ICT304 Tutorial 3

Table of Contents

1. Types of Sampling Methods

1.1 Probability Sampling

In *probability sampling*, each member of the population has a known, non-zero chance of being selected. It is ideal for producing representative and unbiased samples.

1.1.1 Simple Random Sampling (SRS)

- **Description:** Everyone in the population has an equal chance of being selected.
- **Example:** Using a random number generated to select 50 students from a list of 500.
- **Advantages:** Reduce bias and is highly representative of the population.
- **Disadvantages:** Requires a complete list of the population and can be inefficient with large populations.

1.1.2 Stratified Sampling

- **Description:** The population is divided into subgroups (strata), and random samples are taken from each.
- **Example:** Dividing a company's employees into departments and selecting random samples from each department.
- **Advantages:** Ensures all subgroups are represented, providing better precision in estimates.
- **Disadvantages:** Requires knowledge of subgroups, making organising more complex.

1.1.3 Systematic Sampling

- **Description:** Select every nth individual from a list.
- **Example:** Choosing every 10^{th} person from a list of 1000 individuals.
- **Advantages:** Easy to implement without random number generation.
- **Disadvantages:** This can introduce bias if the list has hidden patterns.

1.1.4 Cluster Sampling

- **Description:** The population is divided into clusters. Some clusters are randomly selected, and all individuals within those clusters are sampled.
- **Example:** Select two cities at random and survey all residents within those cities.
- **Advantages:** Cost-efficient and practical for large populations.
- **Disadvantages:** Clusters may not be fully representative, leading to potential bias.

1.2 Non-Probability Sampling

In *non-probability sampling*, only some have a known or equal chance of being selected. It's used when probability sampling isn't feasible, but it's more prone to bias.

1.2.1 Convenience Sampling

- **Description:** Sample selected from a group that's easy to access.
- **Example:** Surveying people in a nearby shopping mall.
- **Advantages:** Quick, easy, and inexpensive.
- **Disadvantages:** There is a high risk of bias since the sample may not represent the population.

1.2.2 Purposive (Judgemental) Sampling

- **Description:** The researcher selects participants based on their judgement of who is most appropriate for the study.
- **Example:** Interviewing experts in a specific field for a specialised study.
- **Advantages:** Allows the researcher to focus on relevant individuals.
- **Disadvantages:** Highly subjective and less generalisable.

1.2.3 Snowball Sampling

- **Description:** Existing participants recruit future participants.
- **Example:** In a study of a niche community, participants are asked to refer others from the same group.
- **Advantages:** Useful for hard-to-reach or hidden populations.
- **Disadvantages:** This can lead to sampling bias as the sample grows based on participant networks.

2. Handling Imbalance Dataset

Refer to the codes folder submitted along with this document.

3. Differences Between Batch Processing and Stream Processing

3.1 Batch Processing

- **Definition:** Processes data in bulk at scheduled intervals or after accumulating enough data. Suitable for non-time-sensitive tasks.
- **Example:** Generating payroll reports by processing all employee data at the end of the month.
- **Advantages:**
    - Efficient for large datasets.
    - Easier to manage and schedule during off-peak hours.
    - Less complexity in implementation.
- **Disadvantages:**
    - High latency results are delayed until the batch is processed.
    - Not suitable for real-time data processing or dynamic systems.

3.2 Stream Processing

- **Definition:** Continuously process data as it arrives, often in real-time. Used when immediate results or decisions are needed.
- **Example:** Fraud detection systems that analyse financial transactions in real-time to detect suspicious activity.
- **Advantages:**
    - Low latency data is processed almost immediately.
    - Ideal for real-time decision-making.
    - Useful in dynamic systems (e.g., IoT, financial trading).
- **Disadvantages:**
    - More complex to implement and maintain.
    - Requires more computing resources for continuous operation.
    - Handling large streams in real-time can be challenging.

3.3 Machine Learning Techniques for Stream Processing

3.3.1 Online Learning (Incremental Learning)

- **Definition:** Models are updated continuously as new data arrives, rather than relying on a fixed dataset.
- **Example:** Spam filters that update their model as new emails are classified.
- **Methods:**
    - **Stochastic Gradient Descent (SGD):** Updates model weights incrementally with each new data point, suitable for online learning.

3.3.2 Sliding Window Techniques

- **Definitions:** Divides incoming data streams into smaller, manageable chunks (windows) for real-time analysis, discarding old data.
- **Example:** Predicting stock market trends by processing the last 5 minutes of trading data in a sliding window.
- **Methods:**
    - **Recurrent Neural Networks (RNNs):** Handle sequential data streams by keeping track of previous data points, ideal for time-series and stream processing.

4. Feature Importance and Feature Selection

4.1 How is Feature Importance Related to Feature Selection?

Definition of Feature Importance: Feature importance refers to the relevance of individual features in predicting the target variable in a machine learning model. It helps identify which features contribute most to the model's performance.

Definition of Feature Selection: Feature selection is selecting a subset of relevant features from the dataset, reducing dimensionality while retaining the most critical information for the model.

Relationship:

- **Feature Importance Drives Feature Selection:** Feature importance helps prioritise which features should be included in the model. You can discard irrelevant or redundant features by identifying the most relevant features.
- **Reduces Overfitting:** Selecting only important features can reduce overfitting, where the model performs well on training data but poorly on unseen data.
- **Improves Model Efficiency:** Using fewer but more important features reduces computational cost and makes the model more interpretable.
- **Example:** In a dataset predicting house prices, *location* and *size* may be deemed necessary, while *window type* might be less relevant. Feature importance metrics help filter out the unimportant features.

4.2 Approaches to Feature Extraction

Definition of Feature Extraction: Feature extraction involves transforming raw data into new features that better represent the underlying patterns, improving the model's predictive power.

4.2.1 Principal Component Analysis (PCA)

Description: A dimensionality reduction technique that transforms the original features into a smaller set of uncorrelated components called principal components. These components capture the maximum variance in the data.

Example: PCA is often used in image compression, where large amounts of pixel data are reduced to a smaller number of components that still capture the essential structure of the image.

Advantages:

- Reduce dimensionality without losing too much information.
- Simplifies models and improves efficiency.

Disadvantages:

- Components may not be readily interpretable.
- Can lose interpretability of the original features.

4.2.2 Linear Discriminant Analysis (LDA)

Description: Similar to PCA, LDA also reduces the number of features but focuses on maximising the separation between multiple classes in classification problems.

Example: LDA is used in face recognition, where the goal is distinguishing between different individuals in an image.

Advantages:

- Maximises class separability, making it useful for classification tasks.

Disadvantages:

- Assumes typically distributed data, which may not always be the case.

4.2.3 Autoencoders

Definition: Neural networks used for unsupervised feature extraction. Autoencoders learn a compressed data representation and reconstruct it from that compressed version. The compressed (hidden) layer captures essential features.

Example: Used in anomaly detection, where compressed representations of standard data are learned, and deviations are flagged as anomalies

Advantages:

- Can capture complex, non-linear relationships.
- Effective for tasks like image and text analysis.

Disadvantages:

- Requires more computational resources.
- May require extensive training and tuning.

4.2.4 t-Distributed Stochastic Neighbour Embedding (t-SNE)

Description: A technique for visualising high-dimensional data by reducing it to two or three dimensions while preserving the relationship between data points.

Example: t-SNE is widely used in clustering and visualising complex datasets, such as genetic or image data.

Advantages:

- Excellent for visualising complex, high-dimensional datasets.

Disadvantages:

- Computationally expensive for large datasets.
- Primarily used for visualisation, not for improving model performance.

4.3 Summary

Feature Importance is crucial for ***Feature selection***, as it helps identify the most relevant features, reducing dimensionality and improving model performance.

Several approaches to ***Feature extraction*** (e.g., PCA, LDA, Autoencoders, t-SNE) allow for transforming raw data into a better format for machine learning, each with its advantages and disadvantages.

5. Understanding Data Leakage in Machine Learning

5.1 Introduction to Data Leakage

Definition: Data leakage occurs when information from the training data that won't be available during actual predictions leaks into the model training process. These results are misleadingly high performance during training or validation but poor performance in production.

Impact: Leads to over-optimistic validation scores and poor generalisation in real-world scenarios.

5.2 Types of Data Leakage

Target Leakage:

Occurs when predictors include information that will not be available at prediction time. This typically happens when data is included that is only known after the target is determined. For Example, predicting pneumonia but including the variable ***took_antibiotic_medicine***. This variable is updated after the patient gets pneumonia, causing the model to overfit. This can be prevented by excluding updated features after the target variable is known.

Train-Test Contamination:

It occurs when training data influences validation or test data. This usually happens when the same preprocessing is applied to both the training and validation datasets, leading to overly optimistic results. For example, an imputer can be fitted for missing values before splitting data into train and test sets. This allows data from the test set to influence the model training process. This can be prevented by separating training and test data before any data preprocessing or using pipelines to ensure correct handling.

5.3 Detecting target Leakage: Case Study

Example: Credit card application dataset

Suspicious Variables: expenditure, share, majorcards, active

Investigation:

The results show that 100% of non-cardholders had no expenditures, while only 2% of cardholders had no expenditures. This suggests a leak, as the variable likely represents spending on the card after acceptance.

Action: Remove leaky variables (*expenditure, share, majorcards, active*), resulting in lower but more reliable model accuracy (98% to 83%).

5.4 Handling Train-Test Contamination

Best Practice: Always separate training and validation data before any preprocessing.

Use Pipelines: Scikit-learn pipelines ensure that preprocessing happens only on training data during cross-validation, preventing contamination.

5.5 Conclusion

Key Takeaways:

- Data leakage can significantly mislead model performance metrics, leading to poor results in production.
- Target leakage and train-test contamination can be avoided with careful data handling and tools like pipelines.
- While removing leakage may reduce apparent accuracy, the model will generalise better to new, unseen data.