1. Currently, there are so many machine learning paradigms used by data scientists to solve some real-life problems in different conditions, thus categorizing in some subcategories:
a) What are supervised and unsupervised learning?

Supervised learning
The algorithm is trained on a labeled dataset. This means that each training example is paired with an output label. The goal is to learn a mapping from the input variables (features) to the output variable (label)

Unsupervised learning
the algorithm is trained on an unlabeled dataset. There are no predefined output labels. The goal is to infer the natural structure within a set of data points. This can involve clustering the data into groups or reducing the data to a lower dimensional space.

b) What is the main different key features of those paradigm in 1.a)?

Labels: Supervised learning uses labeled data, while unsupervised learning uses unlabeled data.
Goals: Supervised learning aims to predict an outcome based on input data, whereas unsupervised learning aims to uncover hidden patterns in input data.

c) How does semi-supervised work? And elaborate with implementation on at least 1 example in real-life case!

Semi-supervised learning leverages both labeled and unlabeled data to improve learning accuracy and generalize better

Basic Concept:

1. **Initial Training**: A model is initially trained on a small labeled dataset.
2. **Label Propagation**: The trained model is used to predict labels for the unlabeled data.
3. **Re-Training**: The model is re-trained using the combination of labeled data and the newly labeled data (initially unlabeled data with predicted labels).
4. **Iteration**: Steps 2 and 3 are repeated, improving the model incrementally as more unlabeled data is integrated into the training set.

Implementation Code:
https://github.com/fadilrisdian/insignia-ds-assesment-2024/blob/main/1.py

2. The most crucial component for building any machine learning model is data. Just as the oil, data has to be processed before being used, so:
a) Describe full-cycle of data flow that you had experienced on also provide tech-stack that you used for each step! And please elaborate the methods or algorithms you used for building that ML model and why did you choose those methods/algorithms?

1. Data Collection

Data is collected from various sources from other departments such as daily consultations, website visitors, and Google ads. These data points are stored in a PostgreSQL database and regularly extracted, transformed, and loaded (ETL) into a centralized data warehouse using Apache Airflow.

Tech Stack:
- Data Source: PostgreSQL, Google Ads, Google Analytics
- ETL Tools: Apache Airflow for scheduling and automating ETL processes
- Storage: Google Cloud Storage for raw data storage

2. Data Preprocessing
Data preprocessing involves cleaning, transforming, and organizing the data into a suitable format for analysis. This step includes handling missing values, normalizing data, and feature engineering (e.g., creating time-based features such as day of the week, holidays, etc.).

Tech Stack:
- Languages: Python
- Libraries: pandas, NumPy

3. Exploratory Data Analysis (EDA)
EDA is performed to understand the patterns, trends, and anomalies in the data. Visualization tools like Matplotlib and Seaborn are used to create graphs and plots to identify seasonality, trends, and potential outliers.

Tech Stack:
- Languages: Python,
- Libraries: Matplotlib, Seaborn, statsmodels

4. Model Selection and Training
For forecasting daily consultations, we will use forecasts with exogenous variables. The models considered include, Autoregressive (AR), Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), Linear Regression, Exponential Smoothing (ES), and XGBoost

Reasons for Choosing XGBoost and Prophet:
1. These models consistently perform others machine learning models
2. XGBoost is particularly effective in capturing complex relationships between variables (exogonous variables from many sources)
3. Facebook Prophet is designed to handle time series data with seasonality, our data show weekly pattern

Tech Stack:
- Libraries: XGBoost, Facebook Prophet, sci-kit-learn, statsmodels

5. Model Evaluation
The models are evaluated using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Cross-validation techniques are also employed to ensure the robustness and reliability of the models.

Tech Stack:
- Libraries: sci-kit-learn

6. Model Deployment
The trained models are deployed as a Streamlit web application. Docker is used to containerize the application.

7. Weekly Retraining and Model Selection
A scheduled job is set up in Apache Airflow to retrain the models every week using the latest data. The retraining process involves:
1. Extracting the latest data from the Google Cloud storage.
2. Preprocessing the data.
3. Training the XGBoost, and Facebook Prophet models.
4. Evaluating the models using RMSE.
5. Track models performance using MLflow
6. Selecting the model with the lowest RMSE as the best model for that week.

Tech Stack:
Scheduler: Apache Airflow
Libraries: XGBoost, Facebook Prophet, scikit-learn, MLflow

8. Automated Inference
Automated inference is set up to run daily predictions using the best model selected based on the RMSE. This involves:
1. Using Apache Airflow to schedule a daily job.
2. Loading the best model from the previous week's retraining process.
3. Making predictions for the upcoming days.
4. Storing the predictions in a Google Cloud storage
5. Show prediction data using Google Looker Studio

Tech Stack:
- Scheduler: Apache Airflow
- Libraries: scikit-learn
- Storage: Google Cloud Storage
- Dashboard: Google Looker Studio

b) In your opinion, if limited only 3 most important things as the reason of why you will (or won't) pick some of data as your oil to develop your own ML model, what are those 3 things will be? Please elaborate!

1. Data Quality
Good quality data is crucial for building reliable machine learning models. The data should be accurate, complete, and consistent, helping the model learn correct patterns and avoid errors. It should also be relevant to the problem. For example, clean and well-structured

past consultation data directly impact the accuracy of future predictions.

2. Data Volume
The amount of data is important for training effective machine learning models. More data improves performance by providing a wider range of examples, reducing overfitting. Larger datasets also allow for better feature creation. For instance, several years of historical consultation data is more beneficial than a few months.

3. Data Representativeness
Data should accurately represent real-world scenarios. This helps the model generalize and avoid biases. For time series data, capturing seasonal patterns and trends is crucial. For example, having data from all times of the day, days of the week, and months of the year ensures reliable forecasts for daily consultations.

c) If any non-data geeks ask for explanation, how do you tell them what are training set, test set, and validation set used for? And why do they need to be split that way?

Training Set
The training set is a portion of your data that you use to teach your machine-learning model. It's like practicing with examples you already know, so the model can learn patterns and relationships.

Test Set
The test set is another portion of your data that you use to evaluate how well your model performs on new, unseen data. It's like giving the model a final exam to see if it has learned the material correctly. This set is only used after the model has been trained.

Validation Set
The validation set is used during the training process to adjust and improve the model. It's like taking practice tests to figure out the best study methods. This helps improve the model's settings and choose the best version before the final evaluation.

Why Split This Way?
Splitting the data into training, test, and validation sets helps the model learn patterns without just memorizing the data. This ensures the model performs well on new data. The validation set helps fine-tune the model, making sure it's the best version before final testing.

3. Given data below:

houseId	length	lengthUnit	width	widthUnit	isCarport	price	notes
1	20	meter	10	meter	1	IDR 5 Billion	TRAINING DATA
2	40	meter	20	meter	0	IDR 18 Billion	TRAINING DATA
3	3000	centimeter	2000	centimeter	1	IDR 13 Billion	TRAINING DATA
4	1000	centimeter	3000	centimeter	0	IDR 6 Billion	TRAINING DATA
5	20	meter	50	meter	1	IDR 21 Billion	TRAINING DATA
6	50	meter	10	meter	0	IDR 11 Billion	TEST DATA
7	20	meter	20	meter	1	IDR 8 Billion	TEST DATA

a) For given data above, process the data and produce a relevant Exploratory Data Analysis (EDA) summary! Then choose an algorithm or a method on machine learning which could predict/estimate house price followed by feature selection or feature engineering!

	houseId	length	lengthUnit	width	widthUnit	isCarport	price	notes	length_m	width_m	area_sqm	price_numeric
0	1	20	meter	10	meter	1	IDR 5 Billion	TRAINING DATA	20.0	10.0	200.0	5.0
1	2	40	meter	20	meter	0	IDR 18 Billion	TRAINING DATA	40.0	20.0	800.0	18.0
2	3	3000	centimeter	2000	centimeter	1	IDR 13 Billion	TRAINING DATA	30.0	20.0	600.0	13.0
3	4	1000	centimeter	3000	centimeter	0	IDR 6 Billion	TRAINING DATA	10.0	30.0	300.0	6.0
4	5	20	meter	50	meter	1	IDR 21 Billion	TRAINING DATA	20.0	50.0	1000.0	21.0
5	6	50	meter	10	meter	0	IDR 11 Billion	TEST DATA	50.0	10.0	500.0	11.0
6	7	20	meter	20	meter	1	IDR 8 Billion	TEST DATA	20.0	20.0	400.0	8.0

Implementation Code:
https://github.com/fadilrisdian/insignia-ds-assesment-2024/blob/main/3.ipynb

b) Describe step-by-step of iterative process how to build the model in terms of mathematical equations, based on chosen algorithm/method on 3.a) and using ONLY GIVEN DATA ABOVE (data on question 3.), and elaborate as detail as possibe!

Step 1: Formulate the Problem
We aim to predict the house prices (Y) based on the area of the property $(X1)$ and the presence of a carport $(X2)$.

Step 2: Prepare the Data
We need to clean and preprocess the data. We already converted length and width to meters, calculated the area, and converted the price to numeric.

Step 3: Define the Linear Regression Model
Linear Regression assumes a linear relationship between the input variables and the output variable. The model can be defined as:
$Y = \beta 0 + \beta 1 X1 + \beta 2 X2 + \varepsilon$

- Y is the target variable (house price).
- $\beta 0$ is the intercept.
- $\beta 1$ and $\beta 2$ are the coefficients for the features (area and carport).
- $X1$ is the area of the property.
- $X2$ is the presence of a carport (1 if present, 0 if not).
- ε is the error term.

Step 4: Estimate the Model Parameters

We estimate the parameters ($\beta 0$,$\beta 1$, $\beta 2$) using the Ordinary Least Squares (OLS) method. The goal is to minimize the sum of squared errors (SSE):

$SSE = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$
$\hat{Y}_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$

The parameters are calculated using the following normal equations:

$$\beta = (X^T X)^{-1} X^T Y$$

Step 5: Train the Model

Using the training data, we solve for β to obtain the best-fit line.

Step 6: Evaluate the Model

We evaluate the model using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2):

MAE:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$$

MSE:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

R-Squared:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}$$

Step-by-Step Iterative Process

1. **Initialization:**
- Start with initial guesses for ($\beta 0$,$\beta 1$, $\beta 2$) (usually set to 0).

2. **Iteration:**
- Compute the predicted prices (Y^) using current parameter values:

  $Y = \beta 0 + \beta 1 X1 + \beta 2 X2 + \varepsilon$

- Calculate the error (ε):

  $\varepsilon = Y - Y^{\wedge}$

- Update the parameters using the gradients of the SSE concerning each parameter:

$$\beta_0^{new} = \beta_0 - \alpha \frac{\partial SSE}{\partial \beta_0}$$
$$\beta_1^{new} = \beta_1 - \alpha \frac{\partial SSE}{\partial \beta_1}$$
$$\beta_2^{new} = \beta_2 - \alpha \frac{\partial SSE}{\partial \beta_2}$$

Where α is the learning rate, and the partial derivatives (gradients) are:

$$\frac{\partial SSE}{\partial \beta_0} = -2 \sum_{i=1}^{n} (Y_i - \hat{Y}_i)$$
$$\frac{\partial SSE}{\partial \beta_1} = -2 \sum_{i=1}^{n} (Y_i - \hat{Y}_i) X_{1i}$$
$$\frac{\partial SSE}{\partial \beta_2} = -2 \sum_{i=1}^{n} (Y_i - \hat{Y}_i) X_{2i}$$

Continue iterating until the changes in $\beta 0$ values are smaller than a predefined threshold (convergence).

3. **Model Training:**

   Use the training dataset to find the optimal ($\beta 0$,$\beta 1$, $\beta 2$) values by minimizing the SSE.

4. **Model Evaluation:**

   Compute the evaluation metrics (MAE, MSE, R^2) using the test dataset.

5. **Model Validation:**

   Validate the model using cross-validation techniques to ensure generalization.

6. **Model Refinement:**

   Adjust the model parameters or features if necessary, based on the evaluation results.

c) Convert your logic and mathematical equations on 3.b) to be block of codes in any programming language you prefered! (*Notes: you are not allowed to use any machine learning package such as sklearn.KMeans, sklearn.svm, torch.nn, etc. but you are allowed to use mathematical and statistical library such as numpy, pandas, statsmodels.api.OLS, and etc.)

Implementation Code:
https://github.com/fadilrisdian/insignia-ds-assesment-2024/blob/main/3.ipynb

4. After built your model on 3.) then you get another case (validation dataset) as below:

houseId	length	lengthUnit	width	widthUnit	isCarport	price	notes
8	50	meter	20	meter	0	IDR 15 Billion	VALIDATION DATA
9	20	meter	30	meter	1	IDR 13 Billion	VALIDATION DATA
10	10	meter	20	meter	0	IDR 11 Billion	VALIDATION DATA

a) How good your model on 3.) predict these data on 4.)? You can share the accuracy or RMSE or else to be an evidence!

Mean Absolute Error (MAE): 4.424144785549848
Mean Squared Error (MSE): 28.575082149390862
Root Mean Squared Error (RMSE): 5.345566588247766
R-squared (R^2): -9.715655806021573

Implementation Code:
https://github.com/fadilrisdian/insignia-ds-assesment-2024/blob/main/3.ipynb

b) If not quite good, how that can be happened? Else, if your model can predict these data very well, how it could be and elaborate your handling methods!

The model's poor performance is mainly because there isn't enough data. With limited data, the model can't learn the patterns correctly. This can cause it to focus on random noise instead of real trends or miss important details, leading to bad predictions. More data would help the model understand and make better predictions for new cases.

c) George Edward Pelham Box (1919–2013) once said, "Essentially, all models are wrong, but some are useful", do you agree?

I agree, it means that models simplify reality and can't capture every detail accurately. Despite their flaws, they offer valuable insights, aid decision-making, and help us understand patterns and make predictions,