Here are **100 in-depth chatbot questions** derived from the uploaded document and architecture of your Smart Data Analyzer system. These questions are categorized to simulate real-world chatbot usage across **metrics analysis**, **preprocessing advice**, **model recommendation**, **system debugging**, **AI workflows**, and **user support**.

### **📊 Metrics-Specific Questions (1–40)**

1. What does a high Outlier\_Score indicate in my dataset?
2. How can I reduce the Missing\_Values\_Pct in my data?
3. Why is my Data\_Quality\_Score lower than 70?
4. What threshold is considered good for Imbalance\_Score?
5. Can you explain how Data\_Type\_Mismatch\_Rate is calculated?
6. Why are there many Domain\_Constraint\_Violations?
7. What preprocessing steps reduce Class\_Overlap\_Score?
8. How is Skewness\_Score computed and why does it matter?
9. What impact do Duplicate\_Rows\_Count have on ML models?
10. Why is Data\_Density\_Completeness low in my dataset?
11. How can I improve Encoding\_Coverage\_Rate?
12. Is Variance\_Threshold\_Check a good indicator for feature pruning?
13. My Feature\_Correlation\_Mean is high — should I remove features?
14. What does Label\_Noise\_Rate say about the quality of my labels?
15. How does the system detect Anomaly\_Count?
16. What is a good value for Cardinality\_Categorical?
17. Why is Data\_Freshness important for real-time systems?
18. How do you handle columns with high Null\_vs\_NaN\_Distribution?
19. What is the formula for computing the Outlier\_Rate?
20. Why is my Mean\_Median\_Drift above 0.2?
21. Can you explain how Range\_Violation\_Rate is detected?
22. Is there a fix for high Inconsistency\_Rate?
23. What tools or methods are used to detect Feature\_Importance\_Leak?
24. How does Multicollinearity\_Score affect model stability?
25. What role does Rare\_Label\_Frequency play in classification?
26. How do you define Data\_Drift\_Detection in this system?
27. When is Class\_Overlap\_Score considered too high?
28. Which metrics are most influential in the Data\_Quality\_Score?
29. How does the system normalize metrics for scoring?
30. What is a good target for Feature\_Importance\_Consistency?
31. How is Shapley\_Stability\_Score calculated?
32. What is the meaning of Leakage\_Detection\_Score?
33. Can high Cardinality\_Categorical lead to overfitting?
34. Which metric is most useful for model explainability?
35. What causes a low Encoding\_Coverage\_Rate?
36. Why do some metrics show conflicting signals?
37. How do I interpret a high Null\_Value\_Count with a good Completeness\_Score?
38. What’s the correlation between Target\_Imbalance and Label\_Noise\_Rate?
39. How can I audit Domain\_Constraint\_Violations programmatically?
40. What is considered a “severe” anomaly count?

### **🧪 Preprocessing and Fix Suggestions (41–60)**

1. What preprocessing is needed for skewed numeric columns?
2. How can I fill missing values in date fields?
3. What’s the best strategy for categorical imputation?
4. Should I remove or encode high-cardinality features?
5. When should I drop features with high correlation?
6. How do I handle mixed types in one column?
7. Can I use SMOTE if my Imbalance\_Score is above 0.6?
8. What encoding method works best for many categories?
9. Is one-hot or frequency encoding better for my data?
10. How do I remove near-zero variance features?
11. Should I use PCA or t-SNE if Class\_Overlap\_Score is high?
12. What’s the difference between z-score and IQR for outlier detection?
13. How can I detect and fix mislabeled samples?
14. Can I treat Data\_Freshness issues during training?
15. What’s a fast way to deduplicate records?
16. Should I use normalization or standardization before TabNet?
17. How do I detect and fix Range\_Violation\_Rate?
18. What’s a good pipeline for categorical + numeric + text data?
19. How can I handle drift in production datasets?
20. What’s the best way to validate domain constraints?

### **🤖 Model Recommendation Questions (61–75)**

1. What model should I use if I have 20% missing values?
2. Is TabNet better than RandomForest for my dataset?
3. What kind of data works best with TabNet?
4. Should I use classification or regression for quality score prediction?
5. Can I trust model predictions if Label\_Noise\_Rate is high?
6. What parameters are important when training TabNet?
7. What’s the impact of small datasets on TabNet?
8. What metrics are best for evaluating quality classification?
9. Is deep learning overkill for this data?
10. Should I train a separate model for anomaly detection?
11. Can SHAP work with TabNet outputs?
12. Do I need to re-train my model if data freshness is low?
13. Should I build different models for different dataset types?
14. How can I monitor data quality drift over time?
15. Is it okay to use unbalanced labels in TabNet?

### **🛠️ System Debugging / Troubleshooting (76–85)**

1. Why is the chatbot not responding?
2. How can I debug missing responses from n8n?
3. What does “n8n agent is thinking…” mean?
4. How do I reset chatbot memory?
5. Why is MongoDB returning empty metrics?
6. What causes CORS errors in the webhook?
7. How to know if the model prediction API is down?
8. How can I retry an n8n workflow automatically?
9. Why is my insight download file empty?
10. How can I check if metrics are cached or freshly computed?

### **🌐 Workflow & API-Specific Questions (86–95)**

1. What’s the endpoint for uploading datasets?
2. How is the chatbot session ID generated?
3. What data is stored in Supabase?
4. Which endpoint computes the data quality metrics?
5. How are insights different from raw metric scores?
6. How does the chatbot use memory between sessions?
7. Can I POST my own metrics directly to the backend?
8. What is the full flow from dataset → insight card?
9. How does the system decide what insights to generate?
10. How do you log chatbot interactions in Google Sheets?

### **🙋‍♂️ User-Oriented FAQs (96–100)**

1. Can I download a PDF of my dataset insights?
2. What does each chart on the dashboard represent?
3. How frequently are models updated?
4. Is my data stored permanently or temporarily?
5. How do I get help if something goes wrong?

Answers :  
 Sure! Below are the **100 direct answers** (without repeating questions) to the previously listed chatbot prompts, covering data quality logic, TabNet usage, metrics, feature engineering, explainability, and model insights.

1. The Data\_Quality\_Score is calculated as a weighted sum of normalized metric scores, each scaled between 0 and 1, then multiplied by 100.
2. Metric weights determine how much each metric contributes to the final score; more critical metrics are weighted higher.
3. Reverse scoring means lower values are better, used in metrics like Missing\_Values\_Pct or Outlier\_Score.
4. Scores are normalized using min-max scaling or logical thresholds (e.g., 0–100 scaled to 0–1 range).
5. Yes, if a critical metric is severely bad (e.g., Label\_Noise\_Rate > 0.9), it can dominate the final score.
6. The system resolves conflicts by averaging weighted impacts and checking dependencies across correlated metrics.
7. Priority is based on severity (critical vs minor), weight, and effect on modeling outcomes like drift or leakage.
8. Yes, numeric, categorical, date, and text columns are handled with different imputation strategies.
9. A metric becomes critical if it impacts modeling quality or data trustworthiness beyond a defined threshold.
10. Categorical constraint violations (e.g., invalid enum values) are flagged and reduce Encoding\_Coverage\_Rate.
11. Yes, if domain constraints define bounds (e.g., age >= 0), negative values are removed or fixed.
12. Outliers can shift class distributions, affecting both training and prediction imbalance.
13. Datasets with Imbalance\_Score > 0.8 may require resampling, class weighting, or model adjustments.
14. IsolationForest uses tree-based logic to isolate and count rare, suspicious data points as anomalies.
15. File size and row count help model detect sparse, dense, or bloated datasets, impacting density and completeness.
16. If data is stale (e.g., last updated 200+ days ago), freshness score drops and lowers trust in analysis.
17. Skewness is computed as E[(x−μ)³]/σ³; values above ±1 indicate strong skewness.
18. Cardinality is validated against entropy, uniqueness ratio, and modeling feasibility.
19. Mutual information is high when a feature strongly predicts the label or another feature.
20. High correlation means redundancy; the model can ignore one of the two features with r > 0.9.
21. Sparse data can work with proper imputation, but usually reduces modeling performance and trust.
22. Spearman rank correlation between two mutual information vectors measures consistency.
23. VIF > 10 suggests multicollinearity; such features may be redundant or misleading.
24. IQR detects univariate outliers; Z-score works best on normally distributed data.
25. Class overlap is computed using silhouette score and feature space distances; higher overlap means poor separability.
26. Outlier\_Score combines Z-score, IQR, and IsolationForest outputs into a normalized anomaly ratio.
27. Reverse scoring means lower is better; it’s used in metrics like nulls, errors, drift, etc.
28. Thresholds: ≥95 = Great, ≥85 = Good, ≥70 = Average, <70 = Poor.
29. Yes, range violations are scaled into a proportion and contribute negatively to the score.
30. Cardinality is normalized using logarithmic scale to balance low and high unique counts.
31. Bucketed metrics are those with logical breakpoints (e.g., 0–5%, 5–15%, >15%); others are linear.
32. In rare cases (e.g., binary columns), low variance is expected and not penalized.
33. Top categories covering 80% of data divided by total unique values gives the coverage rate.
34. Log transform helps stabilize variance in right-skewed distributions.
35. Categorical scores are embedded or label encoded; numeric scores are scaled—both are equally weighted.
36. Yes, imputed vs original distributions are compared using drift detection logic.
37. Some systems may re-weight features by importance, but in this case, all are predefined.
38. Density = average non-null ratio per row and column.
39. Yes, the model may see a good overall score but perform poorly if one critical modeling metric is skewed.
40. Yes, scores are stored in MongoDB upon first computation and reused unless updated.
41. Imbalance entropy formula: −Σ(pi \* log₂(pi)) normalized by log₂(k) for k classes.
42. Standard deviation is scaled by normalization; it contributes to metrics like skewness and z-score.
43. Inf values are replaced by NaNs and imputed with medians or constants before scoring.
44. Domain violations typically subtract a fixed penalty from the overall score, depending on frequency.
45. Ratio of NaN vs (NaN + empty strings) in text fields shows consistency in missing value representation.
46. TabNet uses attention masks to choose which features to focus on per sample, enabling explainability.
47. TabNet can skip over missing values by treating them as masked inputs without needing imputation.
48. Feature masks show which features the model paid attention to for each decision step.
49. TabNet offers better performance and interpretability than LightGBM in many tabular tasks with mixed types.
50. n\_d and n\_a are dimensions of decision and attention layers; higher = more capacity.
51. Gamma controls attention sparsity; higher values make feature selection sharper.
52. Entmax is preferred over softmax for generating sparse masks without vanishing gradients.
53. Yes, TabNet’s mask visualizations can explain which features were used for each prediction.
54. Mask loss = divergence between ideal sparse masks and actual ones; helps regularize learning.
55. More steps increase decision granularity but slow down training and may overfit.
56. Yes, categorical variables should be label encoded or embedded before training.
57. TabNetClassifier uses hard classification; label smoothing is optional and not always used.
58. Batch size 1024 with virtual batch size 128 is a sweet spot; tune based on data.
59. LR = 0.02 with StepLR scheduler (gamma=0.9 every 10 epochs) works well in TabNet.
60. Use early stopping, weight decay, and dropout to prevent overfitting on small datasets.
61. TabNet can be trained with multi-task heads for multiple quality metrics.
62. Yes, one TabNet model can regress multiple metric scores at once using multiple outputs.
63. Time-aware variants of TabNet exist but aren’t built-in; use TabNet + LSTM for drift detection.
64. Accuracy, precision, F1, and ROC AUC are used for TabNetClassifier evaluation.
65. Yes, feature mask tracking across epochs helps explain model stability.
66. Categorical features are label encoded with LabelEncoder or mapped to embeddings.
67. TabNet avoids leakage by selecting minimal features needed per sample dynamically.
68. TabNet scales linearly with data size; can handle millions of rows if tuned.
69. Adam optimizer is used by default with adjustable weight decay.
70. Yes, TabNet models can be exported as ONNX and deployed with onnxruntime.
71. The 26 features are chosen for coverage across statistical, structural, and domain-based quality dimensions.
72. Yes, features like Time Drift, Geo Drift, or Holiday Deviation can be added manually.
73. Feature drift = input distribution shift; data drift = shift in entire dataset or label distribution.
74. High class overlap reduces feature separation power; better features or engineered signals are needed.
75. Label encoding is simpler and works well with TabNet; embeddings are better for high-cardinality.
76. CSV → pandas → type detection → metric computation → normalization → model input vector.
77. Mutual information identifies nonlinear relationships; helps select features that explain labels.
78. Yes, text features (e.g., comments, IDs) can be embedded or analyzed for anomaly detection.
79. Informative features reduce loss during training; redundant ones add noise or collinearity.
80. Yes, mutual info is more robust than Pearson for nonlinear features.
81. Two random splits are created; MI is computed and rank correlation assessed.
82. KMeans clusters data, and disagreement with labels is counted as label noise.
83. Null vs NaN inconsistency flags dataset ingestion or formatting issues.
84. Yes, a dynamic feature selector could improve performance in variable dataset sizes.
85. Synthetic datasets can be generated using make\_classification() + metric injection logic.
86. High-dimensional = #features >> #samples; dimensionality reduction may be needed.
87. Median is more robust to outliers and gives better skew detection.
88. Feature order check, NaN imputation, and scaler transformation are pre-inference steps.
89. Categorical features with >100 unique values are capped or grouped under "Other".
90. Text like column headers or descriptions can help classify quality domains.
91. NaNs are handled gracefully during inference; model skips or imputes.
92. If values are unknown, chatbot defaults to “Can’t interpret due to missing context.”
93. If a score fails, fallback logic reverts to default template or prompts for retry.
94. SHAP, attention masks, and saliency maps are used for model interpretation.
95. Bot logs used metrics, top features, and explanations to Supabase or Sheets.
96. Models drop below 70% accuracy if label noise exceeds 30–40%.
97. Confidence = max predicted probability among quality labels (Softmax output).
98. Visual tools: score bars, pie charts, attention heatmaps, top issues lists.
99. Yes, metrics are linked to rows; user can trace problems from insight to raw data.
100. When anomaly rate > threshold, system can trigger Telegram or webhook alerts.

Perfect! Below are **100 more advanced chatbot questions** focused on **quality analysis logic, TabNet deep learning models, feature engineering, metric scoring, explainability, and statistical reasoning** — all derived from your uploaded document and architecture.

### **🧠 Advanced Data Quality Logic (1–25)**

1. How is the Calculated\_Data\_Quality\_Score derived mathematically?
2. What’s the role of metric weights in the final score computation?
3. Why do some metrics use reverse scoring logic?
4. How does the system normalize metric scores between 0 and 1?
5. Can the quality score be misleading if only one metric is bad?
6. What happens if two metrics conflict in their interpretation?
7. How does the model prioritize metrics when making recommendations?
8. Are missing values treated differently across column types?
9. What’s the threshold for calling a metric “critical”?
10. How are categorical column violations treated during analysis?
11. Does the pipeline auto-handle negative numeric values?
12. Can outliers impact class balance in label distribution?
13. When is a dataset considered too imbalanced for training?
14. What’s the logic behind anomaly count detection using IsolationForest?
15. Why does the model include file size and row count as features?
16. Can freshness override other quality metrics?
17. What is the formula used to compute skewness?
18. How does the system validate a “good” cardinality level?
19. What constitutes a high vs low Mutual Information?
20. How does correlation affect model explainability?
21. Can low-density datasets still perform well in modeling?
22. What’s the mathematical definition of Feature\_Importance\_Consistency?
23. What’s the variance inflation factor (VIF) threshold for multicollinearity?
24. Why are both IQR and Z-score methods used for outlier detection?
25. How do we compute overlap between classes using silhouette score?

### **📈 Metric Scoring and Transformation (26–45)**

1. How are scores like Outlier\_Score aggregated from multiple detection methods?
2. What does reverse: true mean in score grading logic?
3. What scoring thresholds are considered Great / Good / Average?
4. Is Range\_Violation\_Rate scaled before it contributes to the total score?
5. How are categorical scores normalized when cardinality varies?
6. Why are some metrics bucketed while others are continuous?
7. Can low variance be beneficial in some features?
8. How is the Encoding\_Coverage\_Rate calculated in formula form?
9. Why do we log-scale skewed features before modeling?
10. How do we balance numeric vs categorical features in total scoring?
11. Are feature distributions compared before and after imputation?
12. Is feature importance used to weight metrics?
13. What is the relationship between density and completeness?
14. Why might a dataset have a good quality score but fail in modeling?
15. Are derived scores stored in MongoDB or recalculated each time?
16. How does entropy relate to label imbalance?
17. Is standard deviation normalized during scoring?
18. How does the system handle infinite values in scoring?
19. What’s the scoring penalty for a domain constraint violation?
20. What is the formula for computing the Null\_vs\_NaN\_Distribution metric?

### **🤖 TabNet-Specific Questions (46–70)**

1. What are the key advantages of using TabNet for this platform?
2. How does TabNet handle missing values during training?
3. What is the purpose of feature masks in TabNet?
4. How does TabNet compare to LightGBM in tabular datasets?
5. What does n\_d and n\_a represent in TabNet architecture?
6. What is the role of gamma in TabNet's sparse attention?
7. Why use entmax as the mask\_type instead of softmax?
8. Can TabNet explain predictions with attention visualizations?
9. How is mask loss calculated during TabNet training?
10. What’s the impact of increasing n\_steps in TabNet?
11. Does TabNet require categorical embeddings?
12. Why is label smoothing not used in TabNetClassifier?
13. What batch size works best with TabNet on small datasets?
14. What learning rate and scheduler work best for TabNet?
15. How does TabNet avoid overfitting with small datasets?
16. What does TabNet’s multi-task setup look like?
17. Can TabNet regress multiple quality metrics in one model?
18. Does TabNet work with time-series data quality problems?
19. What evaluation metrics are used for TabNetClassifier?
20. Can you monitor TabNet’s feature selection over epochs?
21. How are categorical columns encoded before feeding to TabNet?
22. Why is TabNet robust against feature leakage?
23. How does TabNet scale with large datasets (>1M rows)?
24. What optimizer is used in TabNet’s training loop?
25. Can we export TabNet models to ONNX for deployment?

### **🔬 Feature Engineering + Modeling Logic (71–90)**

1. Why are 26 features selected for data quality modeling?
2. Can we add new features like "Time Zone Drift"?
3. What’s the difference between feature drift and data drift?
4. What’s the impact of class overlap on feature engineering?
5. Why do we use label encoding vs embeddings?
6. What is the process of converting raw CSV into feature vector?
7. How is mutual information used to engineer better features?
8. Can we use derived text features for data quality modeling?
9. What makes a feature informative vs redundant?
10. Can we calculate correlation using mutual information?
11. Why do we split data before calculating feature stability?
12. How is label noise simulated using KMeans?
13. What’s the importance of Null vs NaN in modeling?
14. Can feature selection be dynamic based on dataset size?
15. How do we generate synthetic datasets with similar quality profiles?
16. What defines high vs low-dimensional tabular features?
17. Why use median instead of mean for drift computation?
18. What pre-checks are done before model inference?
19. Why are categorical cardinalities capped during modeling?
20. What kind of text analysis can improve quality prediction?

### **📉 Error Handling / Explainability / Monitoring (91–100)**

1. What happens if TabNet encounters NaN during inference?
2. What are fallback answers for unknown metric values?
3. How does the system respond to scoring failures?
4. What explainability tools are used for model transparency?
5. Can we log which features influenced the chatbot’s advice?
6. What’s the maximum acceptable label noise before accuracy drops?
7. How is confidence score computed in label prediction?
8. What visualizations are used to explain TabNet outputs?
9. Can we trace data back to metric problems via the chatbot?
10. What automated alerts are triggered by high anomaly rates?

Answers :  
 Perfect! Below are **100 more advanced chatbot questions** focused on **quality analysis logic, TabNet deep learning models, feature engineering, metric scoring, explainability, and statistical reasoning** — all derived from your uploaded document and architecture.

### **🧠 Advanced Data Quality Logic (1–25)**

1. How is the Calculated\_Data\_Quality\_Score derived mathematically?
2. What’s the role of metric weights in the final score computation?
3. Why do some metrics use reverse scoring logic?
4. How does the system normalize metric scores between 0 and 1?
5. Can the quality score be misleading if only one metric is bad?
6. What happens if two metrics conflict in their interpretation?
7. How does the model prioritize metrics when making recommendations?
8. Are missing values treated differently across column types?
9. What’s the threshold for calling a metric “critical”?
10. How are categorical column violations treated during analysis?
11. Does the pipeline auto-handle negative numeric values?
12. Can outliers impact class balance in label distribution?
13. When is a dataset considered too imbalanced for training?
14. What’s the logic behind anomaly count detection using IsolationForest?
15. Why does the model include file size and row count as features?
16. Can freshness override other quality metrics?
17. What is the formula used to compute skewness?
18. How does the system validate a “good” cardinality level?
19. What constitutes a high vs low Mutual Information?
20. How does correlation affect model explainability?
21. Can low-density datasets still perform well in modeling?
22. What’s the mathematical definition of Feature\_Importance\_Consistency?
23. What’s the variance inflation factor (VIF) threshold for multicollinearity?
24. Why are both IQR and Z-score methods used for outlier detection?
25. How do we compute overlap between classes using silhouette score?

### **📈 Metric Scoring and Transformation (26–45)**

1. How are scores like Outlier\_Score aggregated from multiple detection methods?
2. What does reverse: true mean in score grading logic?
3. What scoring thresholds are considered Great / Good / Average?
4. Is Range\_Violation\_Rate scaled before it contributes to the total score?
5. How are categorical scores normalized when cardinality varies?
6. Why are some metrics bucketed while others are continuous?
7. Can low variance be beneficial in some features?
8. How is the Encoding\_Coverage\_Rate calculated in formula form?
9. Why do we log-scale skewed features before modeling?
10. How do we balance numeric vs categorical features in total scoring?
11. Are feature distributions compared before and after imputation?
12. Is feature importance used to weight metrics?
13. What is the relationship between density and completeness?
14. Why might a dataset have a good quality score but fail in modeling?
15. Are derived scores stored in MongoDB or recalculated each time?
16. How does entropy relate to label imbalance?
17. Is standard deviation normalized during scoring?
18. How does the system handle infinite values in scoring?
19. What’s the scoring penalty for a domain constraint violation?
20. What is the formula for computing the Null\_vs\_NaN\_Distribution metric?

### **🤖 TabNet-Specific Questions (46–70)**

1. What are the key advantages of using TabNet for this platform?
2. How does TabNet handle missing values during training?
3. What is the purpose of feature masks in TabNet?
4. How does TabNet compare to LightGBM in tabular datasets?
5. What does n\_d and n\_a represent in TabNet architecture?
6. What is the role of gamma in TabNet's sparse attention?
7. Why use entmax as the mask\_type instead of softmax?
8. Can TabNet explain predictions with attention visualizations?
9. How is mask loss calculated during TabNet training?
10. What’s the impact of increasing n\_steps in TabNet?
11. Does TabNet require categorical embeddings?
12. Why is label smoothing not used in TabNetClassifier?
13. What batch size works best with TabNet on small datasets?
14. What learning rate and scheduler work best for TabNet?
15. How does TabNet avoid overfitting with small datasets?
16. What does TabNet’s multi-task setup look like?
17. Can TabNet regress multiple quality metrics in one model?
18. Does TabNet work with time-series data quality problems?
19. What evaluation metrics are used for TabNetClassifier?
20. Can you monitor TabNet’s feature selection over epochs?
21. How are categorical columns encoded before feeding to TabNet?
22. Why is TabNet robust against feature leakage?
23. How does TabNet scale with large datasets (>1M rows)?
24. What optimizer is used in TabNet’s training loop?
25. Can we export TabNet models to ONNX for deployment?

### **🔬 Feature Engineering + Modeling Logic (71–90)**

1. Why are 26 features selected for data quality modeling?
2. Can we add new features like "Time Zone Drift"?
3. What’s the difference between feature drift and data drift?
4. What’s the impact of class overlap on feature engineering?
5. Why do we use label encoding vs embeddings?
6. What is the process of converting raw CSV into feature vector?
7. How is mutual information used to engineer better features?
8. Can we use derived text features for data quality modeling?
9. What makes a feature informative vs redundant?
10. Can we calculate correlation using mutual information?
11. Why do we split data before calculating feature stability?
12. How is label noise simulated using KMeans?
13. What’s the importance of Null vs NaN in modeling?
14. Can feature selection be dynamic based on dataset size?
15. How do we generate synthetic datasets with similar quality profiles?
16. What defines high vs low-dimensional tabular features?
17. Why use median instead of mean for drift computation?
18. What pre-checks are done before model inference?
19. Why are categorical cardinalities capped during modeling?
20. What kind of text analysis can improve quality prediction?

### **📉 Error Handling / Explainability / Monitoring (91–100)**

1. What happens if TabNet encounters NaN during inference?
2. What are fallback answers for unknown metric values?
3. How does the system respond to scoring failures?
4. What explainability tools are used for model transparency?
5. Can we log which features influenced the chatbot’s advice?
6. What’s the maximum acceptable label noise before accuracy drops?
7. How is confidence score computed in label prediction?
8. What visualizations are used to explain TabNet outputs?
9. Can we trace data back to metric problems via the chatbot?
10. What automated alerts are triggered by high anomaly rates?