Project Title	Customer conversion prediction - Insurance
Skills take away From This Project	**Data Preprocessing and Cleaning,** **Exploratory Data Analysis (EDA),** **ML Model Building and Training using Python,** **Model Evaluation,** **Deployment**
Domain	**Finance**

Problem Statement:

In the insurance industry, acquiring new customers and converting leads into sales is crucial for business growth. The dataset provided contains information about a series of marketing calls made to potential customers by an insurance company. The goal is to predict whether a customer will subscribe to an insurance policy based on various attributes of the customer and details of the marketing interactions.

Data Set:

1. **age**: Age of the customer.
2. **job**: Type of job the customer holds.
3. **marital**: Marital status of the customer.
4. **education_qual**: Educational qualification of the customer.
5. **call_type**: Type of marketing call.
6. **day**: Day of the month when the call was made.
7. **mon**: Month when the call was made.
8. **dur**: Duration of the call in seconds.
9. **num_calls**: Number of calls made to the customer before this interaction.
10. **prev_outcome**: Outcome of the previous marketing campaign.
11. **y**: Whether the customer subscribed to the insurance policy (target variable).

Data -
https://raw.githubusercontent.com/GuviMentor88/Training-Datasets/main/insurance_dataset.csv

Approach:

1. **Data Preprocessing**: Clean the data, handle missing values, Scale or Normalize Features(if needed), and encode categorical variables.
2. **Exploratory Data Analysis (EDA)**: Understand the distribution of features, identify patterns, and explore relationships between features and the target variable.
3. **Dataset Balancing:** Is the target feature balanced or not. The imbalance of the target variable requires a careful consideration in the prediction stage in this project.
4. **Model Building**: Train various machine learning models to predict the target variable and segment customers.
   - Find the correlation between features.
   - Split the dataset into training and testing/validation sets.
   - Train and evaluate **all the different classification models**, such as Logistic Regression, Decision Trees, Random Forest, Gradient Boosting etc.
   - Train the **clustering model**: Customer segmentation based on different groups.
   - **Hyperparameter Tuning:** Optimize model hyperparameters using techniques such as cross-validation and grid search or Random search to find the best-performing model.
   - Interpret the model results and assess its performance based on the defined problem statement.

5. **Model Evaluation**: Evaluate the models using appropriate metrics (e.g., Accuracy, Precision, Recall, F1-Score, ROC-AUC) and select the best-performing model.
6. **Feature Importance**: Identify key features that have a significant impact on the target variable.
7. **Model Deployment**: Deploy the final model for use in real-time customer conversion prediction.
8. **Dump model:** Use pickle module to dump and load models for future use.

Results:

1) Exploring skewness and outliers in the dataset.
2) Transform the data into a suitable format and perform any necessary cleaning and pre-processing steps.
3) ML Classification model which predicts Status: YES or NO.
4) ML model for Customer Segmentation:

Project Evaluation metrics:

- You are supposed to write a code in a modular fashion (**in functional blocks**)
- Maintainable: It can be maintained, even as your codebase grows.
- Portable: It works the same in every environment (operating system)
- You have to maintain your code on **GitHub**.(Mandatory)
- You have to keep your **GitHub** repo public so that anyone can check your code.(Mandatory)
- Proper readme file you have to maintain for any project development(Mandatory)
- You should include basic workflow and execution of the entire project in the readme file on **GitHub** (Mandatory)
- Follow the coding standards: https://www.python.org/dev/peps/pep-0008/
- You need to Create a Demo video of your working model and post in **LinkedIn(Mandatory)**

Project Deliverables:

1.**Data Files**:

- Data files used in the project
- If data cannot be shared, provide information on how to access the data you have used.

2. **Source Code**:

- All scripts and code files used in the project.
- A file or Jupyter Notebook that shows your project in action.
- Ensure the code is well-organized and properly commented for clarity and maintainability.

3. **Documentation - README File:**

- The name of your project and what it's about.
- How to set up everything and run your code.
- A quick overview of how your project is organized.
- Any tools or libraries needed to run your code.

Project Guidelines:

1. Coding Standards

- **Readability**: Write clean and readable code. Use meaningful variable and function names.
- **Comments**: Add comments to explain complex logic or important sections of your code.
- **Consistency**: Stick to a consistent coding style. Use tools like linters (e.g., PEP8 for Python) to maintain consistency.
- **Modularity**: Break your code into functions and modules to make it reusable and easier to understand.
- **Error Handling**: Implement proper error handling to make your code robust and user-friendly.
- **Documentation**: Document your functions and classes using docstrings. Include information about inputs, outputs, and any exceptions.

2. Version Control

- **Use Git**: Use Git for version control to keep track of changes and collaborate with others.
- **Regular Commits**: Commit your code regularly with meaningful commit messages. This helps track progress and rollback changes if needed.
- **Branching**: Use branches for new features or bug fixes. This keeps the main branch stable.
- **Pull Requests**: Use pull requests for code reviews before merging branches. This ensures code quality and catches potential issues early.
- **.gitignore**: Use a .gitignore file to exclude unnecessary files from the repository (e.g., compiled files, temporary files, environment-specific configurations).

Timeline:

Two Weeks.

PROJECT DOUBT CLARIFICATION SESSION (PROJECT AND CLASS DOUBTS)

About Session: The Project Doubt Clarification Session is a helpful resource for resolving questions and concerns about projects and class topics. It provides support in understanding project requirements, addressing code issues, and clarifying class concepts. The session aims to enhance comprehension and provide guidance to overcome challenges effectively.
Note: Book the slot at least before 12:00 Pm on the same day

Timing: Tuesday, Thursday, Saturday (5:00PM to 7:00PM)

Booking link :https://forms.gle/XC553oSbMJ2Gcfug9

LIVE EVALUATION SESSION (CAPSTONE AND FINAL PROJECT)

About Session: The Live Evaluation Session for Capstone and Final Projects allows participants to showcase their projects and receive real-time feedback for improvement. It assesses project quality and provides an opportunity for discussion and evaluation.
Note: This form will Open on Saturday and Sunday Only on Every Week

Timing: Monday-Saturday (11:30PM to 12:30PM)

Booking link : https://forms.gle/1m2Gsro41fLtZurRA