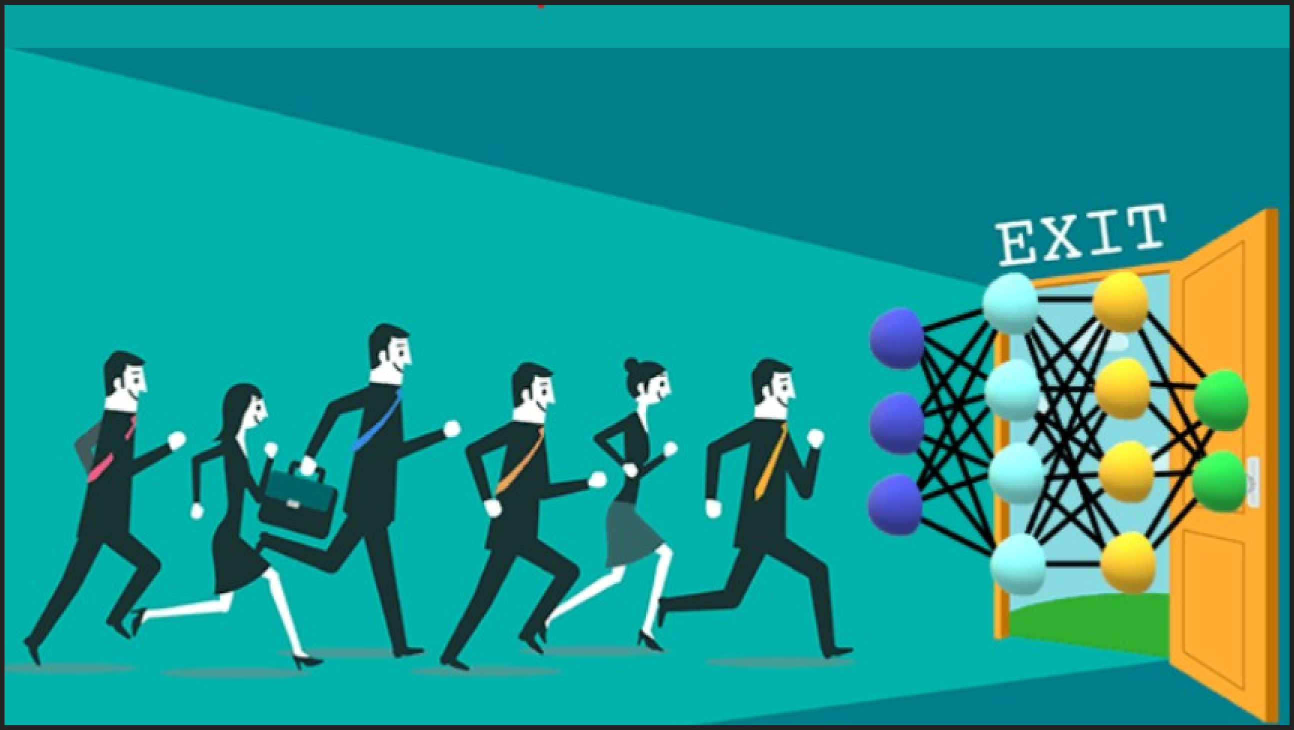
Title : Predicting User Response to New Features or Updates



Problem: Forecasting how users will react to new features or app updates.

Regression Task: Model the relationship between feature characteristics and user adoption, engagement, or feedback.

Algorithms Used: Linear Regression, Logistic Regression (for predicting adoption rates).

Data Used:

Feature characteristics.
Beta testing data.
User feedback surveys.

Training Data

Training data is the foundation upon which a machine learning model learns to perform its task. It's a labeled dataset, meaning each data point (e.g., an image, a text snippet, a set of measurements) is paired with the correct answer or target variable. The model analyzes patterns and relationships within this data to adjust its internal parameters, effectively learning the underlying function that maps inputs to outputs. The quality and representativeness of the training data are critical for the model's ability to generalize to new, unseen data.

Testing Data

Testing data is a separate, labeled dataset that the model has never seen during its training phase. Its purpose is to evaluate the model's performance on unseen data, providing an unbiased estimate of how well it will generalize in real-world scenarios. By comparing the model's predictions on the testing data to the true labels, we can assess its accuracy, identify potential overfitting (where the model performs well on training data but poorly on new data), and compare different models.

Data Dictionary

A data dictionary is a comprehensive document that describes the structure, content, and meaning of the data used in a project. For each variable or feature in the dataset, it typically includes information such as its name, data type, description, possible values or range, units of measurement (if applicable), and any known issues or transformations applied. The data dictionary serves as a crucial reference for anyone working with the data, ensuring clarity, consistency, and a shared understanding of the information being used to train and evaluate models.

Criteria
1. Data Understanding and Exploratory Data Analysis 20%
2. Data Preprocessing and Feature Engineering 25%
Weightage
Expectation Met :
1. Clearly defined the data sources and their relevance to the learning platform.
2. Demonstrated a strong understanding of the dataset(s) through relevant visualizations (e.g., distributions, relationships, missing values).
3. Identified key features and potential challenges (e.g., data quality issues, biases).
4. Provided insightful observations and summaries of the data that directly inform subsequent steps.
5. Justified the choice of exploratory techniques used and clearly explained the findings.
1. Addressed identified data quality issues (e.g., missing values, outliers) with appropriate techniques and justifications. 2. Performed relevant data transformations (e.g., scaling, encoding) suitable for the chosen modeling approach. Engineered new features that are well-motivated and potentially improve model performance or provide valuable insights for the platform.
3. Clearly documented all preprocessing steps and the reasoning behind them.
4. Demonstrated an understanding of how preprocessing choices impact the data and subsequent modeling.
1. Clearly defined the learning task (e.g., classification,
regression, recommendation) relevant to the e-learning platform.
2. Selected and implemented appropriate machine learning model(s) with clear justification.
Expectation not Met
1. Data sources are unclear or their relevance is not established. 2. Exploratory data analysis is superficial or lacks meaningful visualizations.
3. Key features and potential challenges are not identified or overlooked.
4. Observations and summaries are generic or do not connect to the project goals.
5. Exploratory techniques are poorly chosen or findings are not clearly explained.
1. Data quality issues are ignored or handled inappropriately. 2. Data transformations are missing or incorrectly applied. 3. Feature engineering is absent or lacks clear rationale and potential benefit.
4. Preprocessing steps are poorly documented or the reasoning is unclear.
5. Demonstrates a lack of understanding of the impact of preprocessing on the data.
1. The learning task is unclear or not relevant to the platform.
2. Model selection is inappropriate or lacks justification.
3. Properly split the data into training and evaluation sets (and 3. Data splitting is not performed correctly or is absent. potentially validation sets).
4. Used relevant evaluation metrics to assess model performance based on the defined learning task.
5. Analyzed and interpreted the model evaluation results, highlighting strengths and weaknesses.
6. Potentially explored different model parameters or architectures and justified the final model choice.
1. Clearly articulated actionable business recommendations for the e-learning platform based on the insights gained from the data analysis and modeling.
2. Recommendations are specific, measurable, achievable, relevant, and time-bound (SMART principles encouraged). Demonstrated a strong understanding of the e-learning platform's context and potential impact of the recommendations.
3. Prioritized recommendations based on potential value and feasibility.
4. Considered potential ethical implications or limitations of the recommendations.
1. Code is well-structured, organized, and easy to follow. Meaningful variable names and comments are used throughout the code.
2. Code is modular and functions are used effectively to promote reusability.

3. Adhered to consistent coding style conventions (e.g., PEP 8 for Python).
4. Project includes clear documentation (e.g., README file) outlining the project goals, data sources, steps taken, and how to run the code.
4. Evaluation metrics are irrelevant or not properly applied. 5. Model evaluation results are not analyzed or misinterpreted. 6. No exploration of different model options or parameter tuning is evident.
1. Business recommendations are vague, generic, or not directly linked to the project findings.
2. Recommendations lack practicality or feasibility.
3. Demonstrates a limited understanding of the e-learning platform's business context.
4. Recommendations are not prioritized or lack justification for their importance.
5. Ethical considerations or limitations are ignored.
1. Code is poorly structured, disorganized, or difficult to understand.
2. Variable names are unclear or comments are lacking.
3. Code is monolithic and lacks modularity.
4. Coding style is inconsistent or deviates significantly from standard conventions.
5. Project lacks adequate documentation, making it difficult to understand or reproduce the work.
3. Model Building & Evaluation
30%
4. Business Recommendations
15%
5. Coding Guidelines and Standards
10%

Example:

Predict the adoption rate of a new feature based on its complexity and perceived value.
Key Considerations in B2C SaaS/Consumer Tech Regression:

Large Datasets: B2C SaaS often generates massive amounts of user data, requiring scalable regression techniques.

User Privacy: Handling user data requires strict adherence to privacy regulations and ethical considerations.

Real-time Analysis: In some cases, real-time analysis of user data is required to provide personalized experiences or detect anomalies.

A/B Testing: Regression can be combined with A/B testing to measure the impact of different product changes or marketing strategies.

By effectively using regression analysis, B2C SaaS and consumer tech companies can gain valuable insights into user behavior, optimize their products and services, and drive business growth.

Project Submission Guidelines

Deadline and Penalties:

Submissions received within one week following the stated deadline will incur a 30% penalty.
Submissions received more than one week after the stated deadline will not be accepted.

Submission Format:

Submit one text file (preferably with a .txt extension).
The text file must contain a single link to a GitHub repository.

GitHub Repository Contents:

The GitHub repository should include:
One well-commented Jupyter Notebook (.ipynb file) containing the project code and analysis.
One PDF document (.pdf file) containing comprehensive answers to all subjective questions.
A README.md file providing a clear overview of the project, instructions for running the notebook, and any relevant information.

Submission Method:

Upload the text file containing the GitHub repository link via the designated submission platform.