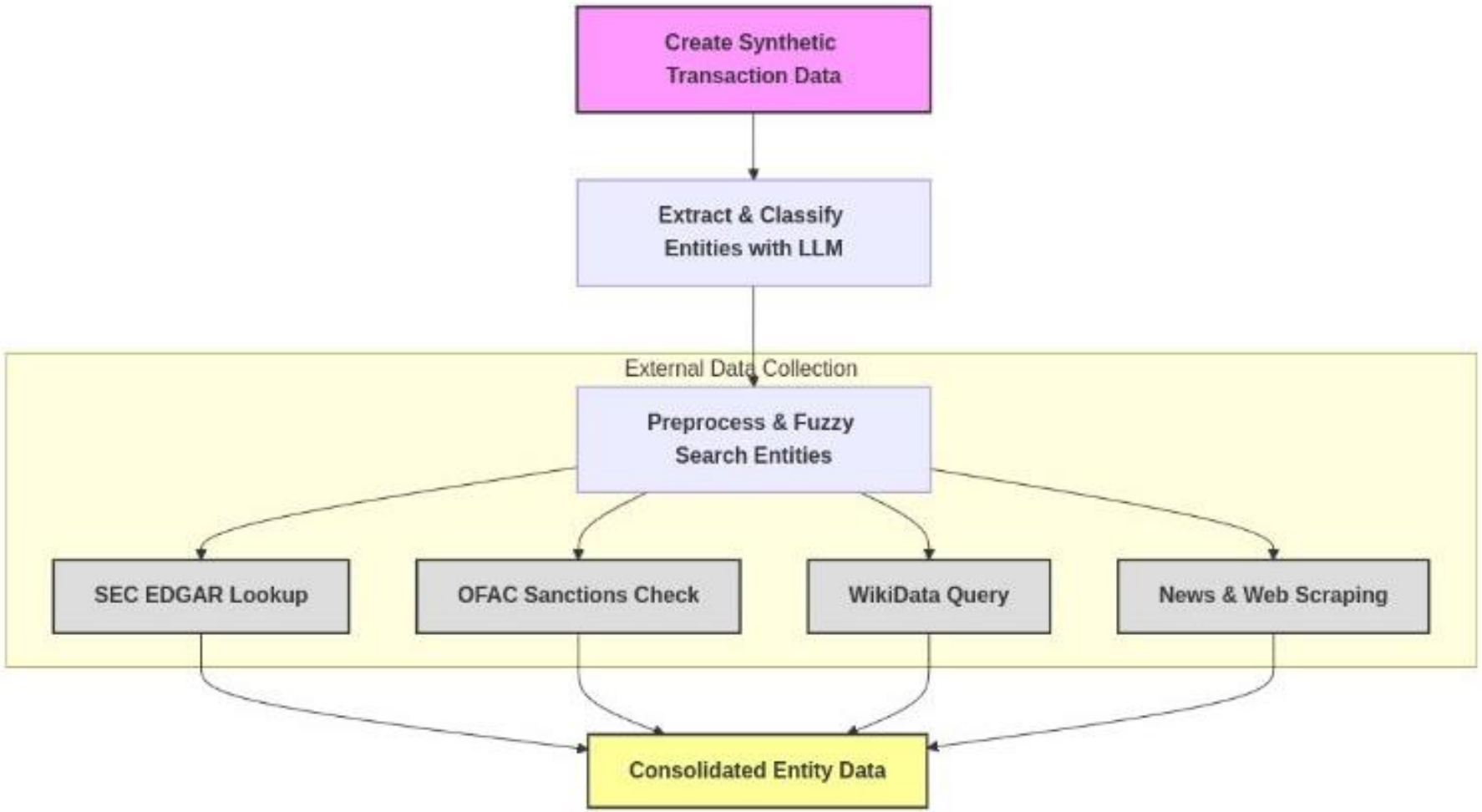
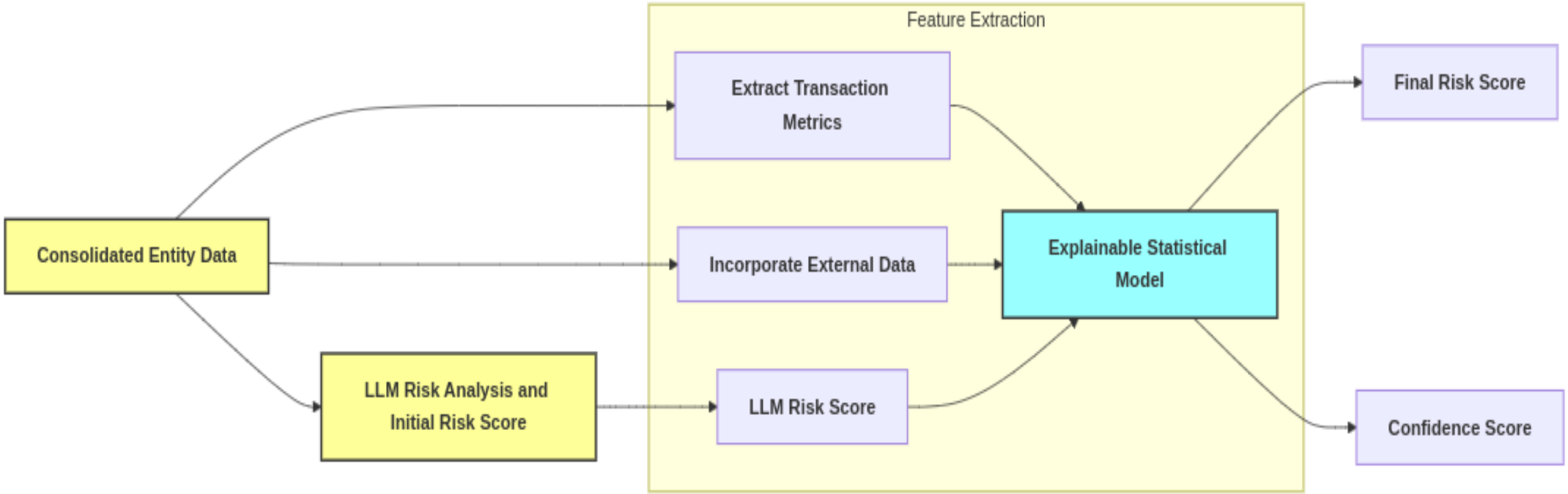
AI Driven Entity Intelligence and Risk Analysis

Risk Scoring Implementation Pipeline



Risk Scoring Implementation Pipeline



Synthetic Dataset Generation

To generate synthetic transaction data, we established empirical assumptions on risk levels and considered five key risk factors. Structured data was first generated algorithmically using the Open Sanctions dataset. Mistral 7B SLM was then used to add additional information and convert it into an unstructured format.

Sender & Receiver Risk

- High Risk: Politically Exposed Persons (PEPs), Anonymous or Shell Companies
- Medium Risk: Non-Profit Organizations in High-Risk Regions, Unregistered or New Entities
- Low Risk: Registered, Well-Known Entities

Merchant Risk

- High Risk: Gambling, Crypto
- Medium Risk: Luxury Items (Watches, Jewelry, Art)
- Low Risk: Essential Services (Groceries, Utilities)

Amount & Currency Risk:

- Medium Risk: Large Transactions (Above $100K), Crypto Transactions in High-Risk Countries
- Low Risk: Regular Currency Transactions (< $100K btw Corporates & < $10K btw Individuals

Country Risk

- High Risk: Transactions involving OFAC-Sanctioned countries
- Medium Risk: Transactions involving Tax Havens (Cayman Islands, Panama, etc.)
- Low Risk: Transactions in stable countries (US, UK, Canada, etc.)

Payment Method Risk

- High Risk: Anonymous Crypto payments
- Medium Risk: Wire transfers to offshore accounts
- Low Risk: Bank Transfers & Credit Cards

Synthetic Dataset Generation

Limitations & Future Improvements:

- **Correlations between risk factors were not accounted in data generation:** The dataset was created by selecting each risk factor independently, which may not fully reflect real-world dependencies. Using real transaction data could help mitigate this limitation.
  For example, If a merchant operates in a high-risk industry (e.g., luxury goods, real estate), the transaction is more likely to involve high-value purchases rather than small, routine payments.
- **Balancing Legitimate & Fraudulent Transactions:** Real-world data is highly imbalanced, with less than 1% of transactions being high risk. Training on such skewed data would be challenging due to small dataset size, which might introduce potential bias. Using a balanced dataset (equal high and low-risk transactions) would not reflect real-world fraud detection scenarios, leading to an unrealistic model that overestimates risk.

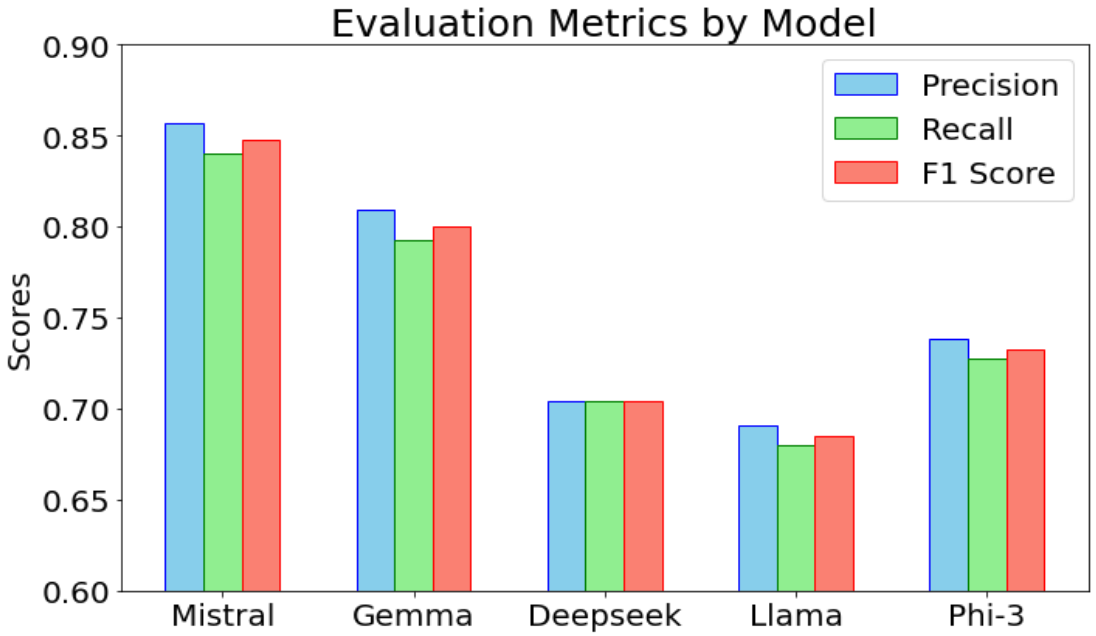
To prevent over-estimating risk, we assume only flagged transactions are being analyzed, where 80% of the transactions are false positives and only 20% are truly high risk. Based on this assumption, we generated a synthetic dataset with below distribution:

- 50% Low Risk
- 30% Medium Risk
- 20% High Risk

Note: Synthetic transaction data are available in `code/src/SyntheticData/`

Entity Recognition and Classification

- Entities within structured and unstructured transaction data were extracted and categorized as Individuals, Corporates, Non-Profit Organizations, Government Agencies, and Politically Exposed Persons (PEPs).
- Five LLMs (Mistral, Google's Gemma 3, DeepSeek R1, Meta's Llama-3.3, and Microsoft's Phi-3) with free API access and entity recognition capabilities were evaluated based on precision, recall, and F1-score. An entity was considered a true positive only if both its name and category were classified correctly.
- Mistral-Small-3.1-24B-Instruct achieved the highest F1-score, minimizing both false positives and false negatives, and was selected for integration into the pipeline.



Note: *Mistral-Small-3.1-24B-Instruct was not further fine-tuned due to compute limitations. However, integrating fine-tuning into the pipeline could further optimize performance.*
Code and sample transaction predictions are available in `code/src/Entity_Recognition.ipynb`

Data Source Integration

LLMs are limited to the information available in their training data. Therefore, recent details about organizations must be provided separately to improve risk scoring. To address this, we integrated four external data sources:

- **SEC EDGAR:**
  - Entity Registration: Checked for corporate registration in SEC EDGAR. All U.S. publicly traded companies are registered in SEC EDGAR.
  - 8-K Filings: Checked for 8-K filings in last 2 years if registered. These filings report significant corporate events such as bankruptcies, mergers, acquisitions, or executive changes, which can impact financial stability and risk assessment.
- **WikiData:**
  - Entity Registration: Checked for company or individual presence in WikiData properties.
  - Checked for mentions in properties indicating legal issues or controversies, such as:
    - P1365 (replaces, indicating leadership changes due to controversy)
    - P576 (dissolution date, which can indicate bankruptcies)
    - P1056 (product or service, which helps detect high-risk industries)
- Checked whether entities appear on OFAC's Specially Designated Nationals (SDN) list, indicating sanctions, money laundering risks, or terrorism-related activities.
- Extracted the top 3 most recent news articles mentioning the organization's name using News API

Entity Matching and Normalization

Data taken from different sources have inconsistencies in entity names. To handle inconsistencies (e.g., "Samsung Electronics," "Samsung," and "Samsung Electronics Pvt Ltd" should be recognized as the same entity) we followed the below approach for data source integration

- **Data Pre-processing and Standardization**
  - Pre-processed entity names by converting them to lowercase, removing special characters, and stripping common business suffixes like "Ltd" and "Solutions" to improve database search accuracy.
- **Reducing Search Space with Cosine Similarity**
  - Since advanced string-matching algorithms are computationally expensive, we used cosine similarity to narrow down potential matches.
  - We transformed company names into n-grams and applied a TF-IDF transformation, creating a sparse matrix where only relevant n-gram elements were filled. By computing the dot product between the dataset matrix and the query name matrix, we obtained cosine similarity scores. We then used a partition function to select the top five best matches.
- **Fuzzy String Matching**
  - To refine the results, we applied discounted Levenshtein distance, which measures the number of edits (substitutions, insertions, deletions) needed to transform one string into another.
  - The discounted version penalizes differences at the beginning of the string more than at the end, as suffix variations (e.g., "Inc.", "Corp.") are common in corporate names. Companies with a similarity score above 90% were considered identical.

Data Source Integration

```python
company = "Lehman Brothers"
results = company_screening.screen_company(company)
print(json.dumps(results, indent=4))
```


```
{
    "Company": "Lehman Brothers",
    "SEC Registered": "Yes",
    "CIK": "0001437948",
    "Recent 8-K Filings": "None",
    "OFAC Sanctioned": "No",
    "Closest OFAC Database Match": "None",
    "Wikidata_QID": "Q212900",
    "Description": "defunct American financial services firm",
    "Scandals": [
        "bankruptcy of Lehman Brothers",
        "https://www.wikidata.org/wiki/Q3269580"
    ],
    "Recent News": [
        "FTX\u2019s US$950 million bankruptcy fees among costliest since Lehman Brothers",
        "FTX\u2019s US$950 million bankruptcy fees among costliest since Lehman Brothers"
    ]
}
```


Note: *Code and sample outputs are available in* `code/src/Corporate_Data_Extractor.ipynb`

Data Source Integration

In [6]:

```
company = "angl carribean"
results = company_screening.screen_company(company)
print(json.dumps(results, indent=4))
```

```
{
    "Company": "angl carribean",
    "SEC Registered": "Yes",
    "CIK": "Not available",
    "Recent 8-K Filings": "None",
    "OFAC Sanctioned": "Yes",
    "Closest OFAC Database Match": "ANGLO-CARIBBEAN CO., LTD.",
    "Status": "Not Found in Wikidata",
    "Recent News": "No relevant news found"
}
```

Sample output demonstrating efficient handling of corporate names in database search: The input "angl carribean" successfully matches "ANGLO-CARIBBEAN CO., LTD." in the OFAC Sanctions database.

Multi-Shot In-Context Learning

Considering the limitations on compute resources and the usage of free API resources, the most efficient way for language models to learn and adapt to current data is through multi-shot in-context learning. To achieve this, we used the Mistral 7B SLM along with generated synthetic data to predict risk scores and provide reasoning.

In [12]:
```
analyzer = TransactionAnalyzer("ofac_list.txt")
result_json = analyzer.analyze_transaction(transactions[1])
risk_model_input = f"{transactions[1]}\n{result_json}"
result = RiskAnalyzer().analyze_risk(combined_inputs[:56], risk, risk_model_input)
print(result)
```

```
{
    "Transaction ID": "TKN-2023-7020",
    "Extracted Entities": ["Quantum Holdings Ltd", "Maria Gonzalez", "Golden Sands Trading FZE", "Deut
sche Bank", "Emirates NBD", "Mr. Viktor Petzov"],
    "Entity Type": ["Corporation", "Person", "Corporation", "Bank", "Bank", "Politically Exposed Perso
n"],
    "Risk Score": 0.8,
    "Confidence Score": 0.5,
    "Reason": "The transaction involves Quantum Holdings Ltd, a corporation based in the British Virgi
n Islands, a known tax haven. The funds are routed through Deutsche Bank Frankfurt and Emirates NBD Du
bai. The approver, Mr. Viktor Petzov, is linked to an OFAC SDN List entry in 2022. Due to the lack of
information from external data sources, the risk score is only based on transaction data.",
    "Supporting Evidence": ["SEC EDGAR", "OFAC Sanctions List", "News Articles", "Wikidata"]
}
```

Sample prediction for Unstructured Transaction data

Multi-Shot In-Context Learning

In [16]:

```
analyzer = TransactionAnalyzer("ofac_list.txt")
result_json = analyzer.analyze_transaction(transactions[5])
risk_model_input = f"{transactions[5]}\n{result_json}"
analyzer = RiskAnalyzer()
result = analyzer.analyze_risk(combined_inputs[:56], risk, risk_model_input)
print(result)
```


```
{
   "Transaction ID": "TXN004",
   "Extracted Entities": ["Green Earth Org", "CCMI"],
   "Entity Type": ["Non-Profit Organization", "Corporation"],
   "Risk Score": 0.5,
   "Confidence Score": 0.6,
   "Reason": "The transaction involves funding for an environmental project with a payer being a non-
profit organization and the receiver being a corporation based in the Cayman Islands, a known tax have
n. The provided risk score is based on transaction data since no external data sources are found for C
CMI.",
   "Supporting Evidence": ["Transaction Data", "SEC EDGAR", "OFAC Sanctions List", "News Articles",
"Wikidata"]
}
```


Sample prediction for Structured Transaction data

Note: The supporting evidence is based solely on the provided external data sources. Free-tier language models do not have the capability to fetch exact references from the web. To prevent mis-referencing, only integrated data sources are used as supporting evidence.

Statistical Model and Explainability

The risk score obtained from LLM is not based on a verifiable metric, and running the same prompt multiple times may yield slightly different results. To ensure explainability, traditional statistical models like logistic regression can be used to predict the final risk score. This model incorporates features from external verifiable data sources (such as SEC registration status and presence in the OFAC sanctions list) alongside the LLM-predicted risk score. The final risk score is explainable through the trained weights of the statistical model, providing transparency in decision-making.

Note: This step was not completely achieved and integrated into the pipeline due to time constraints. However, the provided code can generate the desired output from the LLM. Incorporating this step would enhance the model's explainability by grounding the risk score in verifiable statistical metrics.

Confidence Score Prediction

The confidence scores were derived by quantifying uncertainty in the predicted risk score. In the final explainable statistical model for risk prediction, features extracted from external data sources (SEC EDGAR, OFAC Sanctions, WikiData etc) provide deterministic information, while uncertainty arises solely from the LLM-predicted risk score. The confidence scores were derived following the below approach:

Numerical Quantification:

The LLM prediction was repeated 5 times, and the standard deviation (σ) of the scores was used to measure uncertainty

Scaling Confidence Score:

- Since risk values lie in [0,1], the standard deviation (σ) falls within $0 < \sigma < 0.5$.
- The confidence score was computed as: Confidence Score = $1 - 2\sigma$ (this ensures, confidence scores lie between 0 and 1)
- If all predictions are identical ($\sigma = 0$), confidence is 1 (high confidence).
- If predictions vary widely ($\sigma \approx 0.5$), confidence is 0 (low confidence).

Note: *Code and sample outputs are available in* `code/src/Uncertainty_Quantification.ipynb`