Nice — I’ll do three things in this reply:

1. Clarify what you meant by **“ifs”** in a data-processing / ML context.
2. Give a short, curated list of quality **online references** (problems + worked solutions) that cover each topic you listed.
3. Provide **one full worked problem** (concepts → solution → runnable Python code) that *demonstrates all requested topics*: descriptive & inferential statistics, causation vs correlation, data loading & conditional filters (“ifs”), exploration charts, modelling (train/test split, Linear Regression, Decision Tree), and performance evaluation.

**1) What are “ifs”? (short, precise)**

In data work **“ifs”** usually means conditional logic / conditional selection (a.k.a. *filtering*). Examples:

* if statements in Python used to branch logic (e.g., if x > 0: ...).
* Conditional expressions/operators to compute new columns (e.g., np.where(condition, a, b) or df['flag'] = (df['col'] > 5)).
* Row filtering in tabular data: df[df['age'] > 30], df.query('colA > 10 and colB == "X"'), df.loc[mask].  
  These let you *subset* the data, create masks, handle missing values differently, or implement feature creation rules. (See the Pandas docs and filtering tutorials below.) ([Pandas](https://pandas.pydata.org/docs/getting_started/intro_tutorials/03_subset_data.html?utm_source=chatgpt.com))

**2) Curated online references (problems + worked solutions) — organized by topic**

**Statistics (descriptive & inferential; causation vs correlation)**

* LibreTexts — Correlation versus Causation (clear conceptual notes). ([Statistics LibreTexts](https://stats.libretexts.org/Courses/Taft_College/PSYC_2200%3A_Elementary_Statistics_for_Behavioral_and_Social_Sciences_%28Oja%29/03%3A_Relationships/14%3A_Correlations/14.03%3A_Correlation_versus_Causation?utm_source=chatgpt.com))
* Intro / applied inferential-statistics tutorials (hypothesis testing + CI) — many university notes; for quick video explanation of correlation/causation see Stat/YouTube explainers. ([YouTube](https://www.youtube.com/watch?v=Tg6e2Y3IEUk&utm_source=chatgpt.com))

**Data processing (loading data, *ifs*, filters, etc.)**

* Pandas official tutorial: “How do I select a subset of a DataFrame?” — includes conditional selections, .query(), .isin(). ([Pandas](https://pandas.pydata.org/docs/getting_started/intro_tutorials/03_subset_data.html?utm_source=chatgpt.com))
* Pandas .filter reference and practical filtering patterns (chaining, boolean masks). ([Pandas](https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.filter.html?utm_source=chatgpt.com))

**Data exploration (charts / EDA)**

* Seaborn / Matplotlib tutorials — histograms, scatter plots, boxplots, pairplots, heatmaps for correlation. Datacamp / EliteDataScience guides are practical. ([DataCamp](https://www.datacamp.com/tutorial/seaborn-python-tutorial?utm_source=chatgpt.com" \o "Python Seaborn Tutorial For Beginners: Start Visualizing Data))

**Data modelling (ML models, splitting, evaluation)**

* scikit-learn: dataset loading (California housing), train\_test\_split, LinearRegression, DecisionTreeRegressor, cross-validation, scoring metrics (MSE/MAE/R²). The scikit-learn docs have canonical code & small exercises. ([scikit-learn](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html?utm_source=chatgpt.com))

**Complete worked notebooks (hands-on problems + solutions)**

* Kaggle notebooks with California housing regression (worked EDA → models → evaluation). Good for exercises to adapt / reproduce. ([Kaggle](https://www.kaggle.com/code/nileshmalode1/california-housing-dataset-simple-linear-regressio?utm_source=chatgpt.com))

**3) One full problem (statement → walk-through → clear solution + Python code)**

**Problem statement (concise)**  
Using the **California Housing** dataset (available from sklearn.datasets), build and evaluate models to predict **median house value**. Your deliverables:

1. Load the data and produce descriptive statistics and visual EDA (histograms, scatter, correlation heatmap).
2. Illustrate the **difference between correlation and causation** with a concrete example from the dataset.
3. Use *conditional filtering (“ifs”)* to create two subsets: “high income” blocks (MedInc > 5) and “low income” blocks (MedInc <= 5) and compare summary stats.
4. Build two models (Linear Regression and Decision Tree Regressor). Show train/test split and use cross-validation.
5. Evaluate with MSE, MAE and R²; discuss results and model selection.

**Why this dataset?** It’s real, included in scikit-learn, small enough to run quickly and has continuous target suitable for regression. ([scikit-learn](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html?utm_source=chatgpt.com))

**Full solution — explanation + runnable Python code (copy & paste)**

# Full reproducible example: California housing regression

# Run in Python 3.8+ with scikit-learn, pandas, numpy, matplotlib, seaborn installed.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split, cross\_val\_score, KFold

from sklearn.linear\_model import LinearRegression

from sklearn.tree import DecisionTreeRegressor

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

from sklearn.preprocessing import StandardScaler

# 1) Load dataset

cal = fetch\_california\_housing(as\_frame=True)

df = pd.concat([cal.frame, cal.target.rename('MedHouseVal')], axis=1) # cal.frame already has data+target in newer sklearns

# Quick peek

print(df.shape)

print(df.head())

print("\nDescriptive statistics (numeric):\n", df.describe().T)

# 2) EDA - histograms, scatter, correlation heatmap

plt.figure(figsize=(10, 6))

df['MedHouseVal'].hist(bins=40)

plt.title('Distribution of Median House Value')

plt.xlabel('Median House Value (units of 100k)')

plt.show()

# Pairwise scatter: choose a few features to save time

features = ['MedInc', 'AveRooms', 'AveOccup', 'Latitude', 'Longitude']

sns.pairplot(df[features + ['MedHouseVal']], height=2.2)

plt.suptitle('Pairwise plots (selected features)', y=1.02)

plt.show()

# Correlation heatmap

plt.figure(figsize=(10, 6))

sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap='coolwarm', vmin=-1, vmax=1)

plt.title('Correlation matrix (all features + target)')

plt.show()

# 3) Correlation vs Causation (conceptual demonstration)

# Compute correlation between MedInc and MedHouseVal

corr\_medinc = df['MedInc'].corr(df['MedHouseVal'])

print(f"Pearson correlation (MedInc, MedHouseVal) = {corr\_medinc:.3f}")

# NOTE (concept): A positive correlation means higher median income areas tend to have higher house values,

# but correlation alone does NOT prove income causes house price (many confounders: proximity to coast, city-level demand, supply, zoning).

# See inferential/experimental design for causal claims.

# 4) Conditional filtering - the "ifs"

high\_income = df[df['MedInc'] > 5.0] # "if MedInc > 5 then ..."

low\_income = df[df['MedInc'] <= 5.0]

print("High income subset:", high\_income.shape, "Low income subset:", low\_income.shape)

# Compare means

print("Mean MedHouseVal (high income):", high\_income['MedHouseVal'].mean())

print("Mean MedHouseVal (low income):", low\_income['MedHouseVal'].mean())

# 5) Prepare data for modelling

X = df[cal.feature\_names] # feature matrix

y = df['MedHouseVal']

# Train/test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Optional scaling for linear models

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# 6) Linear Regression

lr = LinearRegression()

lr.fit(X\_train\_scaled, y\_train)

y\_pred\_lr = lr.predict(X\_test\_scaled)

mse\_lr = mean\_squared\_error(y\_test, y\_pred\_lr)

mae\_lr = mean\_absolute\_error(y\_test, y\_pred\_lr)

r2\_lr = r2\_score(y\_test, y\_pred\_lr)

print("Linear Regression test results: MSE={:.3f}, MAE={:.3f}, R2={:.3f}".format(mse\_lr, mae\_lr, r2\_lr))

# 7) Decision Tree Regressor (no scaling needed)

dt = DecisionTreeRegressor(max\_depth=6, random\_state=42)

dt.fit(X\_train, y\_train)

y\_pred\_dt = dt.predict(X\_test)

mse\_dt = mean\_squared\_error(y\_test, y\_pred\_dt)

mae\_dt = mean\_absolute\_error(y\_test, y\_pred\_dt)

r2\_dt = r2\_score(y\_test, y\_pred\_dt)

print("Decision Tree (depth=6) test results: MSE={:.3f}, MAE={:.3f}, R2={:.3f}".format(mse\_dt, mae\_dt, r2\_dt))

# 8) Cross-validation to compare generalization (5-fold)

kf = KFold(n\_splits=5, shuffle=True, random\_state=1)

cv\_lr\_scores = cross\_val\_score(lr, scaler.transform(X), y, scoring='r2', cv=kf)

cv\_dt\_scores = cross\_val\_score(dt, X, y, scoring='r2', cv=kf)

print("CV R2 (LR): mean={:.3f} std={:.3f}".format(cv\_lr\_scores.mean(), cv\_lr\_scores.std()))

print("CV R2 (DT): mean={:.3f} std={:.3f}".format(cv\_dt\_scores.mean(), cv\_dt\_scores.std()))

# 9) Feature importance (Decision Tree)

importances = pd.Series(dt.feature\_importances\_, index=cal.feature\_names).sort\_values(ascending=False)

print("\nDecision Tree feature importances:\n", importances)

# 10) Final comments: interpret metrics and suggest next steps (in comments)

**Explanation & mapping of code → concepts**

* **Descriptive statistics**: df.describe() shows mean, std, quartiles (central tendency, spread, outliers).
* **Exploratory charts**: histogram (target distribution), pairplots (relationships), correlation heatmap (linear associations). Use these to spot skew, collinearity, and candidate predictors. (Seaborn/Matplotlib examples and tutorials linked above.) ([DataCamp](https://www.datacamp.com/tutorial/seaborn-python-tutorial?utm_source=chatgpt.com" \o "Python Seaborn Tutorial For Beginners: Start Visualizing Data))
* **Correlation vs causation**: the example MedInc vs MedHouseVal shows correlation (computed) but we explain confounders (location, ocean proximity). To *claim causation* you'd need an experiment or quasi-experimental design / causal inference tools (instrumental variables, differences-in-differences, randomized control). See LibreTexts for conceptual warnings. ([Statistics LibreTexts](https://stats.libretexts.org/Courses/Taft_College/PSYC_2200%3A_Elementary_Statistics_for_Behavioral_and_Social_Sciences_%28Oja%29/03%3A_Relationships/14%3A_Correlations/14.03%3A_Correlation_versus_Causation?utm_source=chatgpt.com))
* **“ifs” / conditional filters**: high\_income = df[df['MedInc'] > 5.0] demonstrates standard Pandas filtering (boolean masks). Use .query() or np.where() for other patterns — see Pandas docs. ([Pandas](https://pandas.pydata.org/docs/getting_started/intro_tutorials/03_subset_data.html?utm_source=chatgpt.com))
* **Data splitting & modeling**: train\_test\_split for holdout test evaluation; LinearRegression & DecisionTreeRegressor show a simple linear baseline and a non-linear tree. Cross-validation (cross\_val\_score) checks stability/generalization. Use scoring metrics MSE, MAE, R². Scikit-learn docs are canonical examples. ([scikit-learn](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html?utm_source=chatgpt.com))

**Short discussion of results (how to interpret)**

* If Linear Regression has similar R² to Decision Tree but lower variance across folds → prefer LR for interpretability.
* If Decision Tree shows higher R² on test but much higher variance in CV → likely overfitting; tune max\_depth or prune.
* Feature importances from the tree give a sense of which variables the model relied on, but remember importance ≠ causation.
* If you want causal claims (e.g., “increasing median income by X causes house price to rise by Y”), use causal inference methods or controlled experiments — not simple correlations or standard supervised models. ([Statistics LibreTexts](https://stats.libretexts.org/Courses/Taft_College/PSYC_2200%3A_Elementary_Statistics_for_Behavioral_and_Social_Sciences_%28Oja%29/03%3A_Relationships/14%3A_Correlations/14.03%3A_Correlation_versus_Causation?utm_source=chatgpt.com))

**Extra resources & exercises you can assign to students**

* Reproduce the notebook and add **regularization** (Ridge/Lasso) — compare coefficients and test R². (See scikit-learn linear model docs.) ([DataCamp](https://www.datacamp.com/tutorial/sklearn-linear-regression?utm_source=chatgpt.com" \o "Sklearn Linear Regression: A Complete Guide with Examples))
* Convert to a classification task (e.g., predict whether MedHouseVal > median) — try logistic regression, random forest, ROC/AUC metrics.
* Try *feature engineering*: compute distance-to-coast proxy (Longitude/Latitude transform) and show how it changes model performance in cross-validation. (Kaggle notebooks demonstrate this.) ([Kaggle](https://www.kaggle.com/code/nileshmalode1/california-housing-dataset-simple-linear-regressio?utm_source=chatgpt.com))

**Quick checklist for a student assignment (so instructors/students know what to hand in)**

* Notebook with: data loading + description, EDA charts, correlation matrix.
* Short written paragraph explaining correlation vs causation with a dataset-specific example.
* Two subsets created with conditional filters and short comparison table.
* Models trained (LR & DT), test metrics and 5-fold CV results, feature importances.
* Short conclusion: which model you'd pick and why, what next experiments you'd run.

If you want, I can:

* Convert the code into a **clean, commented Jupyter notebook** ready for students (I can paste the notebook .ipynb cells here), or
* Provide a **short assignment sheet** + solution rubric based on the above, or
* Run the code here and return the plots and a downloadable notebook (tell me which you prefer).

Which of those would you like next?

Cool — here are two things: (A) a nicely-structured **short assignment sheet** + **rubric** you can use to self-assign / practice; (B) a list of LinkedIn Learning & DataCamp courses/projects corresponding to the individual skills you want to build. You can pick and follow those to improve in each area.

**A) Short Assignment Sheet + Solution Rubric**

You (as student) can print/use this to guide your own work. Try to complete each part, then assess according to the rubric.

**Assignment: Predicting Housing Prices & Exploring Data**

**Objective:** Apply statistical, data processing, exploration, and modelling techniques to a dataset to predict a numerical target. Practice descriptive & inferential statistics, correlation vs causation, conditional filtering, and evaluating models (linear regression, decision tree).

**Tasks**

1. **Data Loading & Descriptive Statistics**
   * Pick a dataset with numerical features + continuous target (e.g. California Housing, Ames Housing, or similar).
   * Load into Python (e.g. Pandas).
   * Show the first few rows (head), shape, data types.
   * Provide descriptive statistics: mean, median, std, quartiles etc. for each numeric column.
2. **Data Cleaning & Conditional Filtering**
   * Check for missing values, handle them (drop, fill, etc).
   * Use conditional filtering (“ifs”) to split data into at least two subsets based on a feature. For example: “high vs low income”, “high vs low population density”, etc.
   * Compare summary statistics between the subsets for at least one outcome variable (target).
3. **Exploration / Visualization**
   * Produce histograms of the target (and at least two features).
   * Produce scatter plots of target vs important features.
   * Create a correlation heatmap of numeric features + target.
   * Identify at least two interesting relationships (strong correlation, outliers, skew, etc).
4. **Correlation vs Causation Discussion**
   * Pick two variables with a good correlation; discuss: does this suggest causation? What confounders might exist? What would you need to establish causation?
5. **Modelling**
   * Split the data into train & test sets (e.g. 70-30 or 80-20) with a fixed random seed.
   * Fit a **Linear Regression** model.
   * Fit a **Decision Tree Regressor** model (with at least one depth constraint or hyperparameter).
   * Optionally try cross-validation (e.g. k-fold) to assess generalization.
6. **Performance Evaluation**
   * Compute and report at least: Mean Squared Error (MSE), Mean Absolute Error (MAE), and R² on test set.
   * (Optional) Compare with cross-validation results.
   * For the Decision Tree: show feature importances.
7. **Interpretation & Conclusion**
   * Based on results, which model performs better (in your case)? Why?
   * What are possible ways to improve? (Feature engineering, more data, transformations, regularization etc.)
   * Note limitations: e.g. causation issues, overfitting, data bias etc.

**Rubric / Self-Scoring**

| **Component** | **Full Marks** | **Points You Should Aim For / What to Check** |
| --- | --- | --- |
| Data loading & descriptive statistics | 10 | Data correctly loaded, no major errors; descriptive stats present, clean, interpretable. |
| Cleaning & conditional filtering | 10 | Missing values addressed; filtering correctly implemented; comparisons logical. |
| Visual exploration & charts | 15 | Plots are well labeled, informative; correct choice of features; heatmap correct. |
| Correlation vs causation discussion | 10 | Identifies correlation; mentions plausible confounders; shows understanding of limitations. |
| Model building (LR & Decision Tree) | 15 | Splits done properly; models fit as specified; hyperparameters considered. |
| Performance evaluation | 15 | Metrics correctly calculated; comparisons made; cross‐validation (if done) used properly. |
| Interpretation & conclusion | 10 | Clear conclusions about model choices; suggestions for improvement; honest about limitations. |
| Code style & reproducibility | 5 | Code is readable, organized; reproducible (random seeds, comments); plots & outputs visible. |
| Overall clarity & documentation | 10 | Written explanations are clear; figures & results embedded; good narrative flow. |

**Total: 100 pts**

**Suggested timeline**

* Day 1: Data loading, descriptive stats, cleaning, filtering
* Day 2: EDA (plots, correlation heatmap), correlation/causation discussion
* Day 3: Model fitting, evaluation, interpretation & report

**B) Courses / Projects for Individual Skills**

Here are resources (LinkedIn Learning, DataCamp etc.) mapped to each skill you want. You can use them to target weak spots or to get guided practice.

| **Skill / Concept** | **Good Course / Project Recommendation** |
| --- | --- |
| **Descriptive & inferential statistics** | *LinkedIn Learning:* “Applied Machine Learning: Value Estimation — Data Exploration and Cleaning.” ([LinkedIn](https://www.linkedin.com/learning/applied-machine-learning-value-estimation/data-exploration-and-cleaning?utm_source=chatgpt.com)) *DataCamp:* There are probability & statistics tracks; also “Supervised Learning in R / Python” which include inferential stuff. *Specific article/tutorial:* DataCamp tutorial on model evaluation metrics includes inferential confidence intervals etc. (look up “Evaluating a model graphically” in DataCamp). ([DataCamp Campus](https://campus.datacamp.com/courses/supervised-learning-in-r-regression/tree-based-methods?ex=2&utm_source=chatgpt.com" \o "Predicting with a decision tree | R)) |
| **Correlation / causation** | The general statistics courses (on LinkedIn or free online) that include correlation, regression; plus applied ML courses where they caution about causation. LinkedIn’s AI foundations and ML life-cycle courses usually cover those. ([LinkedIn](https://www.linkedin.com/learning/artificial-intelligence-foundations-machine-learning-22345868/introduction-to-ai-foundations-machine-learning-course?utm_source=chatgpt.com)) |
| **Data processing / conditional filtering / ifs** | *LinkedIn Learning:* “Applied Machine Learning: Value Estimation — Data Exploration and Cleaning” includes loading & cleaning & filtering. ([LinkedIn](https://www.linkedin.com/learning/applied-machine-learning-value-estimation/data-exploration-and-cleaning?utm_source=chatgpt.com)) *LinkedIn Learning:* “Complete Guide to Google BigQuery... DataFrames for Data Exploration” ([LinkedIn](https://www.linkedin.com/learning/complete-guide-to-google-bigquery-for-data-and-ml-engineers/dataframes-for-data-exploration?utm_source=chatgpt.com)) |
| **Exploratory Data Analysis (charts, EDA)** | *LinkedIn Learning:* “Exploratory Data Analysis (EDA) — AI fundamentals for Data Professionals” ([LinkedIn](https://www.linkedin.com/learning/ai-fundamentals-for-data-professionals/exploratory-data-analysis-eda?utm_source=chatgpt.com)) *DataCamp:* Many courses include EDA modules (e.g. “Supervised Learning”, “Pandas / Data Manipulation” courses) The “Challenge: Load, explore, and clean data” on LinkedIn is directly relevant. ([LinkedIn](https://www.linkedin.com/learning/applied-machine-learning-value-estimation/challenge-load-explore-and-clean-data?utm_source=chatgpt.com)) |
| **Data modelling: linear regression, decision trees** | *DataCamp:* “Machine Learning with Tree-Based Models in Python” — good for decision tree & tree-based ensembles in regression & classification. ([DataCamp](https://www.datacamp.com/courses/machine-learning-with-tree-based-models-in-python?utm_source=chatgpt.com" \o "Machine Learning with Tree-Based Models in Python - DataCamp)) *DataCamp:* “Decision tree for regression |
| **Data splitting, performance evaluation** | *DataCamp:* The tree-based models course includes splits, hyperparameter tuning and evaluation. ([DataCamp](https://www.datacamp.com/courses/machine-learning-with-tree-based-models-in-python?utm_source=chatgpt.com" \o "Machine Learning with Tree-Based Models in Python - DataCamp)) *DataCamp:* “Supervised Learning in R” has evaluation metrics as part. ([DataCamp Campus](https://campus.datacamp.com/courses/supervised-learning-in-r-regression/tree-based-methods?ex=2&utm_source=chatgpt.com" \o "Predicting with a decision tree | R)) |

If you want, I can put together a **personal study plan** for you (e.g. 4-6 weeks) with specific courses and assignments so you improve steadily, targeting your weakest spots. Do you want me to build that for you?

Perfect — let’s make you a **4-day focused plan (3 hours/day, ~12 hours total)** that touches all the requested concepts:

* Statistics (descriptive, inferential, correlation vs causation)
* Data processing (loading, ifs/filters)
* Data exploration (charts)
* Data modelling (linear regression, decision trees, splitting, evaluation)

We’ll keep it **Python-only** and use **DataCamp + LinkedIn Learning** + practice assignment (the housing dataset we built above).

**📅 4-Day Study Plan (3 hours/day)**

**Day 1 – Foundations: Statistics + Data Processing (3h)**

**1. Warm-up (15m)**

* Skim Pandas official tutorial on conditional selections (.loc, .query, boolean masks).  
  👉 [Pandas conditional selection](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#boolean-indexing)

**2. Descriptive & Inferential Statistics (1h 15m)**

* *DataCamp*: [Introduction to Statistics in Python](https://www.datacamp.com/courses/introduction-to-statistics-in-python) (focus on Descriptive + Inferential sections).
* *LinkedIn Learning*: [Statistics Foundations: 1](https://www.linkedin.com/learning/statistics-foundations-1) (first half: descriptive statistics, probability basics).

**3. Practice on California Housing dataset (1h 30m)**

* Load dataset (sklearn.datasets.fetch\_california\_housing).
* Compute describe(), mean, median, quartiles, variance, correlation matrix.
* Write short notes: which features are skewed, which correlate strongly with target.
* Apply **conditional filtering (“ifs”)**: split dataset into high vs low income groups, compare target means.

**Day 2 – Data Exploration (Charts & Correlation vs Causation) (3h)**

**1. Visualization crash (1h)**

* *DataCamp*: [Data Visualization with Seaborn](https://www.datacamp.com/courses/data-visualization-with-seaborn) (focus on histograms, scatter, heatmaps).
* *LinkedIn Learning*: [Data Science Foundations: Data Mining](https://www.linkedin.com/learning/data-science-foundations-data-mining) (EDA part).

**2. Practice plots (1h 30m)**

* Plot target distribution (hist).
* Scatterplots (MedInc vs MedHouseVal, AveRooms vs MedHouseVal).
* Correlation heatmap (sns.heatmap).
* Identify two correlations. Discuss: does correlation imply causation? What confounders could exist?

**3. Quick wrap-up reflection (30m)**

* Write short answers:
  + What is correlation?
  + Why does correlation ≠ causation?
  + One example from dataset.

**Day 3 – Modelling I: Linear Regression (3h)**

**1. Learn basics (1h)**

* *DataCamp*: [Supervised Learning with scikit-learn](https://www.datacamp.com/courses/supervised-learning-with-scikit-learn) (focus on Linear Regression, train\_test\_split, metrics).
* *LinkedIn Learning*: [Machine Learning with Scikit-Learn](https://www.linkedin.com/learning/machine-learning-with-scikit-learn) (Linear Regression section).

**2. Practice on dataset (1h 30m)**

* Train/test split (80/20).
* Fit Linear Regression (LinearRegression).
* Evaluate with **MSE, MAE, R²**.
* Try standardizing features (StandardScaler) — compare before/after.

**3. Reflection (30m)**

* When does Linear Regression fail? (nonlinear relationships, multicollinearity).
* Check residual plot (scatter y\_test vs residuals).

**Day 4 – Modelling II: Decision Trees + Evaluation (3h)**

**1. Learn tree models (1h)**

* *DataCamp*: [Machine Learning with Tree-Based Models in Python](https://www.datacamp.com/courses/machine-learning-with-tree-based-models-in-python) (Decision Trees).
* *LinkedIn Learning*: [Applied Machine Learning: Algorithms](https://www.linkedin.com/learning/applied-machine-learning-algorithms) (Decision Trees & Random Forests section).

**2. Practice on dataset (1h 30m)**

* Train DecisionTreeRegressor (vary max\_depth=3,6,10).
* Compare metrics (MSE, MAE, R²) to Linear Regression.
* Show feature\_importances\_ bar plot.
* Run 5-fold cross-validation for both models, compare mean R².

**3. Wrap-up report (30m)**

* 1-2 pages summarizing:
  + Stats/EDA insights.
  + Correlation vs causation example.
  + Linear Regression vs Decision Tree results.
  + Which model you’d pick and why.
  + Next steps (feature engineering, regularization, ensembles).

**✅ Outcome after 4 days**

By the end, you’ll have:

* Comfort with **statistics** (mean, variance, correlation, CI, causation caveats).
* Skills in **data processing** (filters, ifs, subsetting).
* Hands-on **EDA with charts**.
* **Two ML models** (Linear Regression + Decision Tree) applied, evaluated, and compared.
* A short written report — something you could showcase as a small **portfolio project**.

Do you want me to **bundle this into a downloadable PDF “study guide”** (assignment sheet + plan + resource links) so you can keep it offline while studying?