Question ID:	M01-Q01
Topic:	M1a – What is Data Science?
Difficulty:	Easy
Blooms:	Understand

What is Data Science mainly about?

1. Using data to identify patterns and make informed decisions
2. Creating as much data as possible
3. Replacing scientists with AI
4. Making charts without analyzing meaning
5. Searching the internet for interesting pictures

Question ID:	M01-Q02
Topic:	M1a – Flood of Data
Difficulty:	Easy
Blooms:	Understand

Why has Data Science become more important recently?

1. Because we now collect and store enormous amounts of data
2. Because statistics were just invented recently
3. Because computers are disappearing
4. Because hand-drawn measurements are more popular
5. Because people have shorter attention spans

Question ID:	M01-Q03
Topic:	M1a – Data Visualization
Difficulty:	Medium
Blooms:	Apply

What role does visualization play in Data Science?

1. Helps humans understand patterns that exist in large datasets
2. It eliminates the need for data cleaning
3. It guarantees conclusions are correct
4. It replaces statistics entirely
5. It is only useful for small datasets

Question ID:	M01-Q04
Topic:	M1a – Statistics: Nonsense Protection
Difficulty:	Medium
Blooms:	Analyze

What does statistics help prevent in analysis?

1. False conclusions caused by random patterns
2. Collecting too much data
3. Good visualizations
4. The need for computers
5. Faster machine learning training

Question ID:	M01-Q05
Topic:	M1a – Coping with Data
Difficulty:	Medium
Blooms:	Remember

Which is NOT one of the core parts of data science discussed?

1. Handling data scale
2. Visualization
3. Algorithms
4. Astrology
5. Computation

Question ID:	M01-Q06
Topic:	M1b – Tools
Difficulty:	Easy
Blooms:	Understand

Why is Python widely used in Data Science?

1. It has libraries for math, visualization, and machine learning
2. It is the only programming language for data analysis
3. It was designed to replace image editors
4. It is extremely fast for everything
5. It can only be used in Jupyter notebooks

Question ID:	M01-Q07
Topic:	M1b – Google Colab
Difficulty:	Easy
Blooms:	Remember

Which tool provides a Python environment in the browser without local installation?

1. Anaconda
2. Google Colab
3. Visual Basic
4. GitHub Desktop
5. Local terminal only

Question ID:	M01-Q08
Topic:	M1c – Arrays
Difficulty:	Medium
Blooms:	Understand

Which statement about arrays is TRUE?

1. Easy to insert in the middle
2. Always sorted
3. Fast access by index
4. Automatically distributed across different computers
5. Removing an element is always O(1)

Question ID:	M01-Q09
Topic:	M1c – Stacks
Difficulty:	Medium
Blooms:	Analyze

Which is a disadvantage of stacks?

1. Hard to access arbitrary elements
2. Easy to remove from the top
3. Very slow at resizing
4. They don't allow push operations
5. They cannot be implemented in Python

Question ID:	M01-Q10
Topic:	M1c – Queue vs Stack
Difficulty:	Easy
Blooms:	Understand

What is the key difference between a Queue and a Stack?

1. Queue removes the earliest inserted item first; Stack removes the most recently inserted
2. Queue is stored in trees
3. Stack elements must be unique
4. Queue requires a GPU
5. Stack never allows push operations

Question ID:	M01-Q11
Topic:	M1c – Linked Lists
Difficulty:	Medium
Blooms:	Apply

Linked Lists are particularly good when...

1. We need constant-time random indexing
2. Frequent insertions or deletions at ends or arbitrary positions
3. Data never changes
4. We want built-in sorting
5. We only store one element

Question ID:	M01-Q12
Topic:	M1c – Hash Tables
Difficulty:	Medium
Blooms:	Understand

A key-value store such as a Python dictionary...

1. Allows fast lookup by key
2. Preserves items in sorted order
3. Can only store numbers
4. Requires sequential search for access
5. Only allows string keys

Question ID:	M01-Q13
Topic:	M1c – Trees
Difficulty:	Medium
Blooms:	Understand

What is a Tree most useful for compared to linear structures?

1. Representing hierarchical relationships
2. Guaranteeing O(1) for all operations
3. Storing only numbers
4. Using no pointers internally
5. Always being binary

Question ID:	M01-Q14
Topic:	M1c – Broadcasting
Difficulty:	Medium
Blooms:	Apply

In NumPy, what is "broadcasting"?

1. Automatically expanding arrays to compatible shapes during arithmetic
2. Sending arrays to a TV channel
3. Only used when adding identical shapes
4. A memory compression method
5. Copying data into SQL databases

Question ID:	M01-Q15
Topic:	M1c – Shape compatibility
Difficulty:	Medium
Blooms:	Analyze

Why can't you always add two NumPy arrays of different shapes?

1. The shapes must be compatible so elements correspond
2. NumPy refuses to add integers
3. Arrays must always be 2×2
4. The larger array deletes itself
5. Only works for prime-number shapes

Question ID:	M01-Q16
Topic:	M1c — Array memory organization
Difficulty:	Medium
Blooms:	Understand

One major reason arrays allow fast random access is:

1. The array keeps a hash table of all indexes
2. Each element stores the address of the next
3. Elements are stored contiguously so index lookup is constant time
4. The OS automatically accelerates access for arrays
5. Arrays are always stored in CPU cache

Question ID:	M01-Q17
Topic:	M1c — Array disadvantages
Difficulty:	Medium
Blooms:	Analyze

Why is deleting an element from the middle of a Python list inefficient?

1. All later elements must shift to fill the gap
2. Python must rebuild the entire interpreter
3. Removal causes permanent fragmentation
4. Python sorts the list after every deletion
5. The list converts to a linked list internally

Question ID:	M01-Q18
Topic:	M1c — FILO semantics
Difficulty:	Medium
Blooms:	Apply

Which scenario best leverages a stack data structure?

1. Tracking nested function calls
2. Processing real-time streaming logs
3. Storing only unique items
4. Implementing alphabetical search
5. Indexing stock-market time series

Question ID:	M01-Q19
Topic:	M1c — FIFO
Difficulty:	Easy
Blooms:	Understand

A queue supports which behavior?

1. Requires O(1) random access
2. Only works if all items are numbers
3. Removes the earliest inserted item first
4. Can only grow, never shrink
5. Automatically sorts items by priority

Question ID:	M01-Q20
Topic:	M1c — Linked lists
Difficulty:	Medium
Blooms:	Analyze

Linked lists scale well for which workload?

1. Fast binary search
2. Frequent insertions at arbitrary locations
3. Automatic rebalancing of data
4. Multi-dimensional indexing
5. Constant-time random access

Question ID:	M01-Q21
Topic:	M1c — Hash functions
Difficulty:	Medium
Blooms:	Understand

Why are dictionaries (hash tables) efficient for lookup by key?

1. Keys are stored in sorted order
2. A hash function maps keys to near-constant-time index access
3. Values are duplicated in multiple locations
4. Every operation scans the entire table
5. The data is stored on the GPU

Question ID:	M01-Q22
Topic:	M1c — Trees
Difficulty:	Medium
Blooms:	Understand

A tree differs from a stack or queue primarily because it:

1. Represents hierarchical relationships
2. Stores only numbers
3. Guarantees all operations are O(1)
4. Uses no pointers internally
5. Is always binary

Question ID:	M01-Q23
Topic:	M1c — Composite
Difficulty:	Hard
Blooms:	Create

Which situation is best for a composite data structure (e.g., tree of arrays)?

1. You only need sequential iteration
2. You must store data that never changes
3. You need to mix hierarchical lookup with fast local indexing
4. You want to minimize implementation complexity
5. You require strict alphabetical ordering

Question ID:	M01-Q24
Topic:	M1c — Hash table disadvantages
Difficulty:	Medium
Blooms:	Analyze

Which is a key drawback of hash tables?

1. They require keys to be sorted
2. They cannot delete items
3. They do not preserve meaningful order
4. Searching is always slower than a linked list
5. They can only store fixed-size elements

Question ID:	M01-Q25
Topic:	M1c — Broadcasting rules
Difficulty:	Medium
Blooms:	Apply

In NumPy, why does adding a 1×3 vector to a 3×3 matrix work?

1. The vector is broadcast across rows to match shape
2. The matrix is flattened automatically
3. NumPy guesses the user's intention
4. The vector overwrites the diagonal
5. Because 3 is a special broadcast-safe number

Question ID:	M01-Q26
Topic:	M1c — Shape mismatch
Difficulty:	Medium
Blooms:	Analyze

Which NumPy operation fails without broadcasting compatibility?

1. Adding arrays with mismatched dimensions that cannot expand
2. Adding a scalar to any array
3. Element-wise multiplication of equal shapes
4. Index slicing
5. Dot product of same-length vectors

Question ID:	M01-Q27
Topic:	M1b — Tools justification
Difficulty:	Easy
Blooms:	Understand

Why is Python frequently chosen for data science workflows?

1. The language forces all variables to be floats
2. Python code always runs faster than C
3. Strong ecosystem of numerical and ML libraries
4. It only works in notebook environments
5. It replaces the need for distributed computing

Question ID:	M01-Q28
Topic:	M1a — Human cognition limits
Difficulty:	Medium
Blooms:	Evaluate

Why is data visualization essential in data science?

1. Visualizations remove all noise from data
2. Humans understand visual patterns better than raw numbers
3. Plots guarantee a correct conclusion
4. Visualizations replace statistical inference
5. Visualization is only useful on labeled data

Question ID:	M01-Q29
Topic:	M1a — Big data scale challenges
Difficulty:	Medium
Blooms:	Apply

Handling big data requires more than spreadsheets because:

1. Distributed tools allow scalable computation
2. Spreadsheets always corrupt files over 10 MB
3. Databases compute without hardware
4. Large datasets require no cleaning
5. CPUs cannot process tabular data

Question ID:	M01-Q30
Topic:	M1a — Cleaning/processing effort
Difficulty:	Medium
Blooms:	Analyze

Why will data engineering remain a major effort in DS pipelines?

1. Real-world data is messy and must be cleaned before modeling
2. Machine learning automatically repairs all errors
3. Storage formats determine the model accuracy
4. Sensor data arrives perfectly structured
5. Data rarely changes once collected

Question ID:	M03-Q1
Topic:	Exploratory Data Analysis
Difficulty:	Easy
Blooms:	Understand

Which part of the data science workflow primarily focuses on understanding the structure, patterns, and anomalies present in the data?

1. Data Collection
2. Exploratory Data Analysis (EDA)
3. Confirmatory Data Analysis
4. Feature Deployment
5. GPU Optimization

Question ID:	M03-Q2
Topic:	Histogram
Difficulty:	Easy
Blooms:	Remember

Which visualization is most appropriate for examining the distribution of a single quantitative variable?

1. Scatter plot
2. Histogram
3. Network graph
4. Choropleth map
5. Box-and-whisker + KDE overlay

Question ID:	M03-Q3
Topic:	Boxplot
Difficulty:	Easy
Blooms:	Understand

Box-and-whisker plots are especially useful for:

1. Showing data on a map
2. Comparing medians and detecting outliers
3. Displaying 3D surfaces
4. Tracking time series index returns
5. Showing relationships between nominal variables

Question ID:	M03-Q4
Topic:	Nominal vs Nominal
Difficulty:	Medium
Blooms:	Apply

A bar chart comparing 'Plant Type' and 'Fruit Variety' represents what type of data?

1. Ordinal vs Quantitative
2. Nominal vs Nominal
3. Ordinal vs Ordinal
4. Quantitative vs Quantitative
5. Geospatial vs Quantitative

Question ID:	M03-Q5
Topic:	Scatter Plot
Difficulty:	Easy
Blooms:	Remember

Scatter plots are primarily used to visualize:

1. Relationships between two quantitative variables
2. Distribution of a single variable only
3. Hierarchical relationships
4. Statistical inference and p-values
5. Survey proportions

Question ID:	M03-Q6
Topic:	Time Series
Difficulty:	Medium
Blooms:	Apply

Which visualization best compares multiple time-dependent quantities simultaneously?

1. Stacked area chart
2. Single-variable histogram
3. Parallel coordinates
4. Radar chart
5. Candlestick chart

Question ID:	M03-Q7
Topic:	Candlestick Chart
Difficulty:	Easy
Blooms:	Remember

Which graph type is explicitly linked to financial market data in the lecture?

1. Horizon graph
2. Candlestick chart
3. Contour map
4. Boxplot
5. Tree map

Question ID:	M03-Q8
Topic:	Matplotlib API
Difficulty:	Medium
Blooms:	Understand

In matplotlib, why is using ‘fig, ax = plt.subplots()‘ preferred over calling ‘plt.plot()‘ directly?

1. It uses more memory so it’s faster
2. It gives explicit references to figure and axes objects for better control
3. It enables automatic machine learning integration
4. It prevents adding labels and legends
5. It is required by NumPy

Question ID:	M03-Q9
Topic:	KDE
Difficulty:	Medium
Blooms:	Analyze

A KDE (Kernel Density Estimate) is used to:

1. Visualize category labels
2. Smooth the distribution of sampled data
3. Display geospatial movement
4. Simulate stock trading
5. Normalize missing values

Question ID:	M03-Q10
Topic:	Outliers
Difficulty:	Medium
Blooms:	Evaluate

The lecture suggests that before trusting an outlier, you should:

1. Remove all outliers automatically
2. Trace back to the original data and verify context
3. Replace with the mean
4. Convert to categorical encoding
5. Report it as a major scientific discovery

Question ID:	M03-Q11
Topic:	High-dimensional visualization
Difficulty:	Medium
Blooms:	Analyze

Parallel coordinate plots are best suited for:

1. Two quantitative variables
2. Nominal-only comparisons
3. High-dimensional numeric data
4. Exact probability estimation
5. Animated game graphics

Question ID:	M03-Q12
Topic:	Q-Q Plot
Difficulty:	Hard
Blooms:	Analyze

Which visualization is used to examine whether data follows a theoretical distribution?

1. Q-Q plot
2. Box plot
3. Stacked bar chart
4. Flow map
5. 3D volume rendering

Question ID:	M03-Q13
Topic:	Composition over Time
Difficulty:	Easy
Blooms:	Remember

Which visualization technique was illustrated using time-series of Panda Hats and Underpants?

1. Index chart
2. Stacked time-series visualization
3. Heat map
4. Network diagram
5. Histogram with multiple bins

Question ID:	M03-Q14
Topic:	Quantitative Relationship
Difficulty:	Medium
Blooms:	Remember

What is added to the mother's height in calculating Galton's 'midparent' height?

1. 0.98 × mother height
2. 1.08 × mother height
3. 1.50 × mother height
4. Subtract father height
5. It uses only father height

Question ID:	M03-Q15
Topic:	3-variable visualization
Difficulty:	Medium
Blooms:	Apply

Which chart type allows three variables to be shown simultaneously using two axes and color/size encoding?

1. Bar chart with sublevels
2. Scatter plot with a third encoding
3. Pie chart slices only
4. Table layout
5. Q-Q plot

Question ID:	M03-Q16
Topic:	Summary Statistics
Difficulty:	Easy
Blooms:	Remember

Which of the following appears in the slides' list of summary statistics for EDA?

1. Skewness
2. Standard deviation
3. Fourier coefficients
4. Mode only
5. Z-score thresholds

Question ID:	M03-Q17
Topic:	Missing Values
Difficulty:	Easy
Blooms:	Remember

Which package was highlighted for visualizing patterns of missing data?

1. missingno
2. seaborn.gridplot
3. plotly.missmap
4. statsmodels.na_viz
5. opencv.impute

Question ID:	M03-Q18
Topic:	Pairwise Relationships
Difficulty:	Medium
Blooms:	Understand

A matrix of scatter plots helps you:

1. Visualize pairwise relationships among many quantitative variables
2. Render 3D surfaces of a function
3. Encode hierarchical trees
4. Compute p-values for regression
5. Normalize geospatial coordinates

Question ID:	M03-Q19
Topic:	Correlation Heat Map
Difficulty:	Easy
Blooms:	Understand

What does a correlation heat map display?

1. The sign and magnitude of linear relationships between variables
2. Raw counts of categories
3. Geographic elevation
4. Financial OHLC patterns
5. Kernel bandwidths

Question ID:	M03-Q20
Topic:	Co-occurrence
Difficulty:	Medium
Blooms:	Apply

Pair-wise co-occurrence (joint probabilities) in EDA are most appropriate for:

1. Relationships between categorical variables
2. Optimizing hyperparameters
3. 3D surface estimation
4. Fourier spectral analysis
5. GPU memory profiling

Question ID:	M03-Q21
Topic:	Automated EDA Tools
Difficulty:	Easy
Blooms:	Remember

Which tool is NOT listed among the automated EDA tools in the slides?

1. Pandas Profiling
2. Sweetviz
3. Autoviz
4. D-Tale
5. PowerBI AutoInspect

Question ID:	M03-Q22
Topic:	Index Charts
Difficulty:	Medium
Blooms:	Understand

In an index chart for time series, what does 'indexing' typically do?

1. Normalizes each series to a common baseline (e.g., 100) to compare relative change
2. Sorts categories alphabetically
3. Automatically removes outliers
4. Interpolates missing geolocations
5. Computes PCA scores

Question ID:	M03-Q23
Topic:	Small Multiples
Difficulty:	Medium
Blooms:	Apply

Small multiples are especially helpful when:

1. Comparing trends across many categories using the same axes and scales
2. Rendering a single 3D surface
3. Computing z-scores
4. Encoding packet network flows
5. Estimating kernel bandwidth

Question ID:	M03-Q24
Topic:	Horizon Graph
Difficulty:	Hard
Blooms:	Analyze

Which statement best describes a horizon graph?

1. A layered time-series display that folds bands of values to save vertical space
2. A 3D bar chart for horizons and altitudes
3. A map projection for polar regions
4. A network centrality diagram
5. A chord diagram for gene expression

Question ID:	M03-Q25
Topic:	Radar Chart
Difficulty:	Medium
Blooms:	Apply

Radar charts are typically appropriate when:

1. Variables are positive and you want to compare multivariate profiles
2. You need to visualize negative-only values
3. You must encode geographic directions
4. You want to estimate probability density
5. You need exact correlation coefficients

Question ID:	M03-Q26
Topic:	Maps
Difficulty:	Easy
Blooms:	Remember

Which of the following is NOT listed among the map types in the lecture?

1. Flow map
2. Graduated symbol map
3. Choropleth
4. Cartogram
5. Sankey map

Question ID:	M03-Q27
Topic:	Matplotlib Save
Difficulty:	Easy
Blooms:	Remember

Which command saves a matplotlib figure to PDF as shown in the slides?

1. plt.save('figure.pdf')
2. fig.savefig('figure.pdf')
3. ax.savefigure('figure.pdf')
4. plt.writepdf(fig)
5. np.savetxt('figure.pdf')

Question ID:	M03-Q28
Topic:	Matplotlib Styling
Difficulty:	Easy
Blooms:	Remember

In the matplotlib example, which keyword argument sets the line's color?

1. linewidth
2. color
3. alpha
4. style
5. markerface

Question ID:	M03-Q29
Topic:	Matplotlib API Style
Difficulty:	Medium
Blooms:	Understand

Which coding style does the lecture label as 'Lazy (try to avoid)'?

1. Using the stateful pyplot interface without figure/axes objects
2. Using the object-oriented fig/ax API
3. Saving figures to files
4. Calling NumPy linspace
5. Adding labels and titles

Question ID:	M03-Q30
Topic:	Seaborn Facets
Difficulty:	Medium
Blooms:	Apply

For faceted comparisons across categories, which approach is highlighted in the visualization lecture?

1. Seaborn's FacetGrid
2. Matplotlib's imshow
3. D3's force simulation
4. NetworkX spring layout
5. OpenCV's cvtColor

Question ID:	M06-Q1
Topic:	ML Process: Problem Framing
Difficulty:	Medium
Blooms:	Analyze

Which step most directly ensures that the problem you're solving is actually a machine-learning problem and not a simple rules or query task?

1. Check whether a labeled target exists and if patterns must generalize to new data
2. Visualize predictions with a confusion matrix
3. Collect more data first
4. Tune hyperparameters with cross-validation
5. Train a baseline linear model

Question ID:	M06-Q2
Topic:	ML Process: Data Splits
Difficulty:	Easy
Blooms:	Understand

In a standard ML workflow, a hold-out test set is primarily used to...

1. Select the best model during tuning
2. Fit the feature scaler and imputer
3. Estimate the final generalization performance after all modeling choices are frozen
4. Balance class labels in the training data
5. Visualize learning curves

Question ID:	M06-Q3
Topic:	ML Process: Leakage Prevention
Difficulty:	Medium
Blooms:	Apply

Which practice best prevents **data leakage** when scaling features?

1. Fit the scaler only on the training data, then apply the fitted transform to validation/test
2. Use MinMax scaling instead of standardization
3. Use more features so leakage averages out
4. Shuffle the rows before scaling
5. Fit the scaler on all available data, then transform splits

Question ID:	M06-Q4
Topic:	ML Process: Baselines
Difficulty:	Easy
Blooms:	Understand

A **baseline** model in the ML process is best described as...

1. A model with zero variance predictions
2. The final, most complex model
3. A simple, often naive model used to set a minimum performance bar
4. Any linear model
5. A model trained without regularization

Question ID:	M06-Q5
Topic:	ML Process: Feature Engineering
Difficulty:	Hard
Blooms:	Evaluate

During feature engineering, creating target-aware features (e.g., averaging the target by category using all rows) mainly risks...

1. Worse calibration but correct ranking
2. High bias
3. Data leakage that inflates validation scores
4. Underfitting
5. Improved interpretability with no downside

Question ID:	M06-Q6
Topic:	ML Process: Pipelines
Difficulty:	Medium
Blooms:	Analyze

Which statement about **pipelines** is most accurate?

1. Pipelines only chain models, not transforms
2. Pipelines are slower but identical to manual code
3. Pipelines ensure that cross-validation folds fit transforms (impute/scale) using only the training fold each time
4. Pipelines prevent overfitting by adding noise
5. Pipelines eliminate the need for a test set

Question ID:	M06-Q7
Topic:	ML Process: Splits for Time Series
Difficulty:	Medium
Blooms:	Apply

Which split strategy is most appropriate for **time-series** forecasting?

1. Random k-fold cross-validation
2. Bootstrap resampling only
3. Stratified shuffle split
4. Leave-one-out cross-validation
5. Forward-chaining (expanding window) validation

Question ID:	M06-Q8
Topic:	ML Process: Lifecycle
Difficulty:	Easy
Blooms:	Remember

In CRISP-DM-like lifecycles, the step after 'Modeling' that checks fitness-for-use against stakeholder goals is...

1. Data understanding
2. Evaluation
3. Business understanding
4. Deployment
5. Data preparation

Question ID:	M06-Q9
Topic:	ML Process: Monitoring
Difficulty:	Medium
Blooms:	Analyze

Which warning most strongly indicates **concept drift** after deployment?

1. Feature distributions in production shift relative to training and error rises on recent labeled samples
2. Stable calibration curves
3. Steady validation accuracy
4. Lower variance in predictions
5. Slightly higher training loss

Question ID:	M06-Q10
Topic:	ML Process: Metric Fit
Difficulty:	Medium
Blooms:	Analyze

A confusion matrix is **not** the right tool to evaluate a model when...

1. You need class-wise errors
2. You want false positive rate
3. You must compare ranking quality at varying thresholds
4. You want recall per class
5. You are solving classification

Question ID:	M06-Q11
Topic:	Regression: OLS Objective
Difficulty:	Easy
Blooms:	Remember

In simple linear regression, the **least squares** estimator chooses coefficients that...

1. Equalize residuals across x
2. Minimize mean absolute error of residuals
3. Maximize R^2 directly
4. Minimize classification error
5. Minimize the sum of squared residuals

Question ID:	M06-Q12
Topic:	Regression: Assumptions
Difficulty:	Medium
Blooms:	Evaluate

Which situation most clearly violates a key linear regression assumption?

1. n ¿ p
2. Residuals are approximately normal
3. Residual variance increases with x (fan-shaped residual plot)
4. There are categorical predictors encoded with dummies
5. Predictors are scaled to zero-mean

Question ID:	M06-Q13
Topic:	Regression: Multicollinearity
Difficulty:	Medium
Blooms:	Analyze

Multicollinearity primarily affects which aspect of a linear model?

1. Feasibility of predictions
2. Training MSE only
3. Interpretation of intercept only
4. Variance/stability of coefficient estimates
5. Unbiasedness of OLS coefficients

Question ID:	M06-Q14
Topic:	Regression: Overfitting
Difficulty:	Easy
Blooms:	Understand

You add a perfectly predictive but noisy feature. Training MSE drops sharply; validation MSE rises. The model most likely...

1. Underfit
2. Overfit due to high variance
3. Suffers from label leakage
4. Is unbiased
5. Has perfect calibration

Question ID:	M06-Q15
Topic:	Regression: Model Comparison
Difficulty:	Medium
Blooms:	Analyze

Compared to **R^2**, ****Adjusted R^2**** is preferred for model comparison because it...**

1. Is invariant to scaling of y
2. Penalizes added predictors that don't improve fit enough
3. Always increases when you add predictors
4. Is threshold-independent
5. Equals correlation squared between y and $\hat{y}$ always

Question ID:	M06-Q16
Topic:	Regression: Ridge
Difficulty:	Medium
Blooms:	Understand

Ridge regression differs from OLS by...

1. Guaranteeing sparsity
2. Adding an L1 penalty on coefficients
3. Optimizing MAE instead of MSE
4. Adding interaction terms automatically
5. Adding an L2 penalty that shrinks coefficients toward zero

Question ID:	M06-Q17
Topic:	Regression: Lasso
Difficulty:	Medium
Blooms:	Analyze

The **Lasso** is especially useful when...

1. You only have categorical variables
2. All predictors are essential
3. You need unbiased estimates regardless of p ¿¿ n
4. You want feature selection via many coefficients exactly zero
5. You need grouped shrinkage

Question ID:	M06-Q18
Topic:	Regression: Model Complexity
Difficulty:	Easy
Blooms:	Understand

Polynomial regression of degree 10 on small n most likely increases...

1. Sample size
2. Linearity
3. Bias
4. Variance
5. Homoskedasticity

Question ID:	M06-Q19
Topic:	Regression: Preprocessing
Difficulty:	Easy
Blooms:	Apply

Centering and scaling predictors before regularized regression mainly...

1. Ensures the penalty treats coefficients comparably across features
2. Makes intercept exactly zero
3. Changes predictions drastically
4. Improves RMSE regardless of data
5. Eliminates multicollinearity

Question ID:	M06-Q20
Topic:	Regression: Dummy Variables
Difficulty:	Medium
Blooms:	Apply

In multiple regression with a categorical variable of k levels, correct dummy encoding requires...

1. Dropping the intercept and using k-1 dummies and one interaction
2. k-1 dummy columns with an intercept (reference level)
3. Only one dummy column regardless of k
4. k dummy columns plus intercept
5. Target encoding by default

Question ID:	M06-Q21
Topic:	Evaluation: Cross-Validation
Difficulty:	Easy
Blooms:	Understand

Which statement about **k-fold cross-validation** is correct?

1. It uses the test set k times
2. It provides an estimate of generalization by repeatedly training on k-1 folds and validating on the remaining fold
3. It cannot be used with pipelines
4. Stratification is only for regression
5. It eliminates the need for a final test set

Question ID:	M06-Q22
Topic:	Evaluation: Metrics for Imbalance
Difficulty:	Medium
Blooms:	Analyze

For **imbalanced** binary classification, which metric is most informative across thresholds?

1. Accuracy
2. Precision-Recall (PR) curve / Average Precision
3. ROC AUC only
4. Explained variance
5. R^2

Question ID:	M06-Q23
Topic:	Evaluation: Metric Matching
Difficulty:	Easy
Blooms:	Remember

Which pair is correctly matched?

1. Calibration ↔ Adjusted R^2
2. Ranking ↔ ROC AUC
3. Classification ↔ RMSE
4. Clustering ↔ MAE
5. Regression ↔ F1-score

Question ID:	M06-Q24
Topic:	Evaluation: Confusion Matrix
Difficulty:	Easy
Blooms:	Remember

Given a confusion matrix, **recall** for the positive class equals…

1. FP / (FP + TN)
2. TP / (TP + FP)
3. TP / (TP + TN)
4. TN / (TN + FP)
5. TP / (TP + FN)

Question ID:	M06-Q25
Topic:	Evaluation: Calibration
Difficulty:	Medium
Blooms:	Apply

When calibration matters (e.g., risk estimation), which is best practice?

1. Optimize only for accuracy
2. Replace the loss with hinge loss
3. Threshold at 0.5 for all datasets
4. Use isotonic or Platt scaling on a validation set and check calibration curves
5. Prefer hard labels to probabilities

Question ID:	M06-Q26
Topic:	Evaluation: Regression Metrics
Difficulty:	Easy
Blooms:	Understand

Which metric is **scale-sensitive and penalizes larger errors more strongly?**

1. R^2
2. MAE
3. F1-score
4. Accuracy
5. RMSE

Question ID:	M06-Q27
Topic:	Evaluation: Threshold Metrics
Difficulty:	Easy
Blooms:	Understand

You care equally about precision and recall at a single operating point. Which metric best summarizes this?

1. Matthews correlation
2. Average precision
3. Balanced accuracy
4. F1-score
5. ROC AUC

Question ID:	M06-Q28
Topic:	Evaluation: Nested CV
Difficulty:	Hard
Blooms:	Analyze

During hyperparameter tuning, **nested cross-validation** is used to...

1. Speed up grid search
2. Balance classes automatically
3. Provide an outer loop for unbiased performance estimation while inner CV selects hyperparameters
4. Reduce data leakage from scaling
5. Avoid creating a test set

Question ID:	M06-Q29
Topic:	Evaluation: Averaging
Difficulty:	Medium
Blooms:	Analyze

Which scenario calls for **macro-averaged** F1 instead of micro-averaged F1?

1. You want to give equal weight to each class regardless of frequency
2. You're evaluating regression models
3. You want to weight larger classes more
4. Binary classification only
5. Classes are balanced and you want an overall rate

Question ID:	M06-Q30
Topic:	Evaluation: ROC Interpretation
Difficulty:	Medium
Blooms:	Evaluate

An ROC curve that lies **below** the diagonal indicates...

1. Random performance
2. High calibration error only
3. A perfect classifier
4. Systematic reversal of labels; flipping the score sign would perform above chance
5. Data leakage

Question ID:	M07-Q1
Topic:	Classification overview
Difficulty:	Easy
Blooms:	Understand

In supervised classification, what is the primary goal when some rows have known labels?

1. To delete rows without labels so training is cleaner
2. To reduce dimensionality for visualization only
3. To generate synthetic labels for all rows from noise
4. To label previously unlabeled rows using a learned model
5. To cluster the data into groups without labels

Question ID:	M07-Q2
Topic:	Applications of classification
Difficulty:	Easy
Blooms:	Remember

Which example task is most clearly a multi-class classification problem?

1. Predicting house prices in dollars
2. Grouping unlabeled news articles by topic
3. Estimating the mean of a Gaussian distribution
4. Finding a low-dimensional embedding of images
5. Assigning each handwritten digit image a class from 0–9

Question ID:	M07-Q3
Topic:	Usual steps / preprocessing
Difficulty:	Medium
Blooms:	Analyze

Which sequence best reflects the 'Usual Steps' emphasized for building a classifier in these slides?

1. Preprocessing & data cleaning → Feature finding/selection → Train & evaluate
2. Hyperparameter search only → Deploy
3. Visualization only → Deploy model
4. Train model → Collect data → Deploy → Clean data
5. Feature engineering → Ignore preprocessing → Train

Question ID:	M07-Q4
Topic:	KNN and effect of k
Difficulty:	Medium
Blooms:	Understand

For k-Nearest Neighbors (kNN), increasing k typically has what effect on the decision boundary?

1. Has no effect on the boundary; only runtime changes
2. Converts the classifier into a linear separator
3. Smooths the boundary by averaging over more neighbors
4. Forces the classifier to overfit training data
5. Makes the boundary more jagged and sensitive to noise

Question ID:	M07-Q5
Topic:	Evaluation protocol
Difficulty:	Medium
Blooms:	Evaluate

What principle about test data is stressed on the evaluation slide?

1. Any data used to help prediction must not later be used to change the model
2. Only accuracy matters when evaluating
3. Test sets should be larger than training sets
4. You may tune the model after peeking at the test set once
5. It's fine to mix training and test if you cross-validate

Question ID:	M07-Q6
Topic:	Cross-validation
Difficulty:	Easy
Blooms:	Understand

Cross-validation is introduced primarily to combat which issue?

1. Overfitting and unreliable estimates from a single split
2. Class imbalance exclusively
3. Label noise only
4. GPU memory limits
5. Too many features

Question ID:	M07-Q7
Topic:	1-D decision threshold
Difficulty:	Medium
Blooms:	Apply

In one-dimensional two-class separation, the 'overlap region' between class distributions most directly corresponds to:

1. Training time
2. Regularization strength
3. Bayesian prior probability
4. Model bias
5. Classification error rate

Question ID:	M07-Q8
Topic:	Otsu thresholding
Difficulty:	Easy
Blooms:	Remember

Otsu's method chooses a threshold by optimizing which quantity?

1. Equalizing class priors
2. Minimizing within-class variance only
3. Maximizing inter-class variance (between-class)
4. Minimizing training loss
5. Maximizing overall accuracy on test data

Question ID:	M07-Q9
Topic:	LDA intuition
Difficulty:	Medium
Blooms:	Analyze

Linear Discriminant Analysis (LDA) can be viewed as finding:

1. A nonlinear kernel mapping to infinite dimensions
2. A decision tree that partitions space by axis-aligned cuts
3. A clustering of unlabeled data by k-means
4. A random forest that averages many trees
5. A projection that maximizes between-class separation relative to within-class scatter

Question ID:	M07-Q10
Topic:	LDA pros/cons
Difficulty:	Easy
Blooms:	Remember

Which is listed as a *pro* of LDA in the deck?

1. Nonlinear boundaries by default
2. Requires deep networks to perform well
3. Often overfits small datasets
4. Usually doesn't overfit and works with much less data
5. Slow classification at inference

Question ID:	M07-Q11
Topic:	Naïve Bayes assumptions
Difficulty:	Easy
Blooms:	Understand

Why is Naïve Bayes called 'naïve' in these slides?

1. It assumes features are independent given the class
2. It assumes k is chosen by cross-validation
3. It assumes infinite training data
4. It assumes a linear decision boundary
5. It assumes labels are uniformly distributed

Question ID:	M07-Q12
Topic:	Logistic regression
Difficulty:	Easy
Blooms:	Remember

Logistic regression models the log-odds (logit) as:

1. A sum of kernel functions over support vectors
2. A linear function of input features ($w^T x$)
3. A decision tree depth
4. A constant independent of input features
5. The reciprocal of a Gaussian density

Question ID:	M07-Q13
Topic:	Sigmoid link
Difficulty:	Easy
Blooms:	Remember

In the logistic regression derivation provided, which activation function maps z=w^{T}x to a probability p?

1. σ(z)=eˆz/(1+eˆz)
2. Hard threshold at z=0
3. tanh(z)
4. Softplus(z)=log(1+eˆz)
5. ReLU(z)=max(0,z)

Question ID:	M07-Q14
Topic:	Linear separators
Difficulty:	Easy
Blooms:	Understand

Which statement about linear classifiers is consistent with the slides?

1. They never misclassify overlapping classes
2. They can only be used for regression
3. They require kNN distance voting
4. They find a line/plane/hyperplane that separates classes
5. They search for polynomial decision boundaries of degree $\geq$2

Question ID:	M07-Q15
Topic:	Baselines to try first
Difficulty:	Medium
Blooms:	Remember

Which of the following is *not* one of the four 'brain-dead' baseline algorithms the slides recommend trying first?

1. Logistic Regression
2. Support Vector Machines with RBF kernel
3. Naïve Bayes
4. Linear Discriminant Analysis
5. k-Nearest Neighbors

Question ID:	M07-Q16
Topic:	Projection quality
Difficulty:	Medium
Blooms:	Analyze

Which plot-based intuition is used in the LDA section to contrast poor vs. better projections?

1. Calibration curves
2. Box plots of residuals
3. ROC curves
4. Precision–recall curves
5. Histograms of the projected feature

Question ID:	M07-Q17
Topic:	Feature engineering motivation
Difficulty:	Medium
Blooms:	Understand

The 'Feature Problem' slide motivates which broader idea?

1. Labels are optional if we have enough features
2. We should avoid features and learn end-to-end only
3. Only image datasets need features
4. Going from raw inputs to informative features is crucial
5. Preprocessing is unnecessary with deep learning

Question ID:	M07-Q18
Topic:	Transparent vs Opaque models
Difficulty:	Easy
Blooms:	Understand

Which pair best exemplifies the slides' distinction between transparent and opaque models?

1. Random forests (transparent) vs. logistic regression (opaque)
2. Decision trees (transparent) vs. deep neural networks (opaque)
3. k-means (transparent) vs. linear regression (opaque)
4. Autoencoders (transparent) vs. KNN (opaque)
5. GANs (transparent) vs. PCA (opaque)

Question ID:	M07-Q19
Topic:	Feature importance in logistic regression
Difficulty:	Medium
Blooms:	Apply

According to the M07B slides, what is a *critical* step before using logistic regression weights as feature importance?

1. Apply PCA to all features
2. Normalize or standardize features to comparable scales
3. Remove the intercept term
4. Use L1 regularization
5. Convert all features to binary indicators

Question ID:	M07-Q20
Topic:	Post-hoc model-agnostic tools
Difficulty:	Easy
Blooms:	Understand

Which statement about LIME and SHAP matches the deck?

1. They require the model to be a decision tree
2. They are post-hoc explanation libraries that can be applied to black boxes
3. They are training algorithms for deep networks
4. They only provide global explanations, not local ones
5. They directly change model weights to be interpretable

Question ID:	M07-Q21
Topic:	Local vs Global
Difficulty:	Easy
Blooms:	Understand

Which pair correctly contrasts local vs. global explanations as discussed?

1. Local—visualize weights; Global—visualize a single gradient
2. Local—train a smaller model; Global—train a larger model
3. Local—understand model overall; Global—explain one prediction
4. Local—explain a single prediction; Global—summarize model behavior across the dataset
5. Local—optimize hyperparameters; Global—optimize loss

Question ID:	M07-Q22
Topic:	Surrogate models
Difficulty:	Medium
Blooms:	Apply

Which technique is an example of a *surrogate model* approach?

1. Ensembling many neural networks into a single predictor
2. Normalizing inputs to zero mean
3. Using dropout at test time
4. Computing the gradient of the loss w.r.t. inputs
5. Training an interpretable decision tree to mimic a complex model

Question ID:	M07-Q23
Topic:	Feature effects
Difficulty:	Medium
Blooms:	Understand

Partial Dependence Plots (PDP) and Individual Conditional Expectation (ICE) primarily help with:

1. Understanding feature effects on predictions
2. Speeding up training
3. Detecting data leakage
4. Generating counterfactuals by optimization
5. Estimating label noise

Question ID:	M07-Q24
Topic:	Counterfactuals
Difficulty:	Medium
Blooms:	Analyze

A counterfactual explanation is best described as:

1. A visualization of gradients in a neural network
2. An example from the training set closest to a query point
3. A proof that the model is globally optimal
4. The minimal change to an input that would change the prediction
5. Random noise added to test robustness

Question ID:	M07-Q25
Topic:	Motivation for XAI
Difficulty:	Easy
Blooms:	Understand

Which of the following best captures the motivation slide 'Why do we need explainable models?'

1. To increase training speed only
2. To support trust, debugging, fairness, and accountability in decisions (e.g., 'Why didn't I get the loan?')
3. To reduce the need for labeled data
4. Because explainability always improves accuracy
5. So we never need test sets again

Question ID:	M07-Q26
Topic:	Practical transparency
Difficulty:	Medium
Blooms:	Analyze

Which statement aligns with the slides' nuance that 'both can use black-box methods'?

1. Even simple models can act as black boxes if complexity or access prevents interpretation
2. Opaque models are always interpretable
3. Only neural networks are black boxes by definition
4. Transparent models cannot be used in black-box fashion
5. Black-box methods require no validation

Question ID:	M07-Q27
Topic:	Feature importance methods
Difficulty:	Medium
Blooms:	Understand

Permutation Feature Importance (PFI) belongs to which family of explanation methods?

1. Fairness constraints at training time
2. Counterfactual generation by gradient descent
3. Exact Shapley value computation for trees only
4. Perturbation-based, model-agnostic post-hoc importance
5. Model training objectives

Question ID:	M07-Q28
Topic:	XAI libraries
Difficulty:	Medium
Blooms:	Remember

Which pairing is accurate per the slides' library list?

1. SHAP — by scikit-learn core team only
2. LIME — exclusively for image models
3. Captum — official TensorFlow library
4. ELI5 — IBM's enterprise XAI suite
5. InterpretML — Microsoft's explainability toolkit

Question ID:	M07-Q29
Topic:	TreeSHAP
Difficulty:	Medium
Blooms:	Understand

TreeSHAP is a specialized variant of SHAP intended for:

1. Kernel density estimators
2. Convolutional neural networks only
3. Tree-based ensemble models
4. Hidden Markov Models
5. Model-agnostic linear models

Question ID:	M07-Q30
Topic:	Fairness notions (context)
Difficulty:	Medium
Blooms:	Evaluate

Which statement about fairness and explainability is consistent with the surrounding course materials?

1. Choosing one fairness metric can create trade-offs with others
2. Fairness is solved by increasing model capacity
3. All statistical definitions of fairness are mutually compatible
4. Only demographic parity matters in practice
5. Explainability eliminates the need for fairness analysis

Question ID:	MT-Q1
Topic:	N/A
Difficulty:	Easy
Blooms:	N/A