Data Science Canvas			**Project:**	**Predictive Delivery Management Engine**			
			Team:	Kashinath Alias Kapil Naik, Pranav N, Atreyee Mondal, Sujith Shetty			
Problem Statement				**Execution & Evaluation**		**Data Collection & Preparation**	
Business Case & Value Added Food delivery ETAs are inaccurate due to driver performance variability, reducing customer satisfaction and causing inefficient resource utilization. <u>Value Added:</u> Driver profiling improves ETA accuracy, enhances customer trust, enables performance-based incentives, optimizes assignments, reduces delays, and boosts operational efficiency.	**Model Selection** It is a regression problem: Linear Regression, Random Forests , Gradient Boosting, Stacking Regression, MLPRegressor, ANN	**Model Requirements** **Linear Regression:** Features should have a linear relationship with the target; minimal multicollinearity; outliers should be handled; **Random Forests: :** Ensure sufficient data to avoid overfitting; **Gradient Boosting Machines (GBM):** sufficient data needed for sequential learning;	**Skills** Python programming, data preprocessing, EDA Analysis, Model Selection/training, Validation, data visualization techniques	**Model Evaluation** MAE,MSE,R2, RMSE,	**Data Storytelling** Any doorstep delivery business can leverage our Predictive Delivery Management Engine to gain actionable insights through driver performance profiling.	**Data Selection & Cleansing** We have public data available, need pre-processing	**Data Collection** Based on the coordinates of the restaurant and delivery location, we calculate the distance between the two points.
Data Landscape The available dataset includes order and driver details such as *ID, Delivery_person_ID, Delivery_person_Age, Delivery_person_Ratings, restaurant and delivery coordinates, Type_of_order, Type_of_vehicle, and Time_taken(min)*. These provide the foundation for tracking driver performance and analyzing delivery durations. Driver efficiency can then be calculated by comparing the predicted time against the actual Time_taken(min), highlighting overperformance or underperformance.		**Software & Libraries** pandas – data preprocessing and manipulation NumPy – numerical computations scikit-learn – Linear Regression, Decision Trees, Random Forests, Gradient Boosting XGBoost / LightGBM – advanced gradient boosting models Tensorflow geopy.distance Matplotlib / Seaborn – data visualization and analysis				**Data Integration** We are dealing with single source of data.	**Explorative Data Analysis** Univariate Analysis Bivariate Analysis Multivariate Analysis