a more useful and meaningful format. Below is a simplified example of data processing using Python, a popular programming language for data manipulation and analysis. In this



example, we'll assume that you have a dataset containing information about product sales and you want to perform some basic data processing tasks.

1. We import the `pandas` library, which is commonly used for data processing in Python.
2. We create a sample dataset (you would typically load data from an external source, such as a CSV file or a database).
3. We create a DataFrame named `df` to store and manipulate the data.
4. We calculate the `TotalSales` by multiplying the `UnitsSold` by a fixed price
5. We group the data by product name and calculate the total sales for each product.

6. We sort the products by total sales in descending order.
7. Finally, we display the processed dataset showing the total sales for each product

FEATURE ENGINEERING:

Feature engineering is a critical step in the process of preparing data for machine learning or data analysis. It involves creating new features or modifying existing ones to improve the performance of a machine learning model or to gain better insights from the data. Here's a simple example of feature engineering using Python's pandas library:

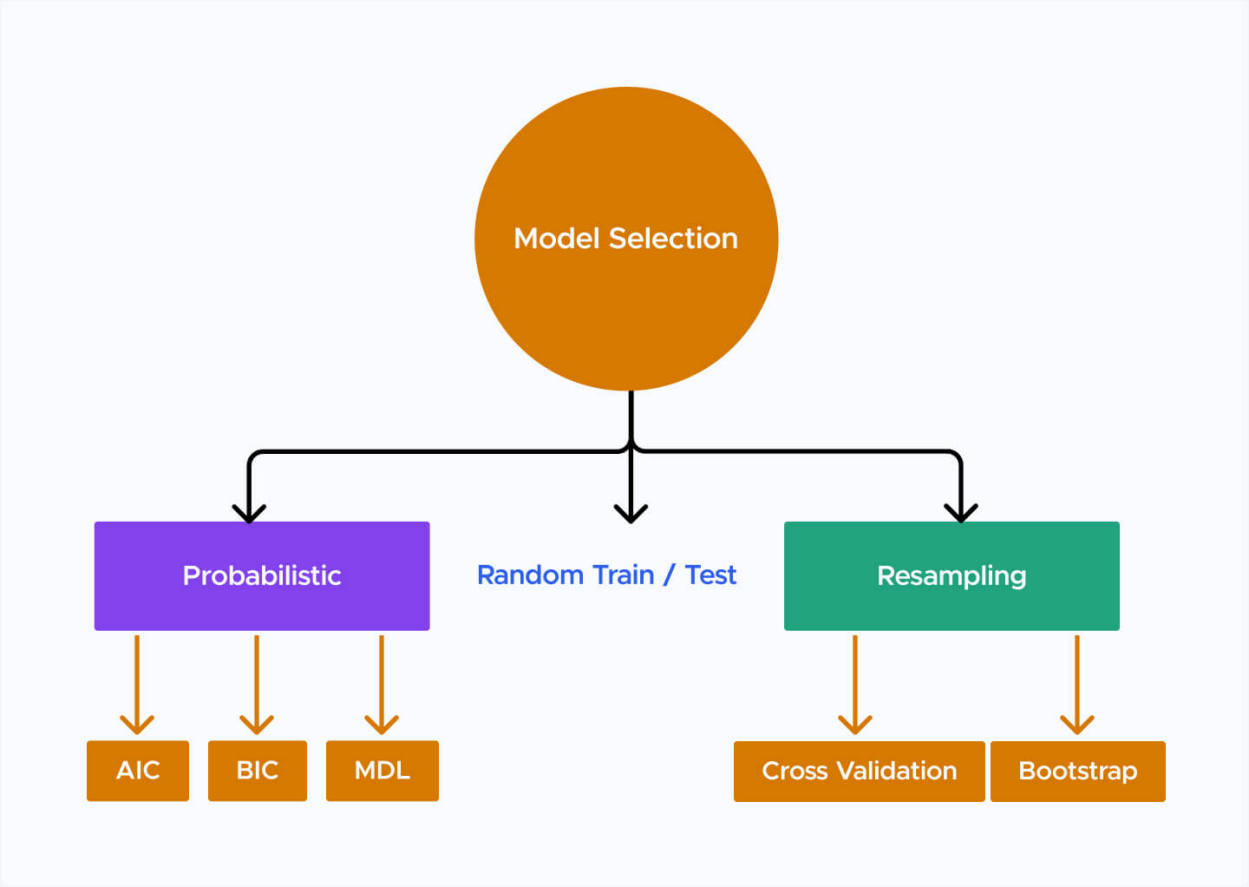
Suppose you have a dataset of e-commerce transactions and want to predict whether a customer will make a repeat purchase. You have a "PurchaseDate" column, and you want to create a feature that represents the time elapsed since the customer's last purchase. Here's how you can do it.

MODEL SELECTION:

Module selection, in the context of programming or software development, typically refers to the process of choosing and importing external libraries or modules to extend the functionality of your code. The specific modules you select depend on the programming language you're using and the requirements of your project. I'll provide an example of module selection in Python.

Python is known for its extensive ecosystem of libraries, and you can import modules to add various capabilities to your code. Here's an example of module selection in Python for a simple data analysis task.

1. We import the `pandas` module for data manipulation and analysis, the `numpy` module for numerical operations, and the `matplotlib.pyplot` module for data visualization.



2. We read data from a CSV file into a DataFrame using `pandas`.
3. We calculate the mean and standard deviation of a column using functions from the `numpy` module.
4. We create a simple bar chart to visualize the data using functions from the `matplotlib.pyplot` module.

MODEL TRAINING:

Model training is a crucial step in machine learning where you build and train a predictive model using your data. I'll provide a simple example of model training using Python and the scikit-learn library. In this example, we'll use a basic linear regression model to predict housing prices based on a dataset.Please note that this is a simplified example for illustration purposes. In a real-world scenario, data preprocessing, feature engineering, and model evaluation should be more comprehensive.

1. We import necessary libraries, including `pandas` for data handling, scikit-learn for machine learning, and metrics for model evaluation.
2. We load a dataset, split it into features (X) and the target variable (y), and further split it into training and testing sets.
3. We create a linear regression model using `LinearRegression` from scikit-learn and train it on the training data.
4. We make predictions on the test set and evaluate the model's performance using mean squared error (MSE) and the R-squared (R2) score.
5. Finally, we demonstrate how to use the trained model to make predictions on new data.

EVALUATION:

Evaluating a machine learning model is a critical step to assess its performance and determine how well it is doing on a particular task. Various evaluation metrics are available depending on the type of problem. I'll provide examples for both classification and regression problems using Python and scikit-learn.

Classification Evaluation:

Here's an example of evaluating a classification model using accuracy, precision, recall, and F1-score.

Regression Evaluation:

Here's an example of evaluating a regression model using mean squared error (MSE) and R-squared (R2) score: