LEARNING OBJECT CLASSIFICATION BASED ON PERSONALISATION

DATA SCIENCE PROJECT PRESENTATION

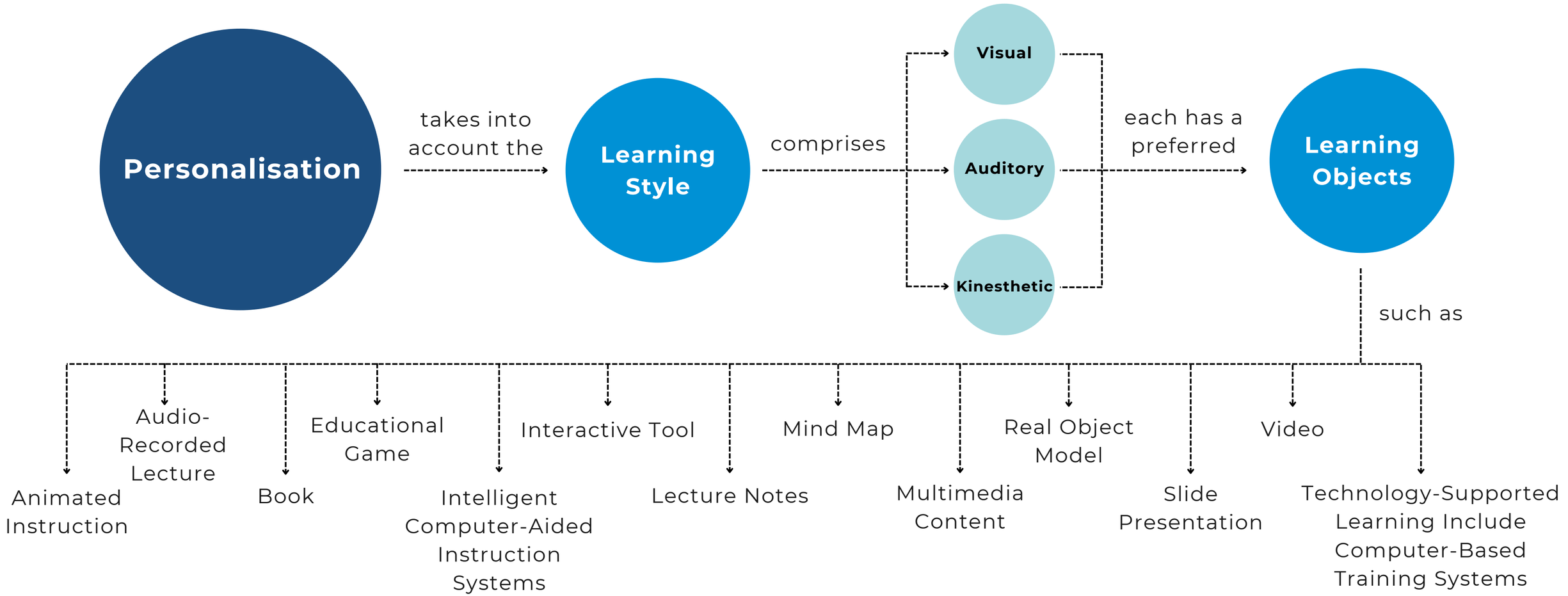
BY: JANICE CHONG SEE WAI (S2132420)

SUPERVISED BY: AP. DR. NOR LIYANA BT MOHD SHUIB

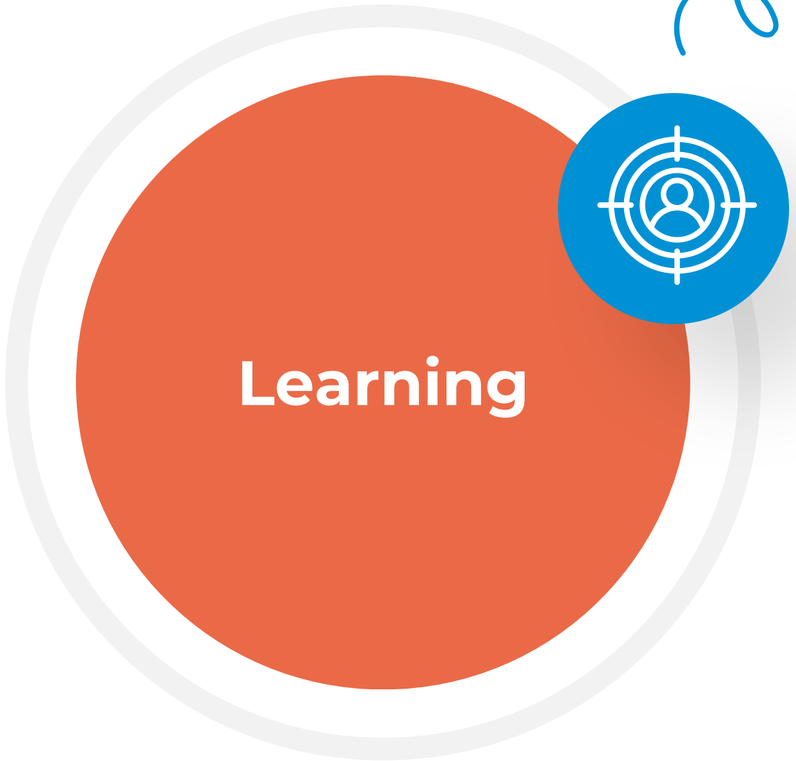
Overview

- Introduction
- Problem Statment
- Objectives
- Data Science Methodology
- Demonstration

Introduction: Overview



Introduction: Background

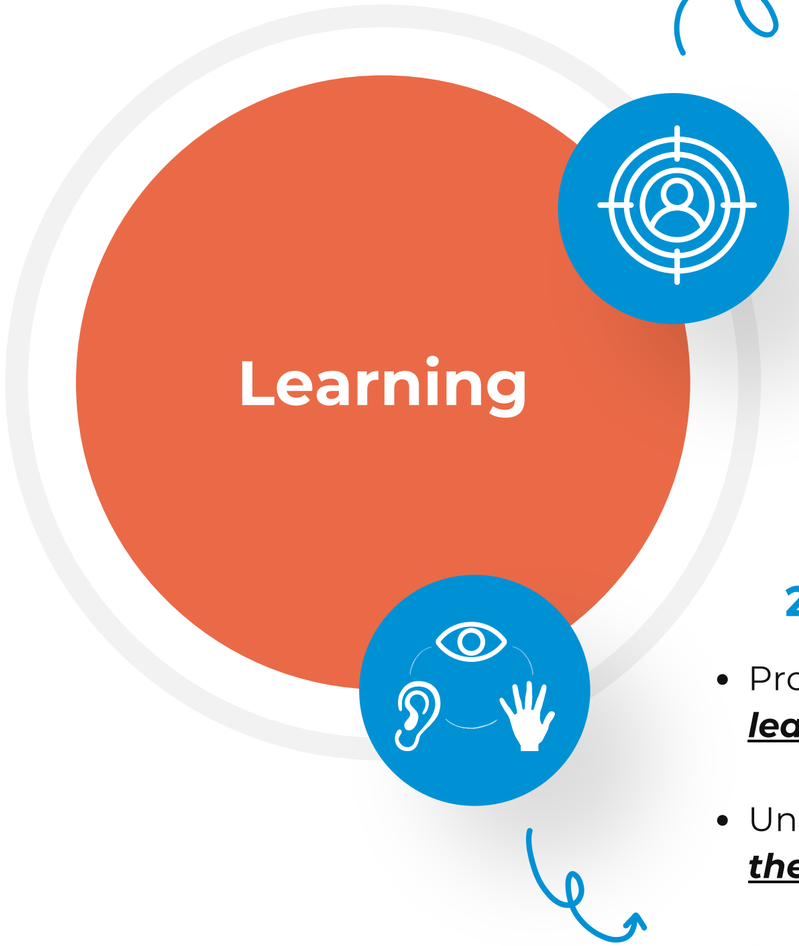


1.0 Personalisation

- ***Key*** to improving learning performance (Martin & Maria, 2019)

- According to Souabi et al., 2021, ***tailoring learning objects to align with students' learning styles and preferences*** can:
  - Enhance educational experience
  - Increases learners' performance and satisfaction

Introduction: Background



1.0 Personalisation

- ***Key*** to improving learning performance (Martin & Maria, 2019)

- According to Souabi et al., 2021, ***tailoring learning objects to align with students' learning styles and preferences*** can:
  - Enhance educational experience
  - Increases learners' performance and satisfaction

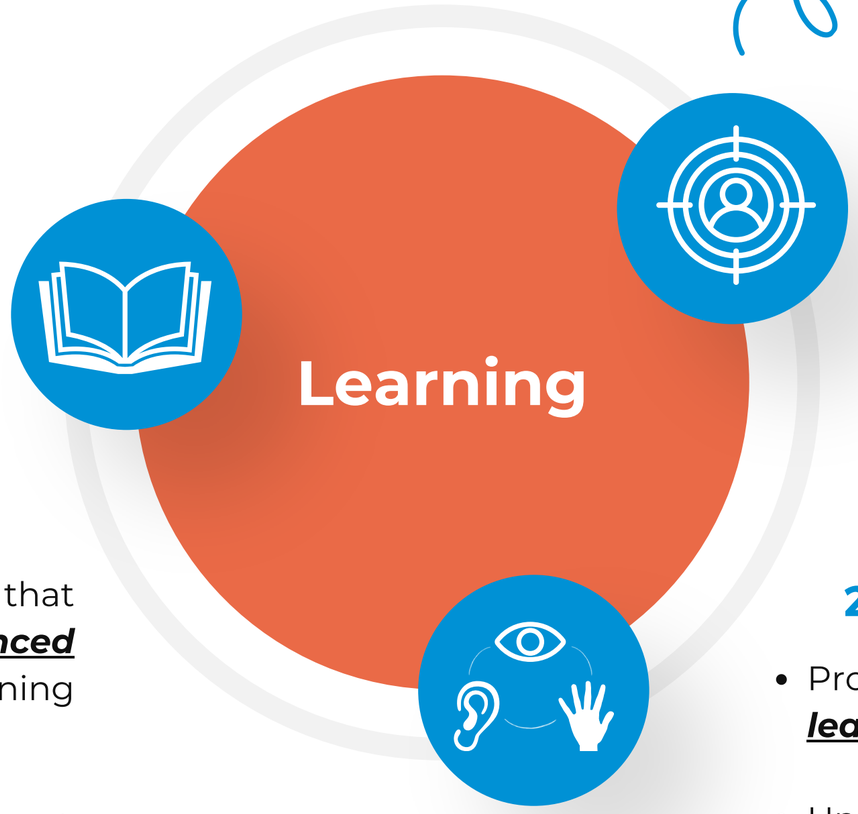
2.0 VAK Model

- Provides the learners with a ***profile of their learning styles*** (Rao, 2023)

- Understand the ***type of learning that best suits them***

- People learn by:
  - Seeing (**V**isual)
  - Hearing (**A**uditory)
  - Doing (**K**inesthetic)

Introduction: Background



1.0 Personalisation

- ***Key*** to improving learning performance (Martin & Maria, 2019)
- According to Souabi et al., 2021, ***tailoring learning objects to align with students' learning styles and preferences*** can:
  - Enhance educational experience
  - Increases learners' performance and satisfaction

2.0 VAK Model

- Provides the learners with a ***profile of their learning styles*** (Rao, 2023)
- Understand the ***type of learning that best suits them***
- People learn by:
  - Seeing (**V**isual)
  - Hearing (**A**uditory)
  - Doing (**K**inesthetic)

3.0 Learning Objects

- Any entity, ***digital or non-digital*** that can be ***used***, re-used or ***referenced*** during technology-supported learning (Demir & Almali, 2020)
- In a world of information overload, students today are ***exposed to extensive learning objects***.

Introduction: VAK Model

a. VAK learning style questionnaire

- 30 statements, each with 3 options (created by Chislett and Chapman in 2005).
- Respondents will need to **choose the options which best describe them**.

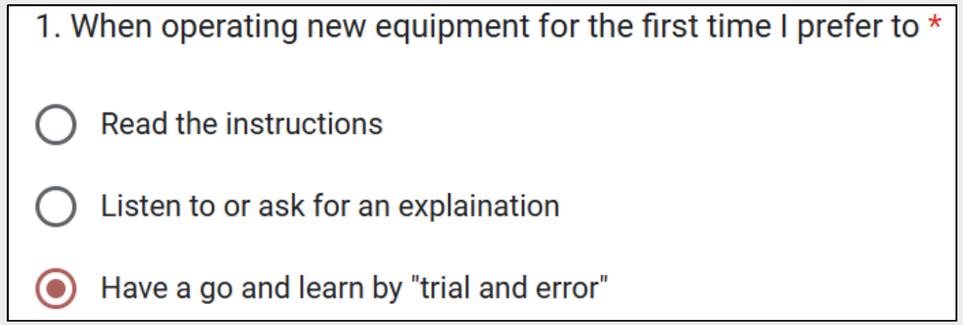
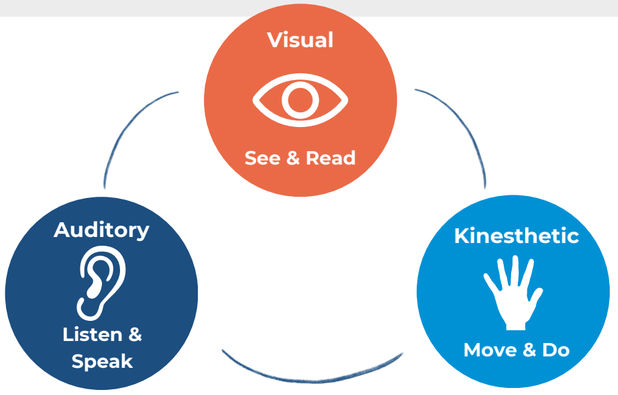


Figure 1: Sample VAK learning style question from the survey form

- Example based on Figure 1:
  - A **visual** learner would most probably choose to **read** the instructions
  - An **auditory** learner would prefer to **listen** to an explanation
  - A **kinesthetic** learner is likely to **have a go directly**

b. Determination of dominant learning style

- **Each option** of the VAK questions **represents a dominant learning style** (either Visual, Auditory or Kinesthetic).
- **Sum** the responses based on V, A, K.
- The **maximum sum** indicates the dominant learning style of the respondent.



Problem Statement: Literature Analysis 1

a. Effective learning through personalisation of learning objects

References	Main Topic	Actions	Findings	Limitations
Souabi et al. (2021)	• Emphasised the **need for recommendation systems** due to the **large amount of learning material**	• Proposed a recommendation systems with the **integration of machine learning algorithms.**	• Stressed the **importance of tailoring learning objects to align with students' learning styles and preferences**	• Did not specify any machine learning algorithm to use
Nabizadeh et al. (2020)	• Sequence of **learning objects recommendation** that accommodate to the users' time constraints while maximizing their scores	• Specify learning objects to a course • Perform Depth First Search to find all paths (LO sequences)	• Recommended learning objects **did help students to get better grades**	• Does not account to learning style and learning object preferences • Did not utilise machine learning models

Table 1: Journal references - 1

Problem Statement: Literature Analysis 2

b. Learning objects recommendation systems

References	Main Topic	Method Used	Findings	Limitations
Nafea et al. (2019)	• Learning object recommendation tool **based on learning style** and learning objects ratings.	• Collaborative Filtering (CF) • Content-Based Filtering (CBF) • Hybrid filtering (HF)	• Hybrid filtering is the best approach • Majority (95%) of the students were satisfied with the LO recommendations from the HF algorithm	• **Further experiments needed** using different personalisation strategies
Syed et al. (2017)	• Personalised learning object recommendation system architecture **based on learning object preferences**	• Hybrid filtering	• Suggested **technical measures for evaluating** recommendation systems, such as **accuracy** and performance	• **Did not train or evaluate any model** • Did not consider the aspect of learning styles
Imran and Abdullah (2010)	• Learning object recommendation tool **based on outstanding learners' ratings on the learning objects**	• Content-based filtering (CBF) • Peer-review mechanism • SVM to calculate objects similarities	• Proposed system has **better precision** as compared to an e-learning with the usual content-based recommender system	• **Does not reflect all learners' preferences** accurately.

Table 2: Journal references - 2

Problem Statement: Literature Analysis 3

c. Classification algorithms comparisons

Glossary:

- *Naive Bayes - NB*
- *Decision Tree - DT*
- *Support Vector Machine - SVM*
- *Connected Neural Network - CNN*
- *k-Nearest Neighbour - kNN*
- *Random Forest - RF*
- *Logistic Regression - LR*
- *eXtreme Gradient Boosting - XGB*

References	Journal Title	Models	Results	Limitation
Deng et al. (2023)	• Comparison of multiple machine learning algorithms for music genre classification	• NB, kNN, DT, RF, SVM, LR, CNN	• CNN and **kNN** has the highest accuracy (~90%) • Naive Bayes has the lowest accuracy of (~51%)	• CNN is computational expensive
Nazish et al. (2021)	• COVID-19 Lung Image Classification Based on Logistic Regression and Support Vector Machine	• SVM, LR	• **SVM** achieved the highest accuracy of 96% • **LR** records 92% accuracy	• Only compares 2 classification models
Santana et al. (2021)	• Classification Models for COVID-19 Test Prioritization in Brazil: Machine Learning Approach	• DT, RF, XGB, kNN, SVM, LR	• **DT, RF, XGB**, and **SVM** models have similar accuracy results (~89% accuracy). • Paper chose DT as the best due to its interpretability.	• Not specified
Kebonye (2021)	• Exploring the novel support points-based split method on a soil dataset	• 4 percentage ratios of 60/40, 70/30, **75/25** and 80/20.	• 70/30, 75/25 and 80/20 ratios have **similar** score	• Not specified

Table 3: Journal references - 3

Problem Statement: Summary

01. Finding the most suitable learning objects for effective learning is a challenge

- The enormity of the amount of learning objects has led students to have difficulties in determining the most suitable learning objects for them (Souabi et al., 2021).

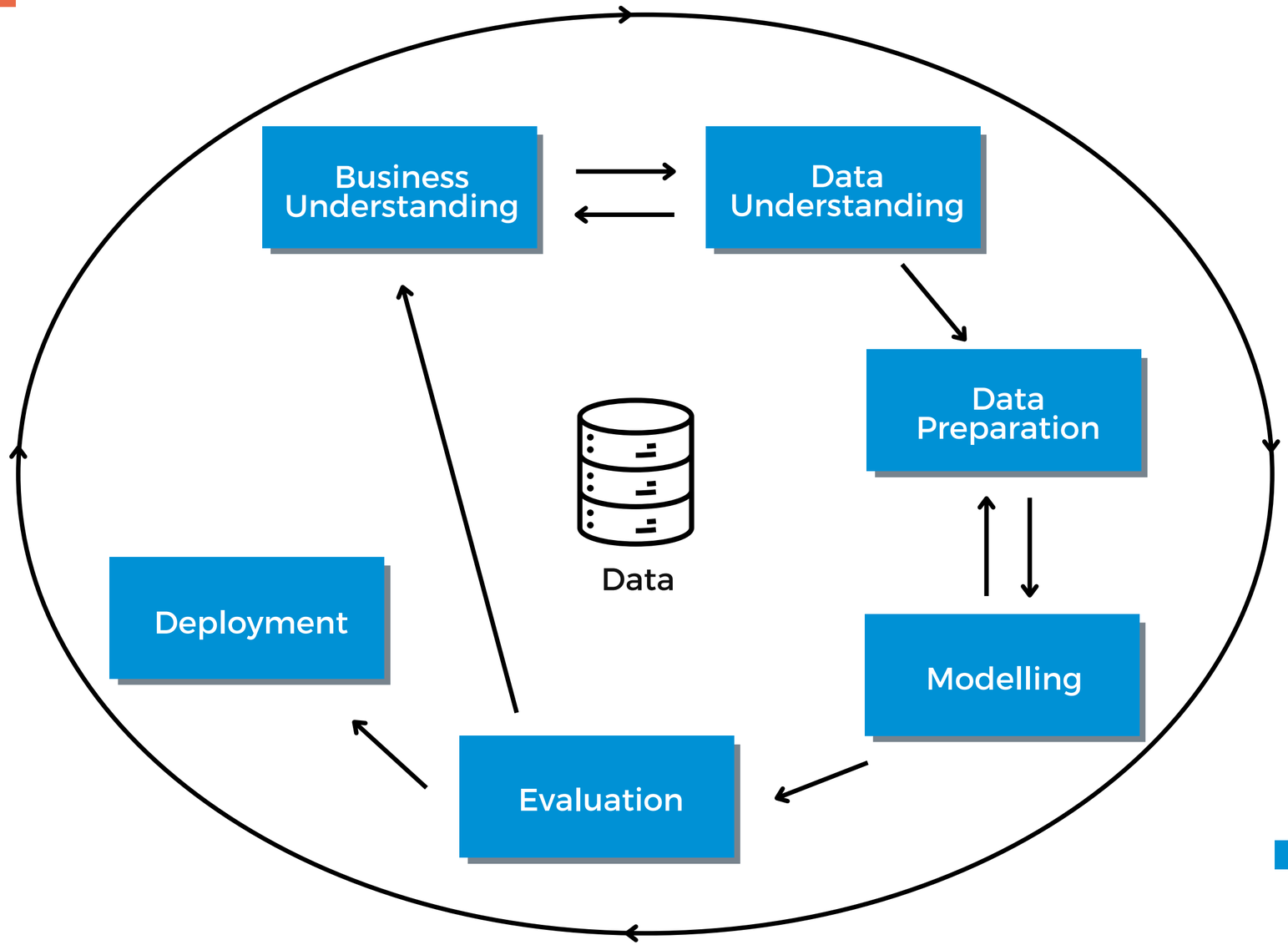
02. Limited classification models have been developed for learning objects based on learning style and learning object preferences

- Previous research only focuses on mainstream recommendation algorithms (Nafea et al., 2019; Syed et al., 2017; Imran and Abdullah, 2010).
- Past research does not take into account both the student's learning style (Imran and Abdullah, 2010) and learning object preferences (Nafea et al., 2019).

Objectives

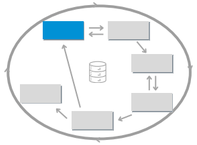
- To **develop** a learning object classification based on personalisation **model**
- To **evaluate** a learning object classification based on personalisation **model**
- To **develop** a functional data product **web application** which can provide learning objects recommendation

Data Science Methodology



CRISP-DM

DS Methodology: Business Understanding



a. Scope

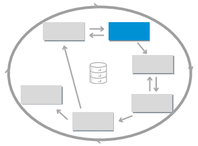
- Learning style i.e., VAK Model
- Learning objects

01. Understand the Domain

c. Define objectives

b. Study previous researches

DS Methodology: Data Understanding 1



02. Understand the Data

a. Obtained from a survey

- Distributed to students of Universiti Malaya from year 2021 - 2022

b. Data Size

- 1036 rows, 104 columns

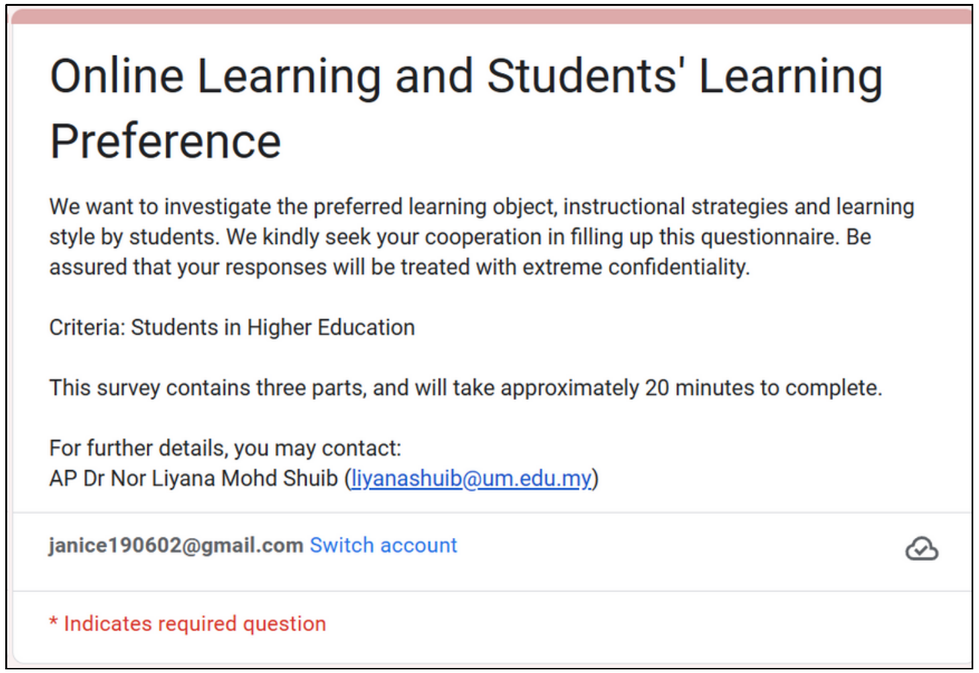
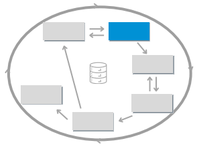


Figure 2: Survey form

DS Methodology: Data Understanding 2



03. Choose relevant columns

a. Use SAS Enterprise Miner

- Chi-square measure
  - To determine columns that have a link to the target variables

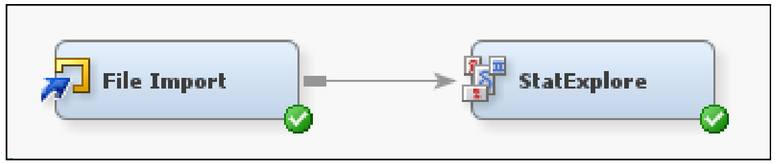
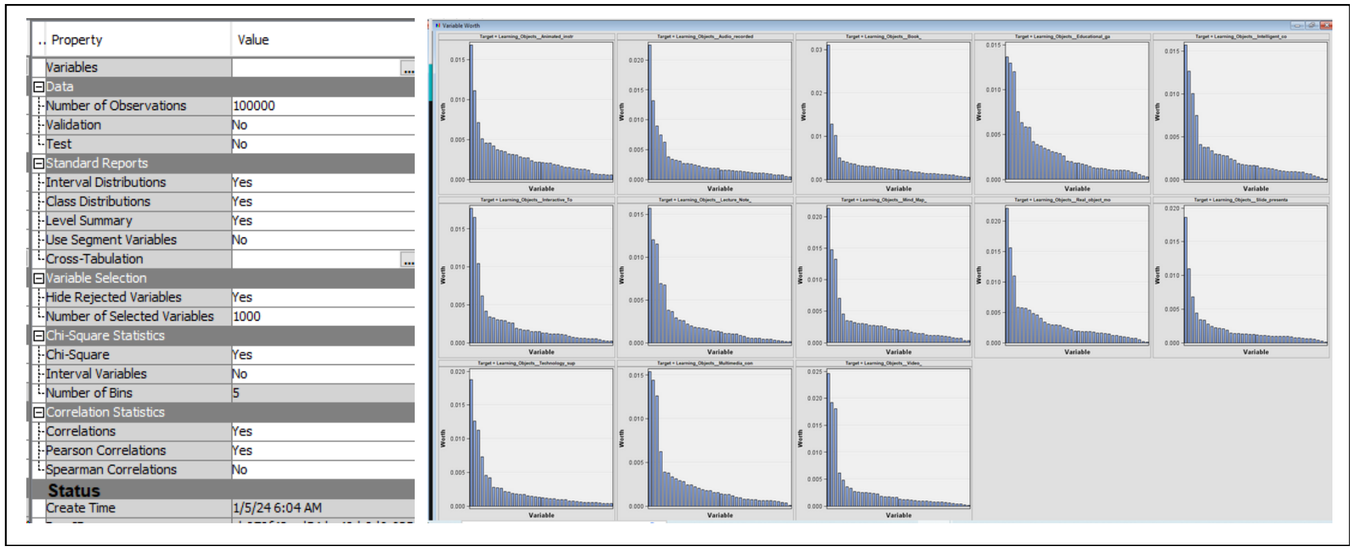


Figure 3: Nodes built in SAS

b. Columns identified

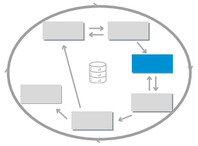
- Gender
- Level of study
- Household income
- Preferred learning mode
- Preferred communication platform
- Learning objects preferences
- VAK learning style questions



Property	Value
Variables	
Data	
Number of Observations	100000
Validation	No
Test	No
Standard Reports	
Interval Distributions	Yes
Class Distributions	Yes
Level Summary	Yes
Use Segment Variables	No
Cross-Tabulation	
Variable Selection	
Hide Rejected Variables	Yes
Number of Selected Variables	1000
Chi-Square Statistics	
Chi-Square	Yes
Interval Variables	No
Number of Bins	5
Correlation Statistics	
Correlations	Yes
Pearson Correlations	Yes
Spearman Correlations	No
Status	
Create Time	1/5/24 6:04 AM

Figure 4: StatsExplore parameters & output

DS Methodology: Data Preparation 1



04.
Check for null values

Check for null values

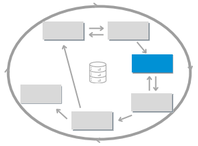
```
# Checking for null values in the dataframe
null_counts = df.isnull().sum()

# Displaying columns with null values
columns_with_null = null_counts[null_counts > 0]
if columns_with_null.empty:
    print("No null values found in the DataFrame.")
else:
    print("Columns with null values:")
    print(columns_with_null)
```

```
No null values found in the DataFrame.
```

→ **No null values found**

Figure 5: Code snippet of null value checks

DS Methodology: Data Preparation 2



05. Perform data Standardisation

a. Fix all data inconsistencies

- Used mappings by defining dictionary and replace the values respectively

```python
# Mapping different representations to a standardized value
standardized_values = {
    'PhD': 'Postgraduate',
    'Master': 'Postgraduate'
}

# Replace values in the 'Institutions' column with the standardized values
df['Level of Study'] = df['Level of Study'].replace(standardized_values)
```

```python
# Mapping different representations to a standardized value
standardized_values = {
    "Less than RM 3000": 'Less than RM 4,849',
    "RM 3001 - 10 000": 'RM 4,850 – RM10,959',
    "RM 10 001 - 25 000": 'More than RM10,960'
}

# Replace values in the 'Institutions' column with the standardized values
df['Household Income'] = df['Household Income'].replace(standardized_values)
```

Figure 6: Code snippet of **mappings** values by defining dictionary

```python
distinct_values = df['Level of Study'].unique()

# Print all distinct values
print("Distinct values in 'Level of Study':")
for value in distinct_values:
    print(value)
```

```
Distinct values in 'Level of Study':
Postgraduate
Undergraduate
Master
PhD
Certificate/Diploma
```

```python
distinct_values = df['Household Income'].unique()

# Print all distinct values
for value in distinct_values:
    print(value)
```

```
RM 3001 - 10 000
RM 10 001 - 25 000
Less than RM 3000
RM 4,850 – RM10,959
Less than RM 4,849
More than RM10,960
```

Figure 7: **Original** values in the 'Level of Study' & 'Household income'

After standardisation

```python
distinct_values = df['Level of Study'].unique()

# Print all distinct values
print("Distinct values in 'Level of Study':")
for value in distinct_values:
    print(value)
```

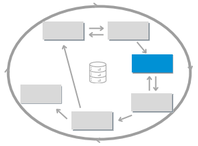
```
Distinct values in 'Level of Study':
Postgraduate
Undergraduate
Certificate/Diploma
```

```python
distinct_values = df['Household Income'].unique()

# Print all distinct values
for value in distinct_values:
    print(value)
```

```
RM 4,850 – RM10,959
More than RM10,960
Less than RM 4,849
```

Figure 8: **Standardised** values in the 'Level of Study' & 'Household income'

DS Methodology: Data Preparation 3



06. Determine the dominant learning style

a. Define answer options to their respective learning style.

```
# Define answers options
visual_keywords = ["Read the instructions",
                   "Look at a map",
                   "Follow a recipe",
```

```
auditory_keywords = ["Listen to or ask for an explaination",
                     "Ask for spoken directions",
                     "Call a friend for explaination",
```

```
kinesthetic_keywords = ["Have a go and learn by \"trial and error\"",
                        "Follow my nose or maybe use a compass",
                        "Follow my instinct, tasting as I cook",
```

Figure 9: Define answers to their respective dominant learning style; V, A, K

b. Calculate the sum of V, A, K options selected respectively.

```
# Iterate the columns (each column is a question)
for column in responses_df.columns:
    response = row[column].lower()

    # Compare keywords with the response
    # Count the number of visual/auditory/kinesthetic answers chosen
    for keyword in visual_keywords:
        if keyword.lower() in response:
            visual_count += 1

    for keyword in auditory_keywords:
        if keyword.lower() in response:
            auditory_count += 1

    for keyword in kinesthetic_keywords:
        if keyword.lower() in response:
            kinesthetic_count += 1
```

Figure 10: Summing the options chosen

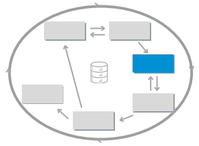
c. Get the maximum sum and append it as a new column

- Max sum represent the dominant learning style

```
dominant_preference = max(preferences, key=preferences.get)
dominant_preferences.append(dominant_preference)
```

Figure 11: Get the learning style with the maximum sum

DS Methodology: Data Preparation 4.0



07.
Perform EDA

a. Created 2 dashboard

- Dataset General Distribution
- Learning Objects Preferences

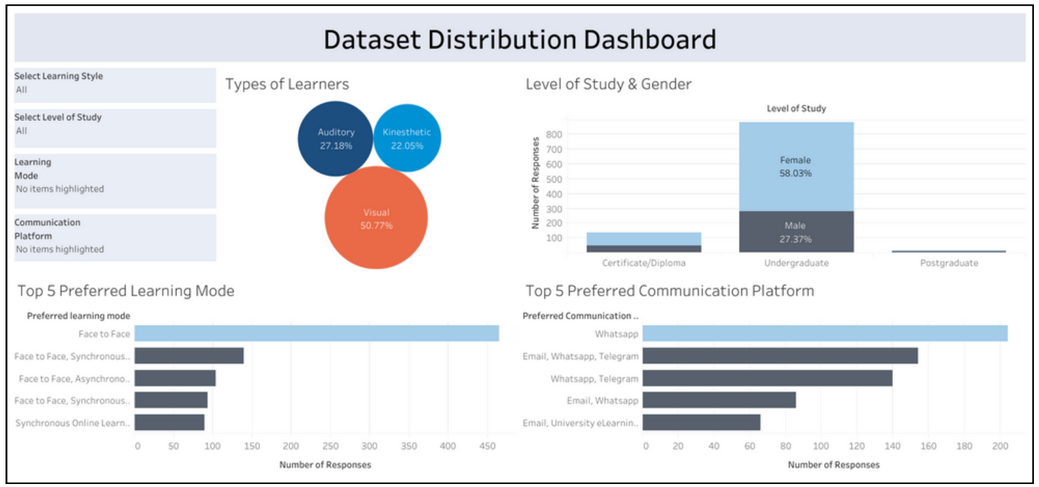


Figure 12: General distribution dashboard

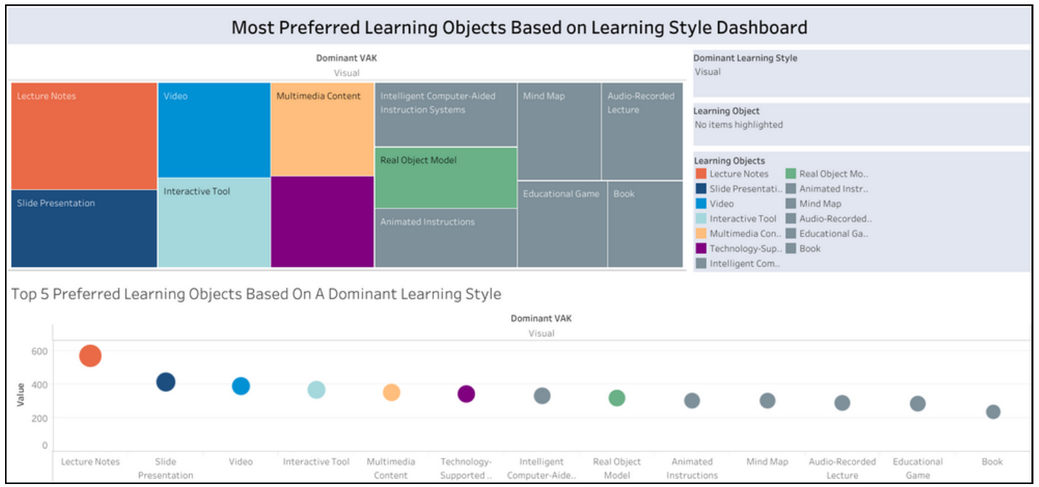
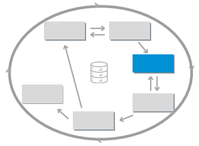


Figure 13: Learning objects preferences based on learning style

Link to dashboard: General Distribution Dashboard

Link to dashboard: Learning Objects Preferences

DS Methodology: Data Preparation 4.1



b. Level of study & Gender

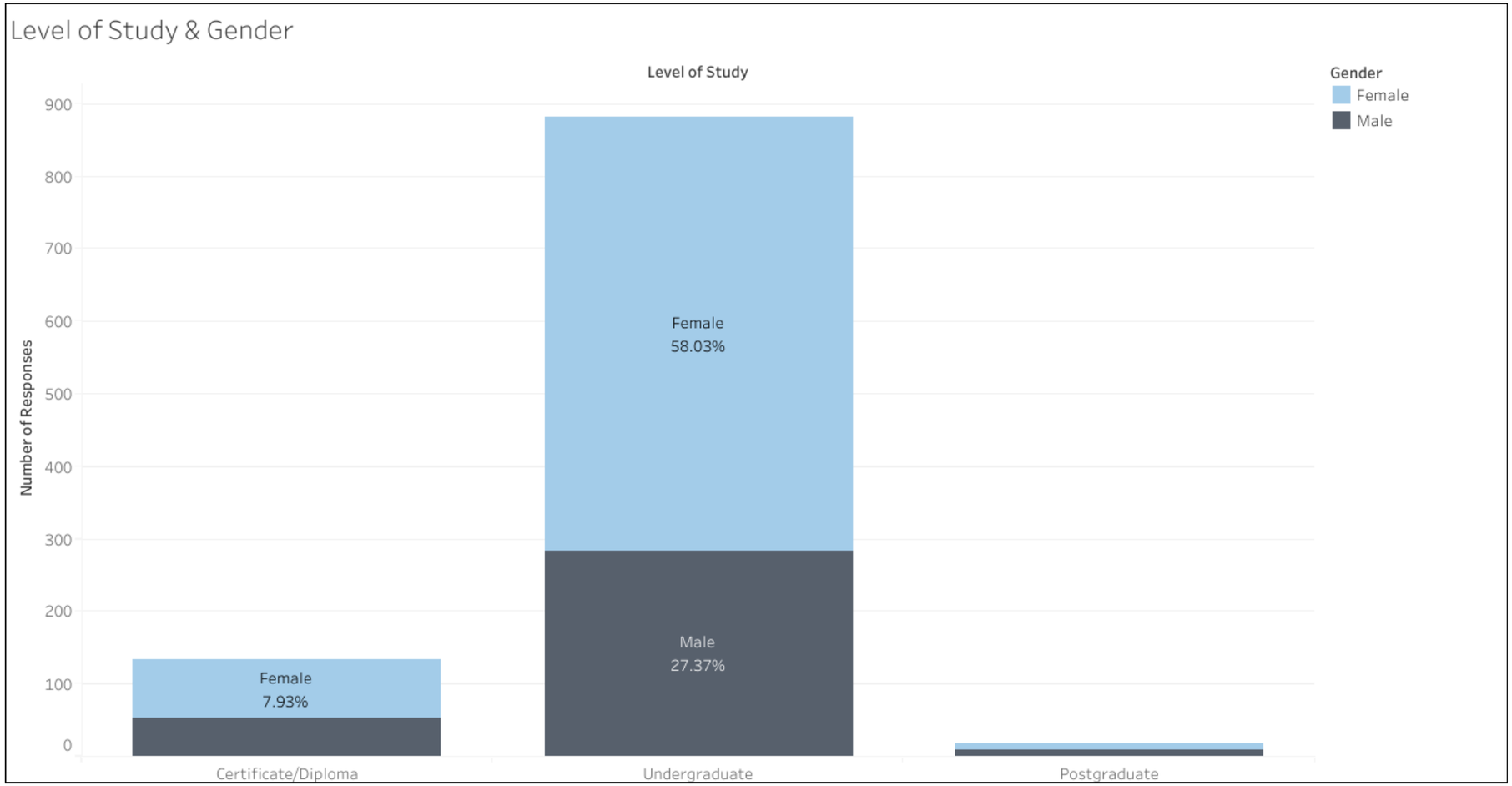
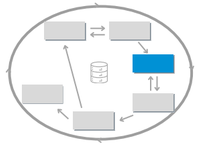


Figure 14: Level of study and gender of respondents

Link to dashboard: General Distribution Dashboard

DS Methodology: Data Preparation 4.2



c. Dominant learning style distribution

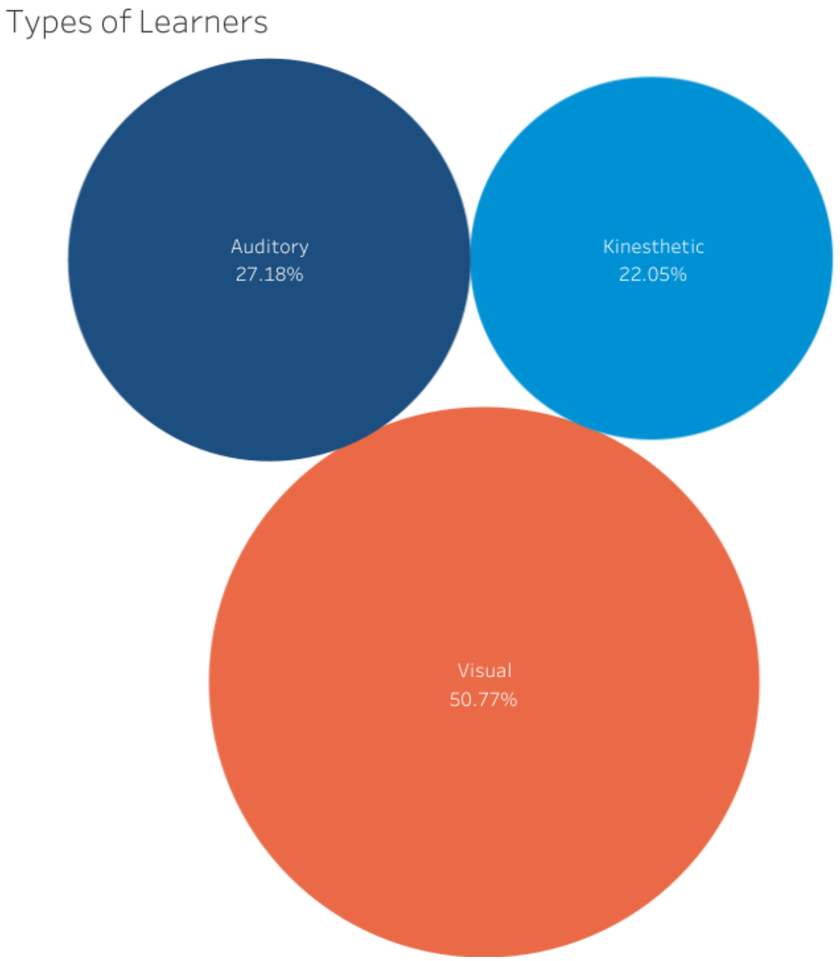
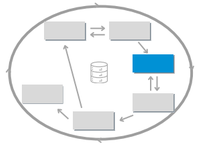


Figure 15: Distribution of the types of learner

Link to dashboard: General Distribution Dashboard

DS Methodology: Data Preparation 4.3



d. Preferred learning mode

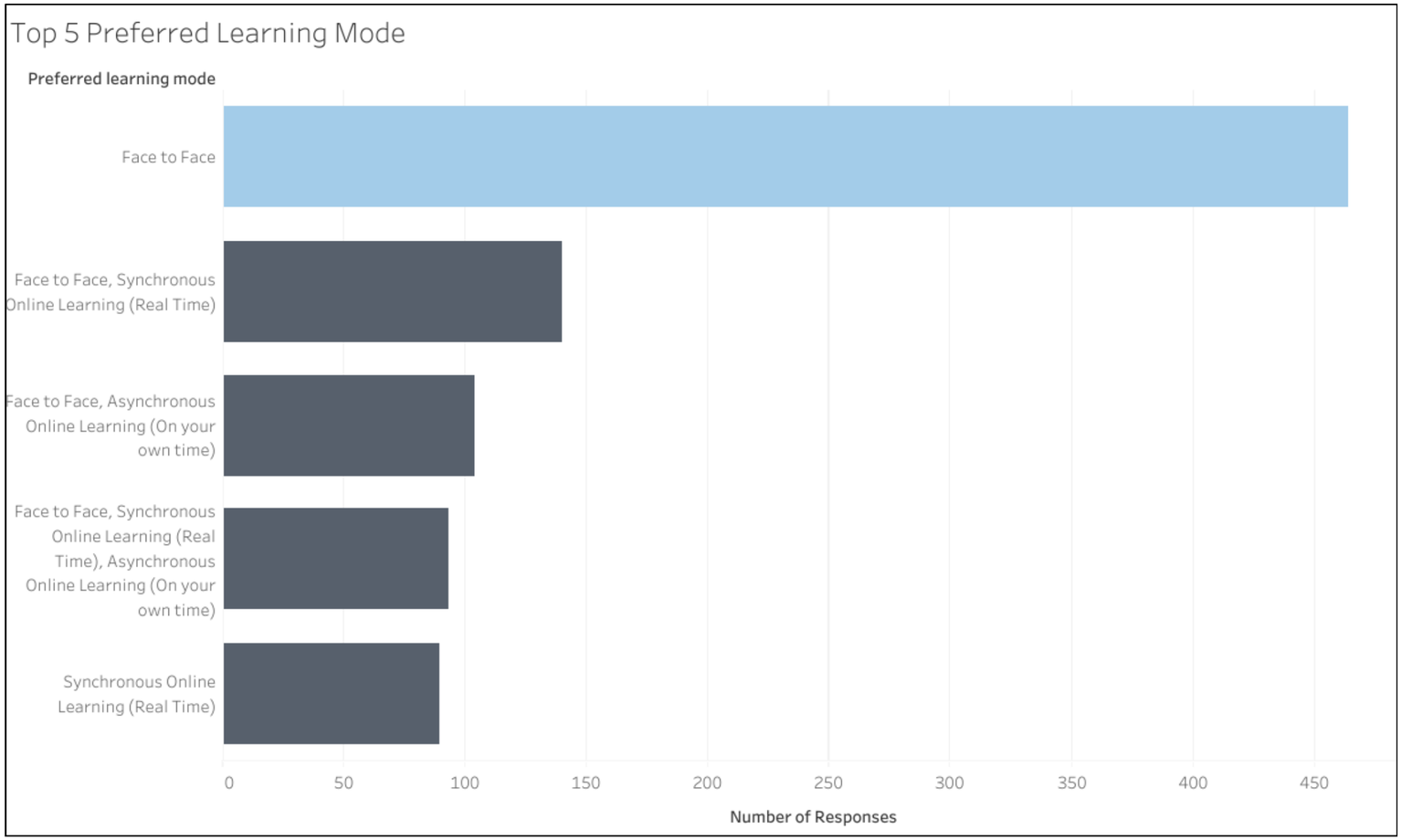
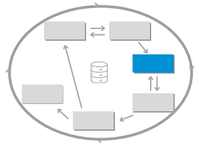


Figure 16: Preferred learning mode by visual, auditory and kinesthetic learners

Link to dashboard: General Distribution Dashboard

DS Methodology: Data Preparation 4.4



e. Preferred communication platform

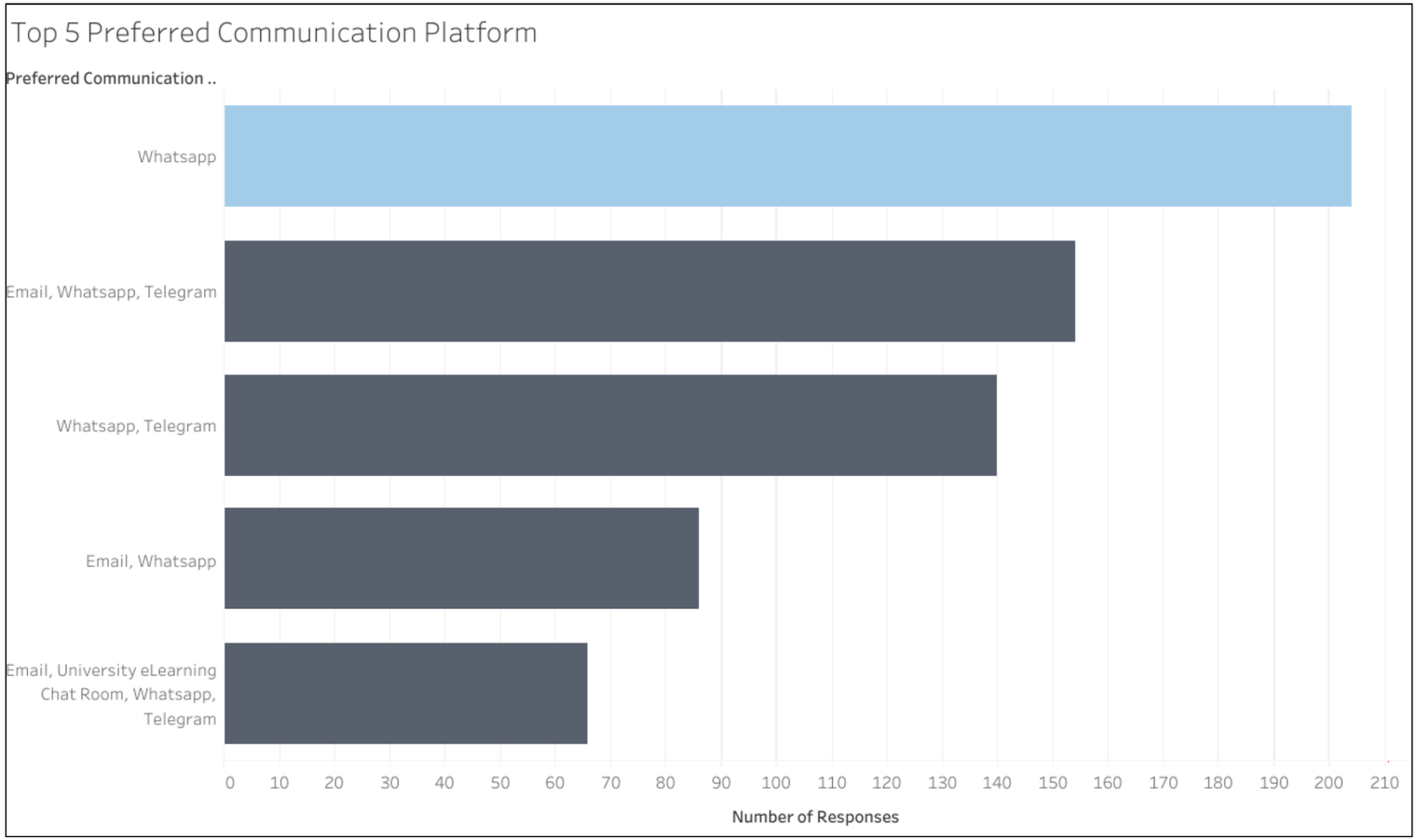
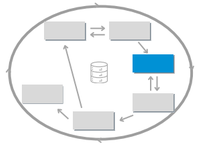


Figure 17: Top 5 preferred communication platform by visual, auditory and kinesthetic learners

Link to dashboard: General Distribution Dashboard

DS Methodology: Data Preparation 4.5



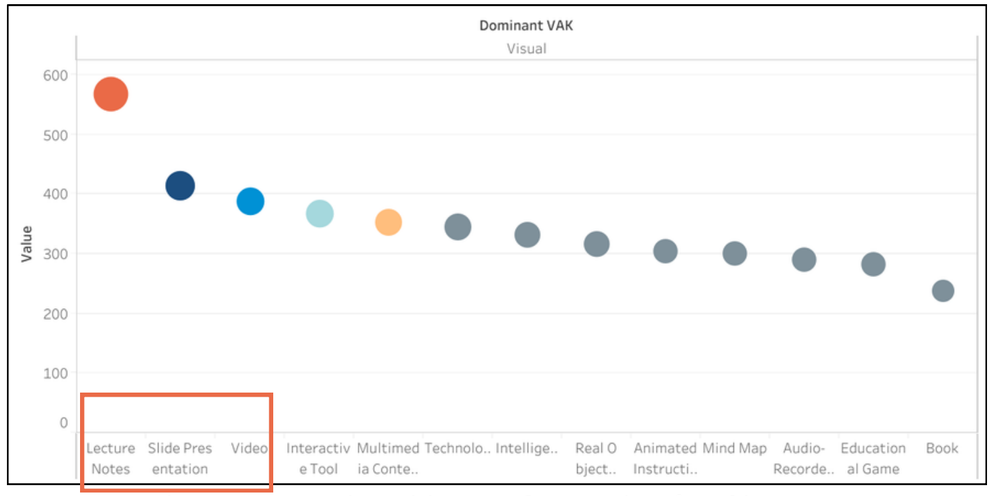


Figure 18: Learning object preference by **visual** learners

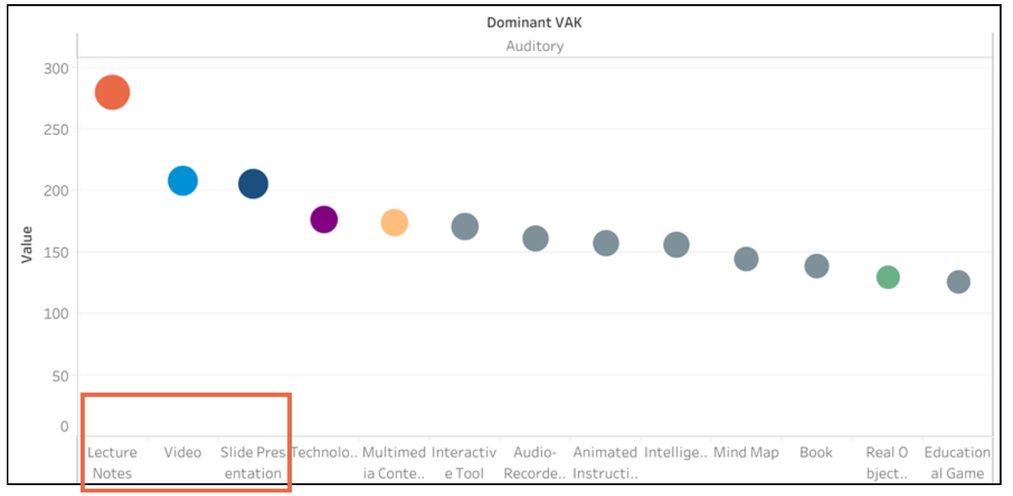


Figure 19: Learning object preference by **auditory** learners

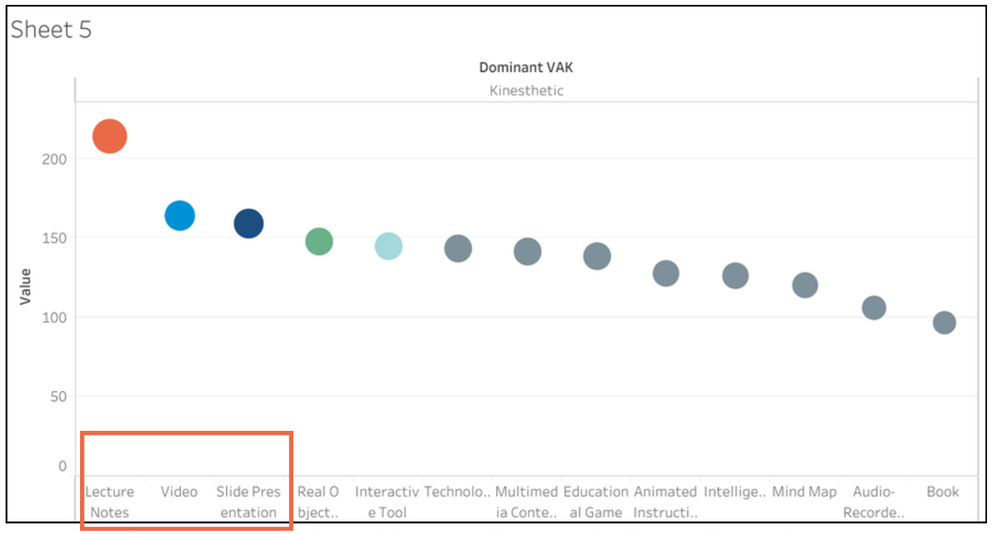


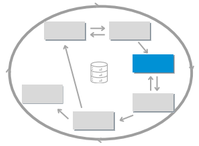
Figure 20: Learning object preference by **kinesthetic** learners

f. Learning objects preferences

- All 3 learning styles have **similar top 3** preferences.
  - Lecture Notes
  - Slides Presentation
  - Video

Link to dashboard: Learning Objects Preferences

DS Methodology: Data Preparation 4.6



g. Visual learners' learning objects preferences

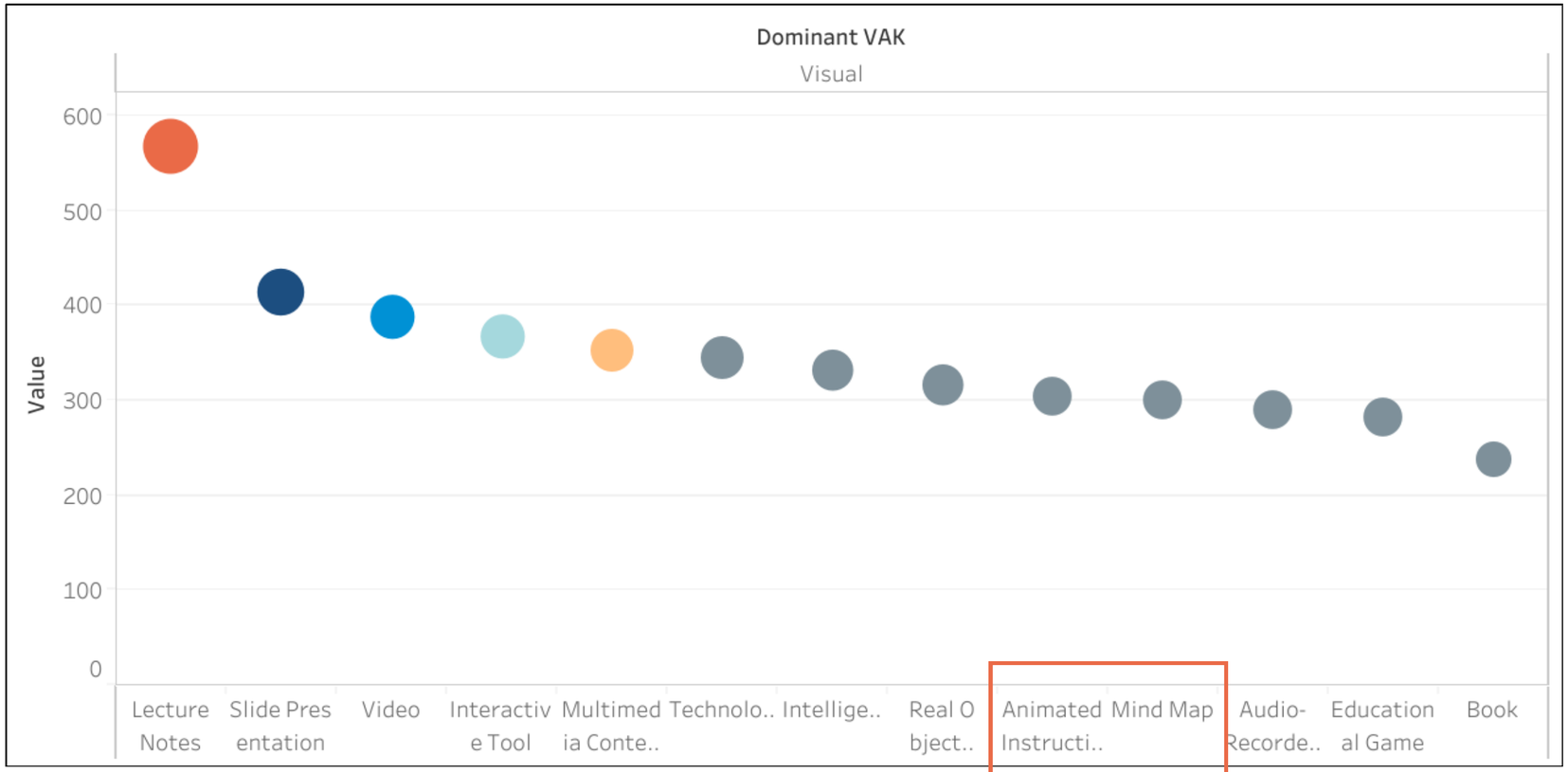
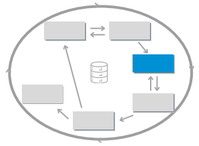


Figure 21: Learning object preference by **visual** learners

- Visual materials like **animated instructional** and **mind maps** are ***less preferred*** for visual learners.

Link to dashboard: Learning Objects Preferences

DS Methodology: Data Preparation 4.7



h. Auditory learners' learning objects preferences

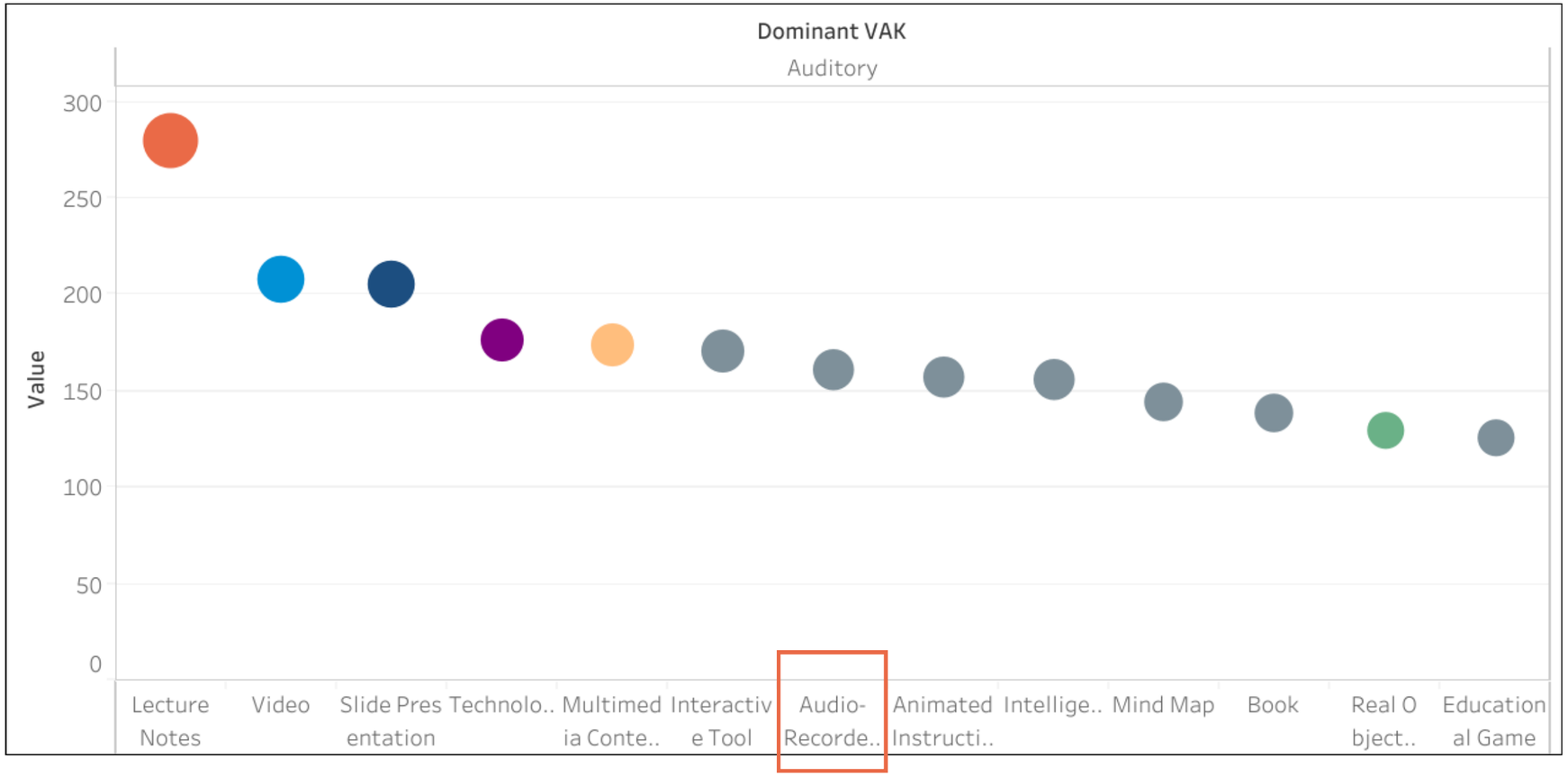
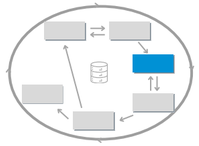


Figure 22: Learning object preference by **auditory** learners

- Auditory materials like **audio-recorded lectures** are ***less preferred*** for auditory learners.

Link to dashboard: Learning Objects Preferences

DS Methodology: Data Preparation 4.8



i. Kinesthetic learners' learning objects preferences

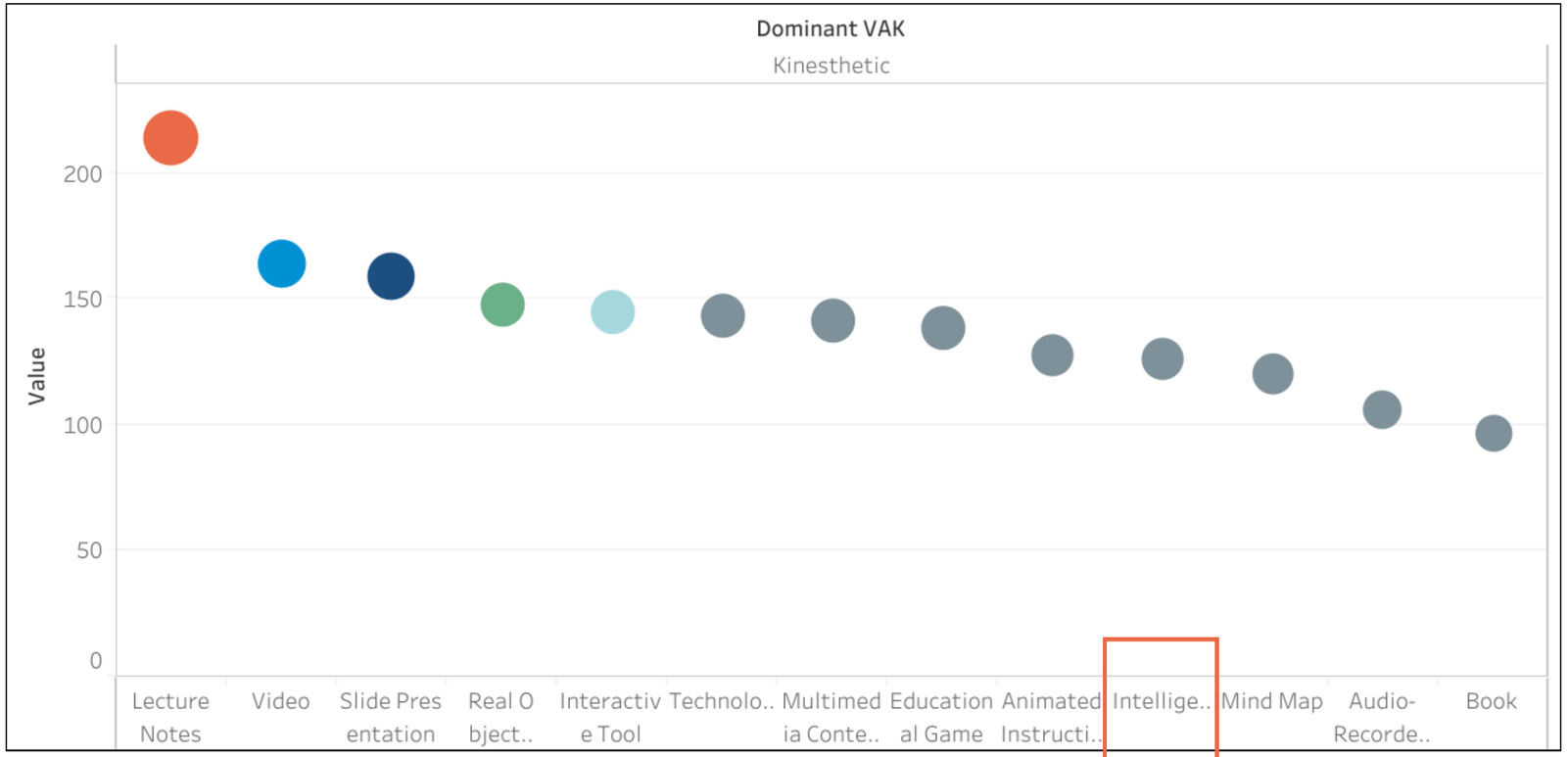
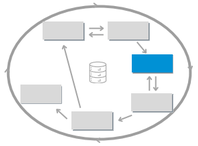


Figure 23: Learning object preference by **kinesthetic** learners

- Hands-on learning objects such as **intelligent computer-aided** are ***less preferred*** for kinesthetic learners.

Link to dashboard: Learning Objects Preferences

DS Methodology: Data Preparation 5



08. Data Exploding → **For multiple option questions**

a. Split options by comma.

```
# Splitting column by ', ' with multiple options
df['Preferred learning mode'] = df['Preferred learning mode'].str.split(', ')
df['Preferred Communication Platform'] = df['Preferred Communication Platform'].str.split(', ')
```

Figure 24: Code snippet of splitting the answers from the column with multiple options

b. Perform 'explode'.

```
# Exploding the columns to separate rows for each value
df = df.explode('Preferred learning mode').reset_index(drop=True)
df = df.explode('Preferred Communication Platform').reset_index(drop=True)
```

Figure 25: Code snippet of exploding the answers from the column with multiple options

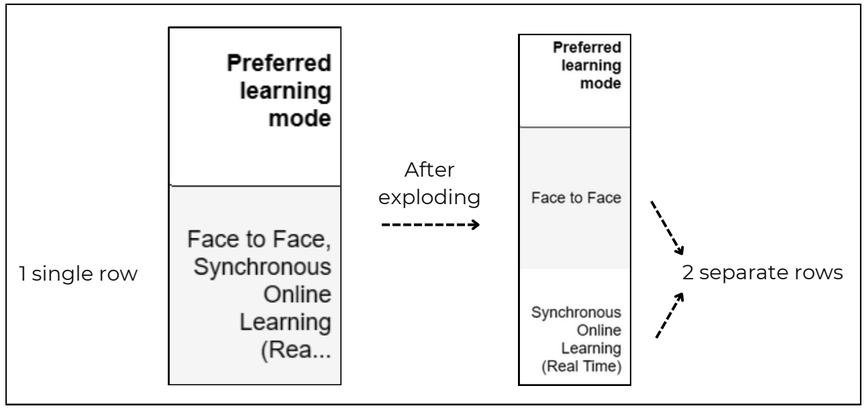
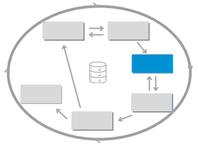


Figure 26: Before and after performing the 'explode'

DS Methodology: Data Preparation 6



09. Data Encoding

a. Ordinal encoding

- Perform for columns that have an order or require custom mappings
  - Gender
  - Level of Study
  - Household income
  - Learning objects preferences
  - Dominant learning style

```
# Define your custom mapping
custom_mapping = {
    "Visual": 1,
    "Auditory": 2,
    "Kinesthetic": 3
}

# Perform ordinal encoding using the defined mapping
for column in domvak_df.columns:
    df[column] = df[column].map(custom_mapping)
```

Figure 27: Sample of performing ordinal encoding

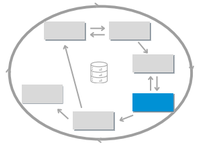
b. One-hot encoding

- Perform for the rest of the columns

```
# Encode categorical variables (one-hot encoding)
df = pd.get_dummies(df, columns=qcolumns_df.columns, prefix=qcolumns_df.columns)
```

Figure 28: Performed one-hot encoding on the VAK questions columns

DS Methodology: Modelling 1



10. Define target variables

a. Target variable are the learning objects

```
# # Target variable: Learning Objects Preference
target = df[[
    'Learning Objects [Slide presentation]',
    'Learning Objects [Book]',
    'Learning Objects [Lecture Note]',
    'Learning Objects [Educational game]',
    'Learning Objects [Video]',
    'Learning Objects [Audio-recorded lecture]',
    'Learning Objects [Animated instruction]',
    'Learning Objects [Real object model]',
    'Learning Objects [Mind Map]',
    'Learning Objects [Multimedia content]',
    'Learning Objects [Interactive Tool]',
    'Learning Objects [Technology-supported learning include computer-based training systems]',
    'Learning Objects [Intelligent computer-aided instruction systems]'
]]
```

Figure 29: Code snippet on defining target variables

---

11. Split test and train data

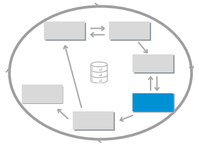
a. Split training & test sets

- test 25%, train 75%

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df.drop(target.columns, axis=1), target, test_size=0.25, random_state=42)
```

Figure 30: Code snippet of splitting data into train & test

DS Methodology: Modelling 2



12. Train, fit classification models

a. Total of 6 models

- Support Vector Machine (SVM)
- Random Forest
- Decision Tree
- eXtreme Gradient Boosting (XGB)
- K-Nearest Neighbour (kNN)
- Logistic Regression

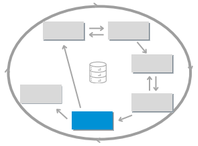
b. Utilised GridSearchCV()

- To find the best parameters for each model

```
# Instantiate GridSearchCV
grid_search = GridSearchCV(SVC(random_state=42), param_grid, cv=5, scoring='accuracy')

# Fit the grid search to the data for the current learning object
grid_search.fit(X_train, y_train[col])

# Get the best parameters and best estimator for the current learning object
best_params = grid_search.best_params_
best_estimator = grid_search.best_estimator_
```

Figure 31: Sample of implementing GridSearchCV in training, fitting and finding the best parameters of the model

DS Methodology: Evaluation



13. Evaluate the models

a. Accuracy score

- Represents a proportion of correct predictions made by a model

b. Classification report

- Macro-averaging: All classes are equally important

```python
# Initialize a dictionary to store accuracy scores
accuracy_scores = {}

# Loop through each column and calculate accuracy score
for col in y_test.columns:
    accuracy = accuracy_score(y_test[col], y_pred[col])
    accuracy_scores[col] = accuracy
    print(f"Accuracy for {col}: {accuracy}")

# Overall accuracy score
overall_accuracy = accuracy_score(y_test.values.flatten(), y_pred.values.flatten())
print(f"\nOverall Accuracy: {overall_accuracy}")
```

```
Accuracy for Learning Objects [Slide presentation]: 0.9547413793103449
Accuracy for Learning Objects [Book]: 0.9665948275862069
Accuracy for Learning Objects [Lecture Note]: 0.9633620689655172
Accuracy for Learning Objects [Educational game]: 0.9601293103448276
Accuracy for Learning Objects [Video]: 0.9536637931034483
Accuracy for Learning Objects [Audio-recorded lecture]: 0.9612068965517241
Accuracy for Learning Objects [Animated instruction]: 0.9612068965517241
Accuracy for Learning Objects [Real object model]: 0.9525862068965517
Accuracy for Learning Objects [Mind Map]: 0.9622844827586207
Accuracy for Learning Objects [Multimedia content]: 0.959051724137931
Accuracy for Learning Objects [Interactive Tool]: 0.9558189655172413
Accuracy for Learning Objects [Technology-supported learning include computer-based training systems]: 0.9644396551724138
Accuracy for Learning Objects [Intelligent computer-aided instruction systems]: 0.9665948275862069

Overall Accuracy: 0.9601293103448276
```

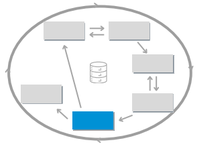
Figure 32: A sample of performing accuracy score on a model

```python
# Make predictions on the test set
y_pred = pd.DataFrame({col: classifier.predict(X_test) for col, classifier in svm_model.items()})

# Classification Report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.95      0.96       492
           1       1.00      0.89      0.94       281
           2       0.96      0.99      0.97       654
           3       0.98      0.91      0.94       343
           4       0.96      0.95      0.95       477
           5       0.97      0.91      0.94       327
           6       0.98      0.92      0.95       365
           7       0.98      0.90      0.94       370
           8       0.99      0.91      0.95       350
           9       0.98      0.93      0.95       428
          10       0.97      0.93      0.95       416
          11       0.99      0.93      0.96       426
          12       0.99      0.93      0.96       392

   micro avg       0.98      0.93      0.95      5321
   macro avg       0.98      0.93      0.95      5321
weighted avg       0.98      0.93      0.95      5321
 samples avg       0.86      0.84      0.84      5321
```

Figure 33: A sample of performing classification report on a model

DS Methodology: Evaluation Results



c. Model evaluation results

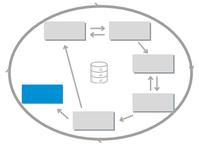
- Support Vector Machine (SVM) has the highest accuracy. Thus, it is the best model.

Model	Accuracy Score	Precision	Recall	F1-score
SVM	**0.9601**	**0.98**	**0.93**	**0.95**
Random Forest	0.9598	0.98	0.92	0.95
kNN	0.9569	0.96	0.94	0.95
XGB	0.9441	0.96	0.91	0.93
Decision Tree	0.9382	0.92	0.93	0.93
Logistic Regression	0.6642	0.62	0.52	0.55

⇢ Best model

Table 4: Accuracy score and classification report results of each model

DS Methodology: Deployment



Streamlit

14. Deploy the data product

a. Deployed a website, *Smart Learn*

- Utilised Streamlit

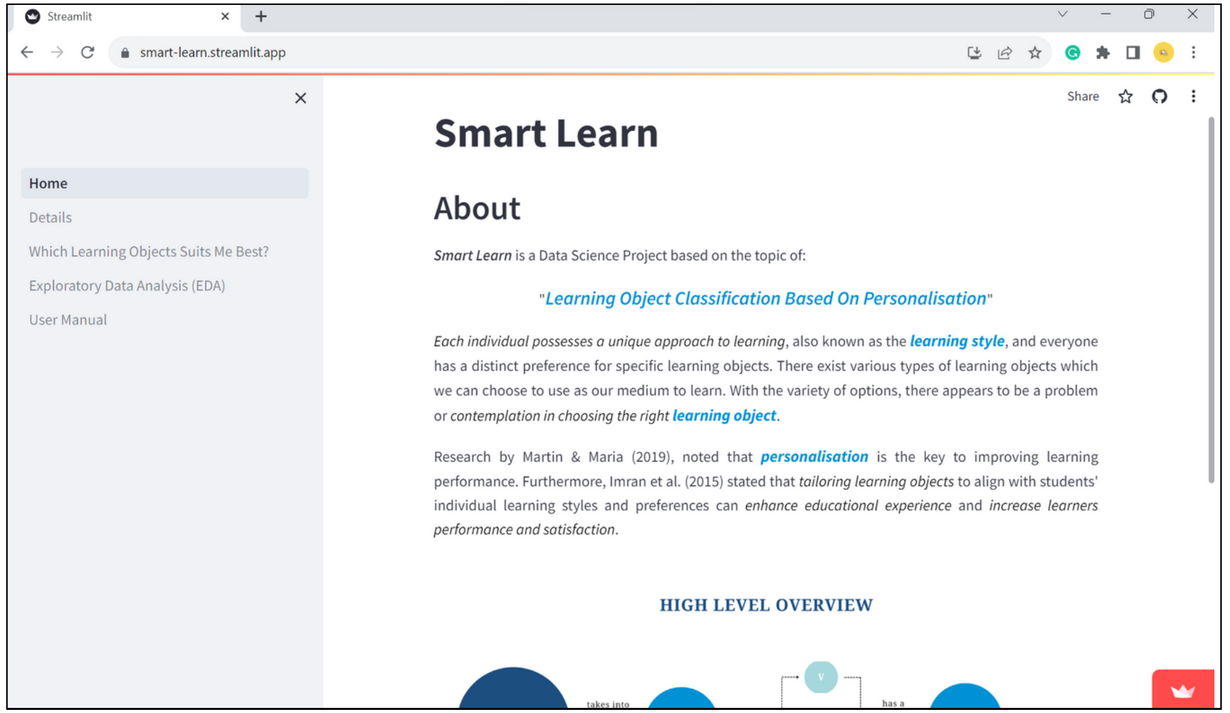
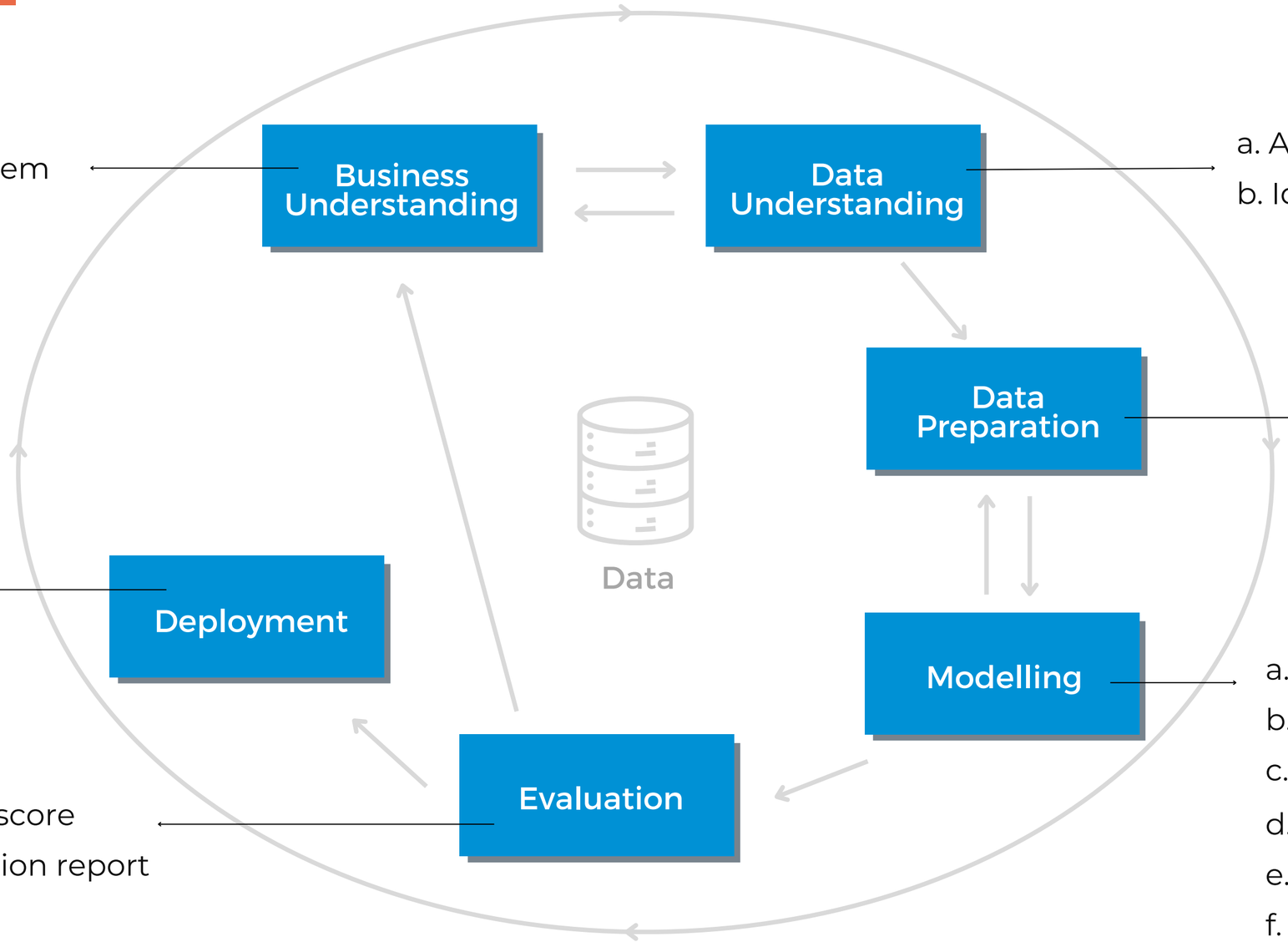


Figure 34: Smart Learn ‘s homepage

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

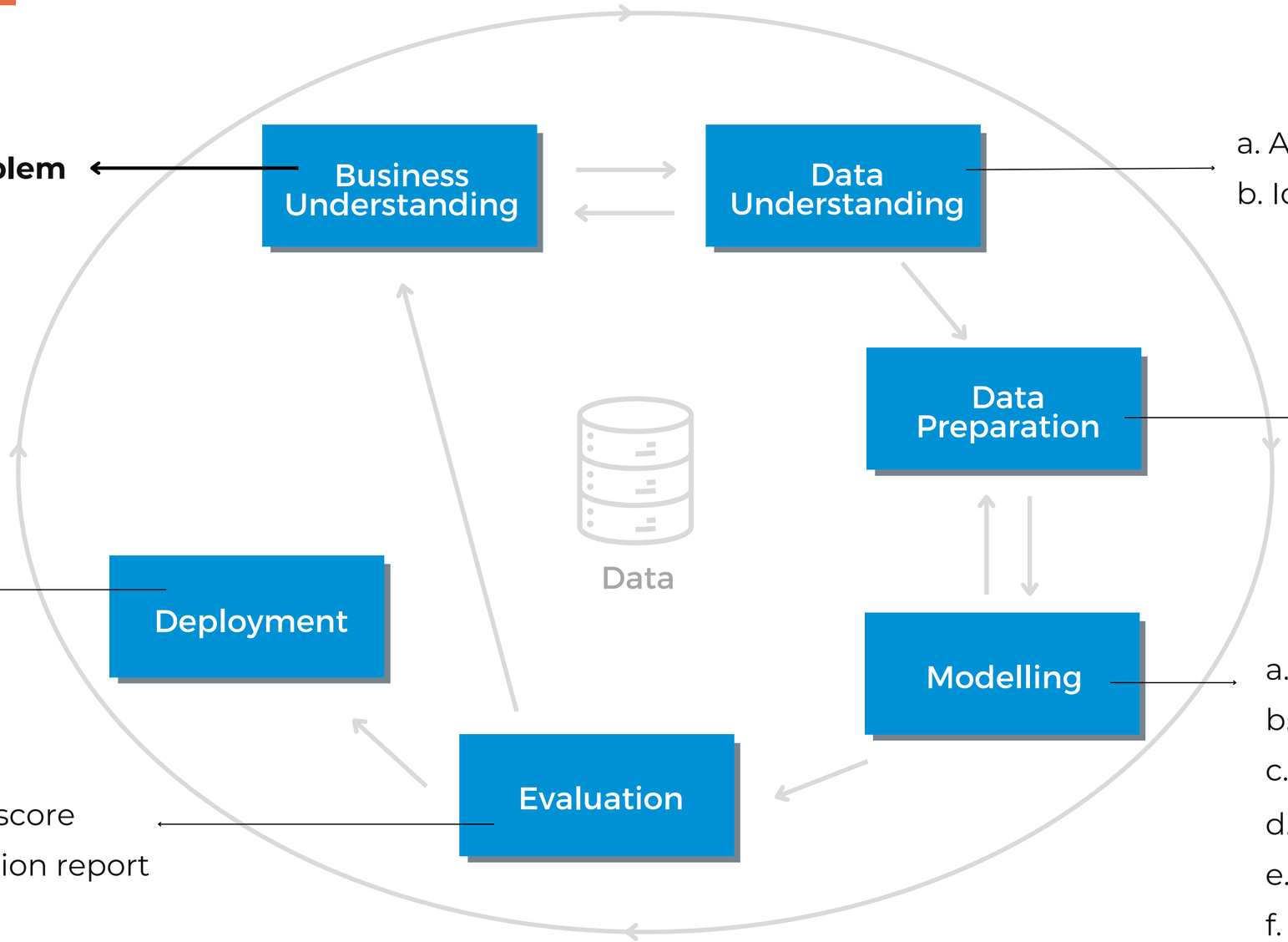
a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Accuracy score
b. Classification report

a. Web application

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

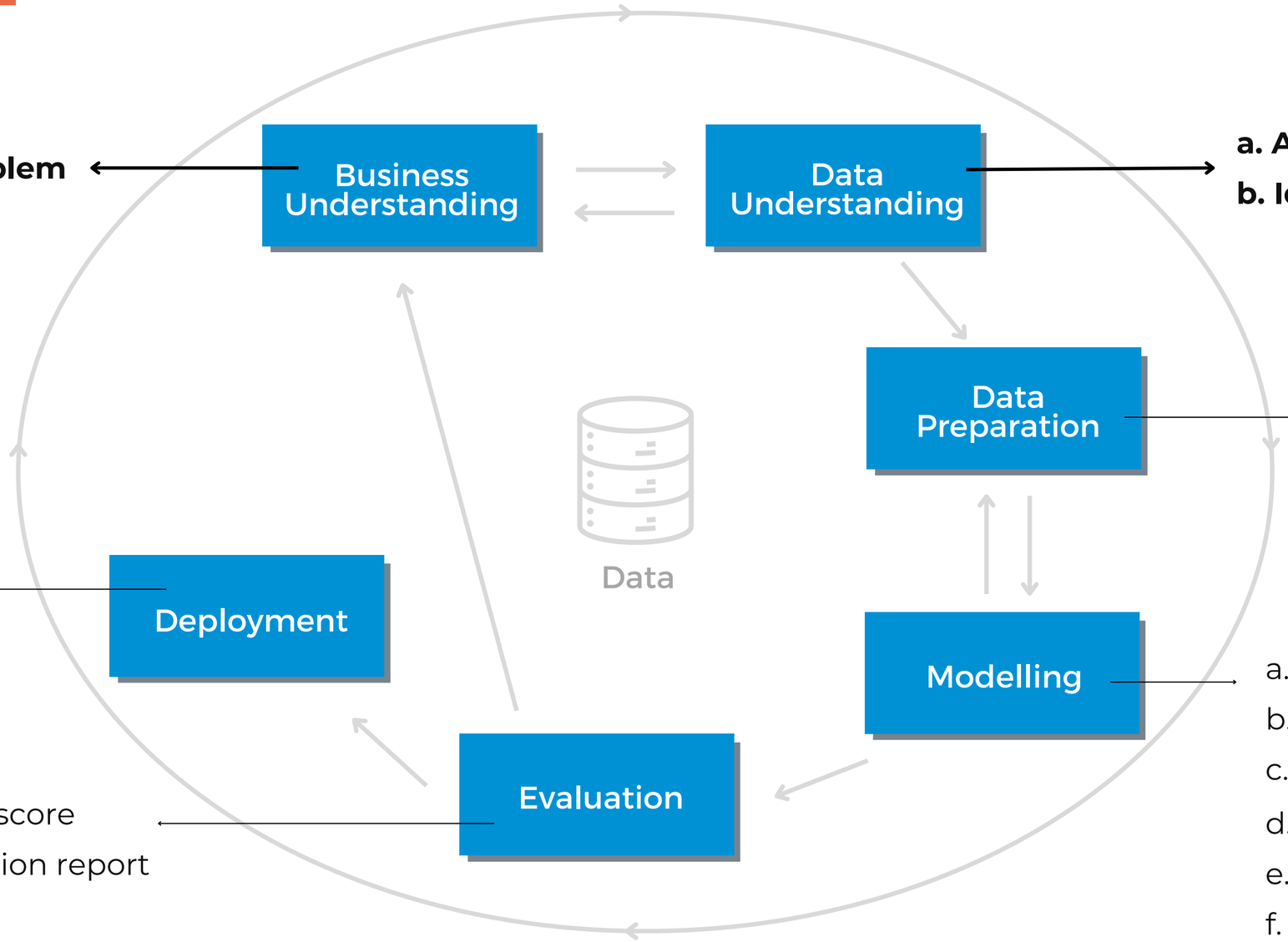
a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Accuracy score
b. Classification report

a. Web application

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

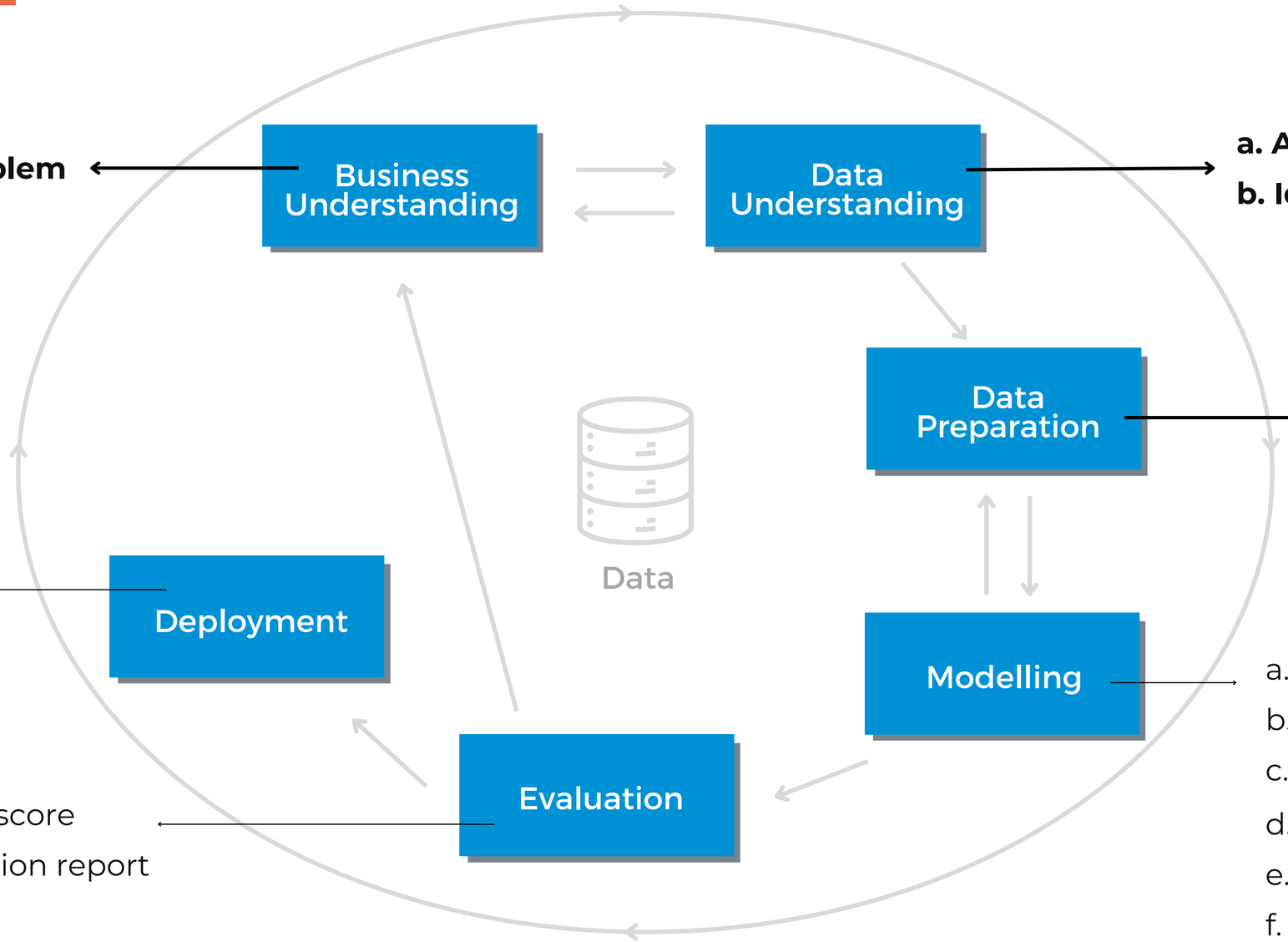
a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Web application

a. Accuracy score
b. Classification report

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

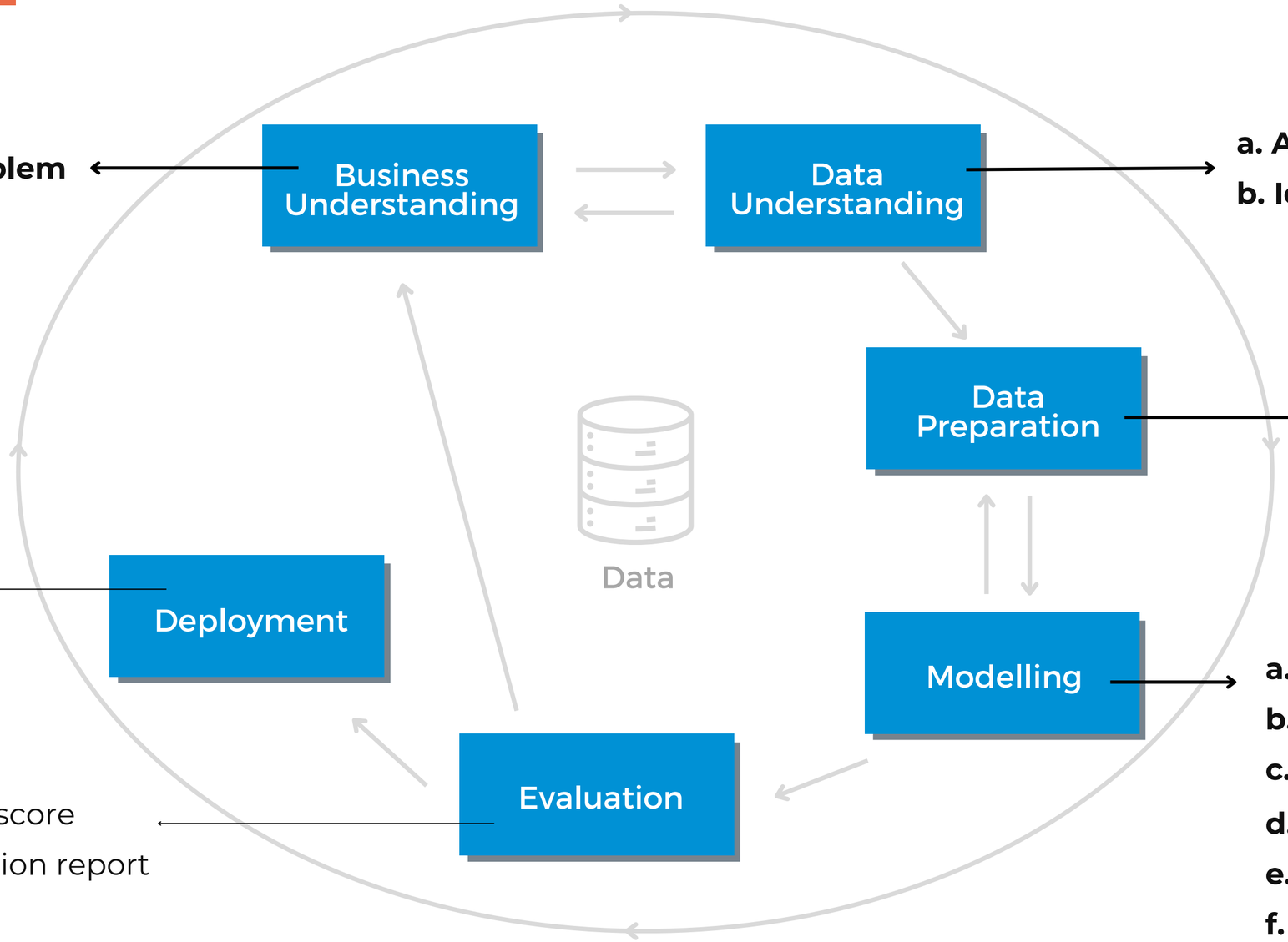
a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Web application

a. Accuracy score
b. Classification report

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

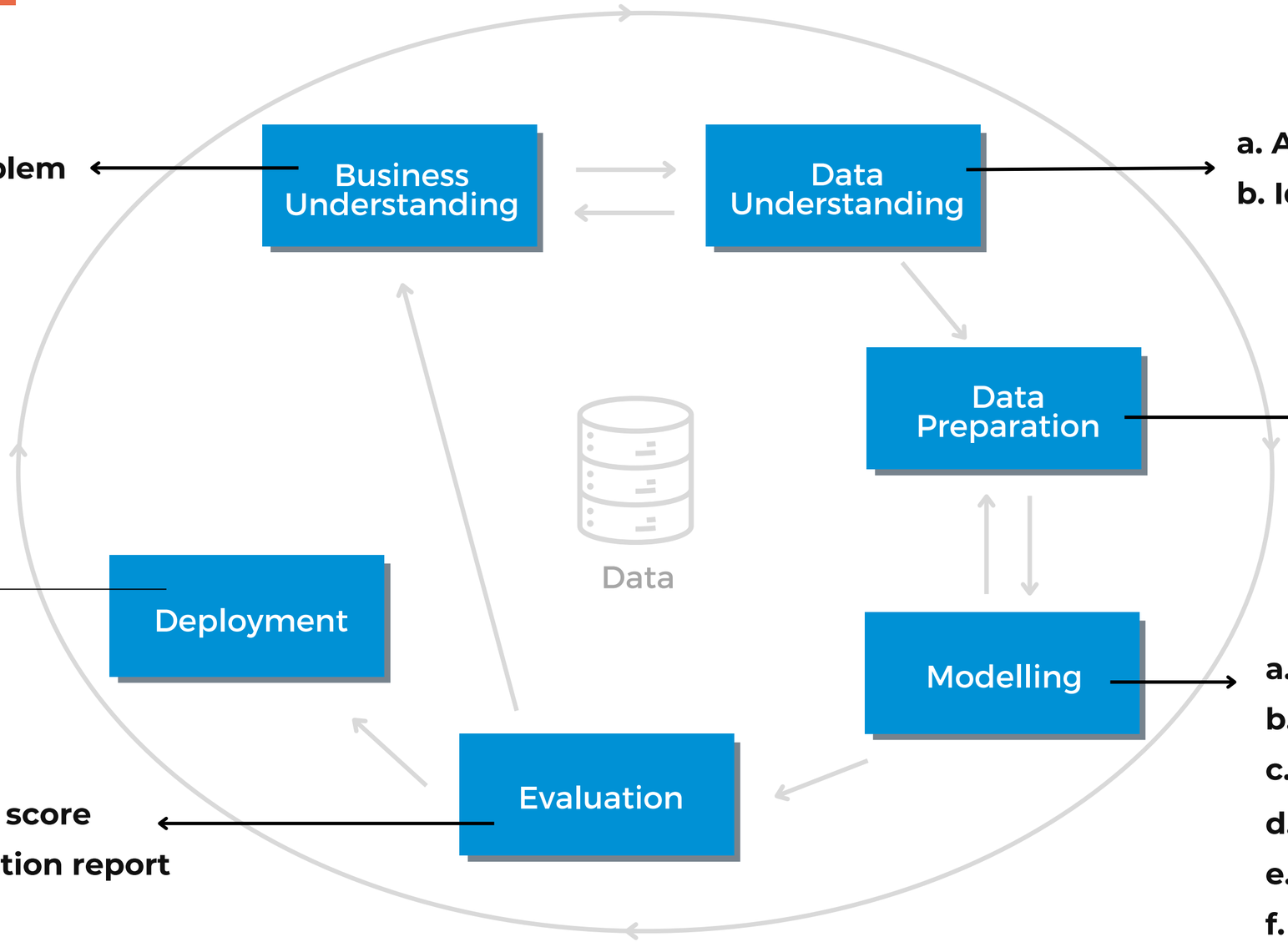
a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Web application

a. Accuracy score
b. Classification report

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

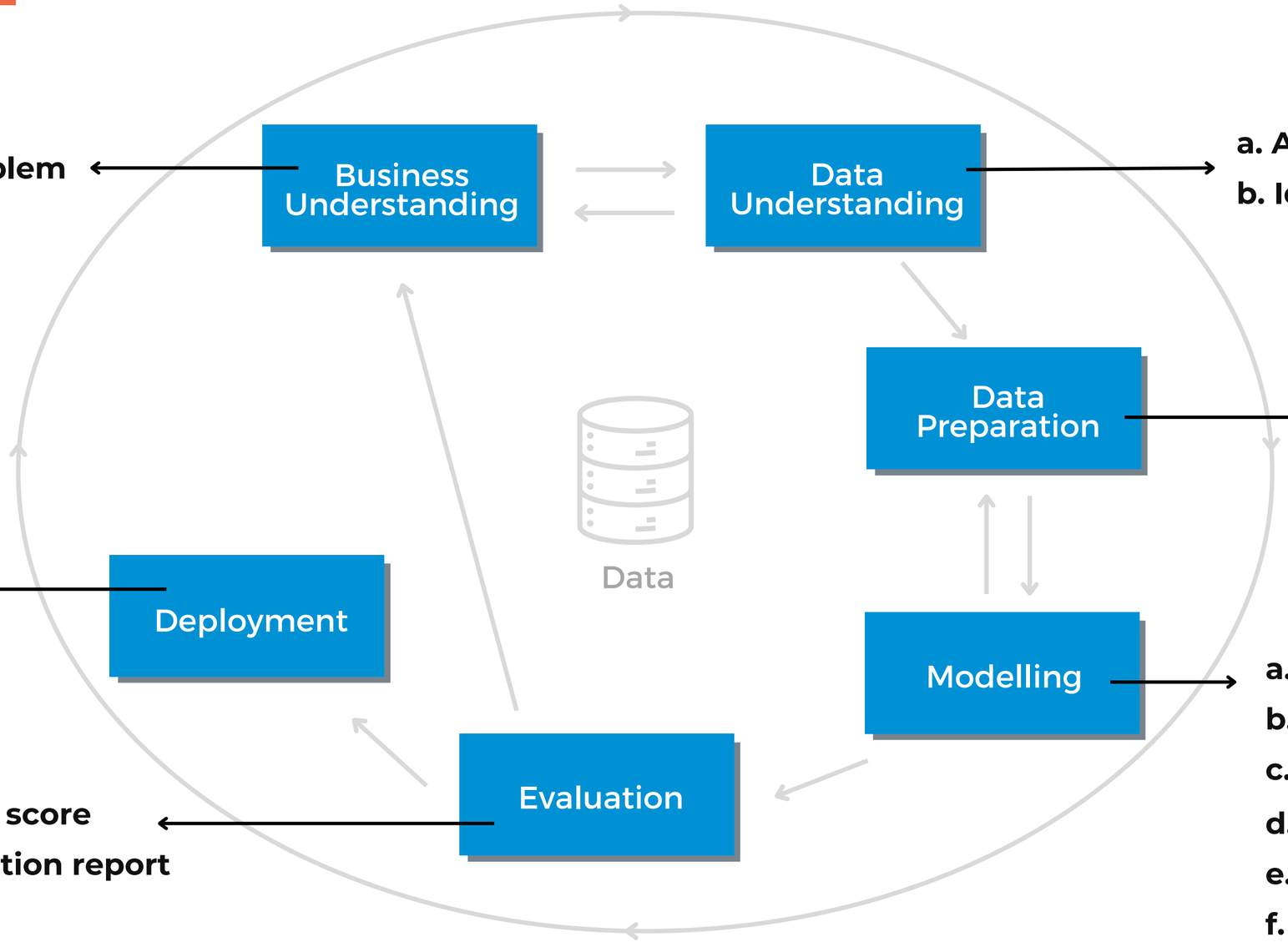
a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Accuracy score
b. Classification report

a. Web application

DS Methodology: Summary



a. Understand the problem

a. Acquire data
b. Identify relevant columns

a. Standardise data
b. Identify dominant learning style
c. EDA
d. Encode data

a. SVM
b. Random Forest
c. Decision Tree
d. XGB
e. kNN
f. Logistic Regression

a. Accuracy score
b. Classification report

a. Web application

Summary of Tools Used

Showcasing Smart-Learn

Conclusion

Objective	Approach/Results	Status
• To **develop** a learning object classification based on personalisation **model**	• Model: ◦ SVM ◦ Random Forest ◦ Decision Tree ◦ XGB ◦ kNN ◦ Logistic Regression	**Achieved**
• To **evaluate** a learning object classification based on personalisation **model**	• Evaluation metrics: ◦ Accuracy score ◦ Classification report • Best model: ◦ SVM	**Achieved**
• To develop a functional data product **web application** which can provide learning objects recommendation	• Web application: ◦ Smart Learn	**Achieved**

Table 5: Status of objectives achievement

Thank you!