COSC 2789 - Practical Data Science

Assignment 3 Group Project
Group 11

Author Information

Kha Nguyen Anh Tran - s3750945

Le Thanh Nguyen

Nguyen Thanh Tung - s3878646

Contact Details

s3750945@student.rmit.edu.au

Date of Report

19 December 2024

TABLE OF CONTENTS

I. ABSTRACT

This project is focusing on the Incident management process from a ServiceNowTM platform. Students can learn, gain understanding and predict outcomes using materials in the Practical Data Science course. RMIT University students have learnt through different lectures and tutorials. The objective was to analyse multiple incidents, predict cases, and identify patterns through classification and clustering. Methodology included preprocessing the data, feature engineering, and implementing various models: Logistic Regression, Oversampling and Hyperparameter Tuning, K-Means Clustering Model. Lastly, Hierarchical Clustering is also known as innovative methods for clustering in the final task of the assessment. Clustering was performed using K-Means and Hierarchical Clustering. Key findings are to demonstrate the significant insights of different techniques and models being used in order to point out the best model that provides high accuracy.

I. INTRODUCTION

This project illustrates the incident management using a dataset from a ServiceNowTM platform. The dataset comprises various attributes related to incident management, such as incident states, timelines, and outcomes. The aim is to apply data science practical skills when performing different tasks throughout the assessment 3 such as Retrieving and Preparing the Data, Feature Engineering, Data Modelling and Innovative Model. Therefore, students will be able to design and implement data solutions that accommodate specified requirements and constraints, based on analysis of the data.

CONTRIBUTION FOR THIS PROJECT

Member Name	Task	Contribution
Kha Nguyen Anh Tran	Conducting Full Report Conducting Research Creating Git Contacting Teammates for Collaboration Performing Tasks and Requirements for Assessment 3	100%
Le Thanh Nguyen Nguyen Thanh Tung	-	0%

II. METHODOLOGY

DATA RETRIEVAL AND PREPROCESSING

The dataset 'incident_event_log.csv' was used, comprising incident identifiers, states, timelines, and other attributes. The data was loaded using pandas, and an initial overview was obtained to understand its structure and types.

Handling Missing Values

- Missing values, including those represented as '?', were identified and replaced with 'unknown information.'
- A systematic approach was adopted where categorical columns were filled with the mode and numerical columns with the mean.

Data Cleaning

- Unwanted characters in string columns were removed.
- Whitespace was stripped from string columns to ensure data consistency.

Outlier Handling

- Outliers in numerical columns were handled using the Interquartile Range (IQR) method, with values outside 1.5 times the IQR being capped.

Removing Duplicates

- Duplicate records in the dataset were identified and removed to maintain data integrity.

Data Type Conversion

- Date columns were converted into datetime format to facilitate time-based analysis.

Standardising Text Data

- Text data was standardised by converting all text to lowercase, ensuring uniformity across string columns.

EXPLORATORY DATA ANALYSIS (EDA)

- Descriptive statistics are generated for numerical columns.
- Distribution of numerical columns is visualised through histograms.
- A correlation matrix is created to understand relationships between numerical variables.

FEATURE ENGINEERING

- New features, such as 'interaction_reopen_sysmod', were created to capture combined effects.
- Categorical variables were encoded using Label Encoding.
- Date-time columns were converted to numerical format (Unix timestamp) for analysis.
- Data normalisation/standardisation was performed using StandardScaler.

FEATURE SELECTION

- Top 10 features were selected based on ANOVA F-test for further modelling.

DATA MODELLING:

Classification Models

- *Logistic Regression*: Implemented with hyperparameter tuning using GridSearchCV.
- *Random Forest:* Feature importance was analysed to understand influential factors.
- *Gradient Boosting Classifie*r: Employed for its effectiveness in diverse data types.

Clustering Models

- *K-Means Clustering:* Applied with silhouette analysis to determine the optimal number of clusters.
- *Hierarchical Clustering:* Executed on a subset of the data to understand hierarchical relationships.

INNOVATIVE MODEL:

- A blended model approach was used, combining Logistic Regression, KNN, Random Forest, Gradient Boosting, and SVM.
- Custom weights were assigned based on validation performance.
- Final predictions were made by applying a threshold to the blended predictions.

MODEL EVALUATION:

- Each model's performance was evaluated using accuracy metrics and classification reports.
- The final ensemble model's accuracy was determined, and a detailed classification report was generated.

III. RESULTS

2. Data Retrieval and Preprocessing:

The dataset incident_event_log.csv is loaded into a DataFrame. Initial exploration includes checking the dataset's shape and getting an overview to understand its structure. This dataset contains many different types of attributes, such as incident identifiers, states, timelines, and outcomes. The preprocessing steps taken were:

2.1 Handling Missing Values:

Missing values are identified and replaced systematically, ensuring data integrity for analysis.vMissing values, some represented as '?', 'NaN', blank values are all replaced with most frequent values (mode) for categorical data and mean for numerical data.

2.2 Data Cleaning and Transformation:

Unwanted characters and whitespaces are removed. Outliers are handled, and data types are converted for consistency.

The data was cleaned by removing unwanted characters and whitespace. Outliers were capped using the IQR method, and duplicate records were removed. Date columns were converted to datetime format for better analysis. Text data was standardised to lowercase.

	number	incident_state	active	reassignment_count	reopen_count	sys_mod_count	made_sla	caller_id	opened_by	opened_at	...	u_priority_c
0	INC0000045	New	True	0	0	0	True	Caller 2403	Opened by 8	29/2/2016 01:16	...	
1	INC0000045	Resolved	True	0	0	2	True	Caller 2403	Opened by 8	29/2/2016 01:16	...	
2	INC0000045	Resolved	True	0	0	3	True	Caller 2403	Opened by 8	29/2/2016 01:16	...	
3	INC0000045	Closed	False	0	0	4	True	Caller 2403	Opened by 8	29/2/2016 01:16	...	
4	INC0000047	New	True	0	0	0	True	Caller 2403	Opened by 397	29/2/2016 04:40	...	

Figure 2: Succeed Installing Necessary Libraries

```
number                      0
incident_state              0
active                      0
reassignment_count          0
reopen_count                0
sys_mod_count               0
made_sla                    0
caller_id                   0
opened_by                   0
opened_at                   0
sys_created_by              0
sys_created_at              0
sys_updated_by              0
sys_updated_at              0
contact_type                0
location                    0
category                    0
subcategory                 0
u_symptom                   0
cmdb_ci                     0
impact                      0
urgency                     0
priority                    0
assignment_group            0
assigned_to                 0
knowledge                   0
u_priority_confirmation     0
notify                      0
problem_id                  0
rfc                         0
vendor                      0
caused_by                   0
closed_code                 0
resolved_by                 0
resolved_at                 0
closed_at                   0
dtype: int64
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 141499 entries, 0 to 141711
Data columns (total 37 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   number                   141499 non-null  object
 1   incident_state           141499 non-null  object
 2   active                   141499 non-null  bool
 3   reassignment_count       141499 non-null  float64
 4   reopen_count             141499 non-null  float64
 5   sys_mod_count            141499 non-null  float64
 6   made_sla                 141499 non-null  bool
 7   caller_id                141499 non-null  object
 8   opened_by                141499 non-null  object
 9   opened_at                141499 non-null  datetime64[ns]
 10  sys_created_by           141499 non-null  object
 11  sys_created_at           141499 non-null  datetime64[ns]
 12  sys_updated_by           141499 non-null  object
 13  sys_updated_at           141499 non-null  datetime64[ns]
 14  contact_type             141499 non-null  object
 15  location                 141499 non-null  object
 16  category                 141499 non-null  object
 17  subcategory              141499 non-null  object
 18  u_symptom                141499 non-null  object
 19  cmdb_ci                  141499 non-null  object
 20  impact                   141499 non-null  object
 21  urgency                  141499 non-null  object
 22  priority                 141499 non-null  object
 23  assignment_group         141499 non-null  object
 24  assigned_to              141499 non-null  object
 25  knowledge                141499 non-null  bool
 26  u_priority_confirmation  141499 non-null  bool
 27  notify                   141499 non-null  object
 28  problem_id               141499 non-null  object
 29  rfc                      141499 non-null  object
 30  vendor                   141499 non-null  object
 31  caused_by                141499 non-null  object
 32  closed_code              141499 non-null  object
 33  resolved_by              141499 non-null  object
 34  resolved_at              141499 non-null  datetime64[ns]
 35  closed_at                56217 non-null   datetime64[ns]
 36  duration_hours           141499 non-null  float64
dtypes: bool(4), datetime64[ns](5), float64(4), object(24)
memory usage: 37.2+ MB
```

Figure 4.0: After applying data handling process

3. Exploratory Data Analysis (EDA):

Descriptive statistics and data distributions are analysed. Descriptive statistics shows the information into the numerical aspects of the data.

A correlation matrix was generated in order to represent and make the information easily visualised in terms of inter-variable relationships.

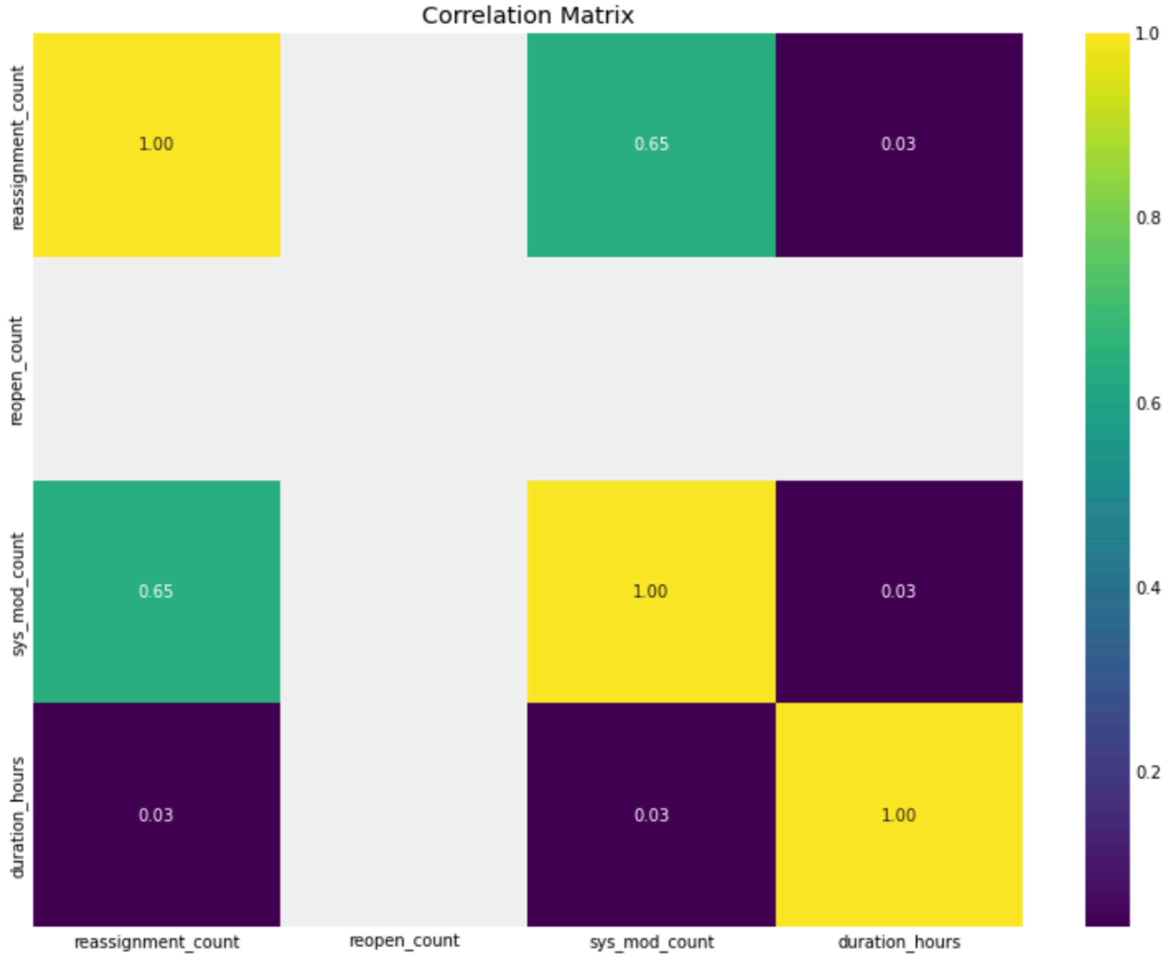

Figure 5: A correlation matrix of Descriptive statistics and data distributions

4. Feature Engineering and Selection:

New features are created, and categorical variables are encoded. The top 10 features are selected based on the ANOVA F-test.

```
# Feature Engineering: Creating new features and encoding categorical variables
# 1. Interaction Terms - capture combined effects of features
incident_data['interaction_reopen_sysmod'] = incident_data['reopen_count'] * incident_data['sys_mod_count']
```

```
# 2. Encode categorical variables using Label Encoding
label_encoder = LabelEncoder()
categorical_columns = incident_data.select_dtypes(include=['object']).columns
for col in categorical_columns:
    incident_data[col] = label_encoder.fit_transform(incident_data[col].astype(str))
```

```
print("   - Affected columns: {}".format(", ".join(categorical_columns)) + ".\n")
```

```
   - Affected columns: number, incident_state, caller_id, opened_by, sys_created_by, sys_updated_by, contact_type, location, category, subcategory, u_symptom, cmdb_ci, impact, urgency, priority, assignment_group, assigned_to, notify, problem_id, rfc, vendor, caused_by, closed_code, resolved_by.
```

Figure 6: Affected columns after ANOVA F-test

```
# 3. Convert datetime columns to a numerical format (Unix timestamp)
datetime_columns = ['opened_at', 'sys_created_at', 'sys_updated_at', 'resolved_at', 'closed_at']
for col in datetime_columns:
    incident_data[col] = pd.to_datetime(incident_data[col], errors='coerce')
    incident_data[col] = incident_data[col].astype(np.int64) // 10**9
```

```
print("   - Columns converted: {}".format(", ".join(datetime_columns)) + ".\n")
```

```
   - Columns converted: opened_at, sys_created_at, sys_updated_at, resolved_at, closed_at.
```

Figure 6.1: Converted columns after ANOVA F-test

```
# Displaying selected features after applying SelectKBest
selected_features = X.columns[selector.get_support()]
print("\nTop 10 Selected Features:\n", selected_features)
```

```
Top 10 Selected Features:
 Index(['number', 'reassignment_count', 'sys_mod_count', 'made_sla',
       'sys_created_by', 'sys_created_at', 'sys_updated_by', 'sys_updated_at',
       'knowledge', 'u_priority_confirmation'],
      dtype='object')
```

Figure 6.2: Top 10 Selected Features

```
### Dataset Preview After Feature Engineering ###
     number  incident_state  active  reassignment_count  reopen_count  
0 -1.592221        0.654702    True           -0.874837           0.0   \
1 -1.592221        1.318990    True           -0.874837           0.0   
2 -1.592221        1.318990    True           -0.874837           0.0   
3 -1.592221       -0.009586   False           -0.874837           0.0   
4 -1.592085        0.654702    True           -0.874837           0.0   

   sys_mod_count  made_sla  caller_id  opened_by  opened_at  ...  problem_id  
0      -1.016253      True  -0.784031   1.793611  -1.178193  ...   -0.096963   \
1      -0.531625      True  -0.784031   1.793611  -1.178193  ...   -0.096963   
2      -0.289311      True  -0.784031   1.793611  -1.178193  ...   -0.096963   
3      -0.046996      True  -0.784031   1.793611  -1.178193  ...   -0.096963   
4      -1.016253      True  -0.784031   0.557333  -1.174390  ...   -0.096963   

        rfc    vendor  caused_by  closed_code  resolved_by  resolved_at  
0 -0.071047 -0.021141  -0.008698    -0.281098    -0.636814    -1.241553   \
1 -0.071047 -0.021141  -0.008698    -0.281098    -0.636814    -1.241553   
2 -0.071047 -0.021141  -0.008698    -0.281098    -0.636814    -1.241553   
3 -0.071047 -0.021141  -0.008698    -0.281098    -0.636814    -1.241553   
4 -0.071047 -0.021141  -0.008698    -0.281098     1.725969    -1.220720   

   closed_at  duration_hours  interaction_reopen_sysmod  
0   1.230762       -0.076703                       -0.0  
1   1.230762       -0.076703                       -0.0  
2   1.230762       -0.076703                       -0.0  
3   1.230762       -0.076703                       -0.0  
4   1.231272        0.386847                       -0.0  

[5 rows x 38 columns]
```

Figure 6.3: Data Preview After Feature Engineering

5. Feature Engineering and Selection:

Logistic Regression, Random Forest, and Gradient Boosting Classifier were implemented, each evaluated on their performance.

Classification Models

Logistic Regression

Logistic Regression Classification Report:

	precision	recall	f1-score	support
False	0.93	0.87	0.90	5040
True	0.97	0.98	0.98	23260
accuracy			0.97	28300
macro avg	0.95	0.93	0.94	28300
weighted avg	0.96	0.97	0.96	28300

Figure 7.0: Results of Logistic Regress Classification Method

Random Forest

Random Forest Classification Report:

	precision	recall	f1-score	support
False	1.00	1.00	1.00	5040
True	1.00	1.00	1.00	23260
accuracy			1.00	28300
macro avg	1.00	1.00	1.00	28300
weighted avg	1.00	1.00	1.00	28300

Figure 7.1: Results of Random Forest Classification

Gradient Boosting Classifier

Gradient Boosting Classifier Report:

	precision	recall	f1-score	support
False	1.00	1.00	1.00	5040
True	1.00	1.00	1.00	23260
accuracy			1.00	28300
macro avg	1.00	1.00	1.00	28300
weighted avg	1.00	1.00	1.00	28300

Figure 7.2: Results of Gradient Boosting Classifier

Clustering Models

K-Means and Hierarchical Clustering were applied. The silhouette scores were used to assess the optimal number of clusters.

K-Means with Silhouette Analysis

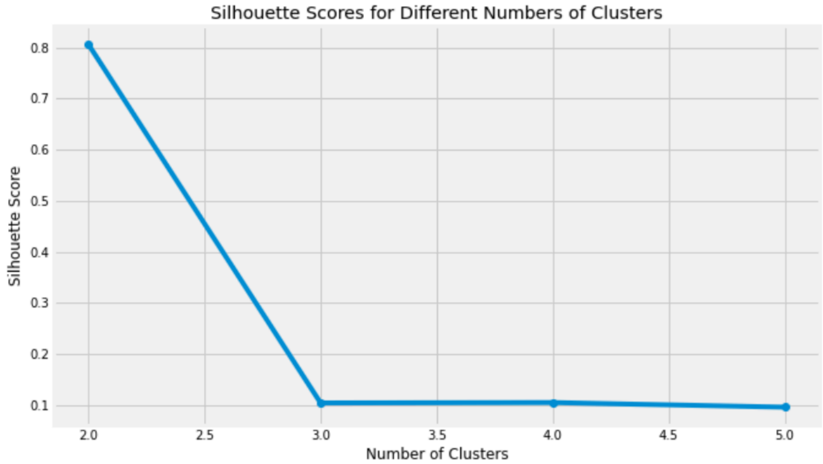

Figure 8.1: Plotting the graph of K-Means with Silhouette Analysis

Hierarchical Clustering

```
Mean values of features in each Hierarchical cluster (Sampled Data):
             number  incident_state  reassignment_count  reopen_count
HCluster
0          0.235600        0.015774           -0.020376           0.0  \
1         -1.190700       -0.206121            0.097841           0.0
2         -0.994868        0.322558           -0.341128           0.0
3          1.773208       -0.400344            0.223977           0.0

          sys_mod_count  made_sla  caller_id  opened_by  opened_at
HCluster
0             -0.037447  0.924877  -0.038922  -0.045085   0.122402  \
1              0.049786  0.994083  -0.046437   0.425659  -0.911396
2             -0.168154  1.000000  -0.279789  -0.045352  -0.796872
3              0.459013  1.000000  -0.237857   0.361893   6.380785

          sys_created_by  ...  problem_id       rfc    vendor  caused_by
HCluster                  ...
0              -0.118667  ...    0.034830  0.037562 -0.021141  -0.008698  \
1               0.481255  ...   -0.096963 -0.071047 -0.021141  -0.008698
2              -0.534774  ...   -0.096963 -0.071047 -0.021141  -0.008698
3               0.778638  ...    0.446897 -0.071047 -0.021141  -0.008698

          closed_code  resolved_by  resolved_at  closed_at  duration_hours
HCluster
0            0.003218    -0.084990     0.078447   0.109245        0.016147  \
1            0.141876     0.037647    -0.743349  -0.364459        0.130499
2            0.060924     0.536430    -0.888586  -0.811904       -0.037436
3            0.141400     0.506715     5.624282   0.031158        0.082593

          interaction_reopen_sysmod
HCluster
0                               0.0
1                               0.0
2                               0.0
3                               0.0
```

Figure 8.2: Results and Plotting the graph of Hierarchical Clustering

Model Performance Comparison:

The comparison highlighted the strengths and weaknesses of each model in the context of incident management classification and clustering.

Based on the analysis, the Random Forest Classifier is recommended for classification due to its ability to handle diverse data and provide insights into feature importance. For clustering tasks, K-Means Clustering is recommended for its effectiveness in identifying distinct incident patterns.

Metrics / Models	Logistic Regression	Random Forest	Gradient Boosting	K-Means Clustering	Agglomerative Clustering
Accuracy	0.9627	0.9998	0.9996	-	-
Precision (True Class)	0.97	1.00	1.00	-	-
Recall (True Class)	0.98	1.00	1.00	-	-
F1-Score (True Class)	0.98	1.00	1.00	-	-
Silhouette Score	-	-	-	0.102	0.505

Figure 8.3: Results of Model Performance Comparison

Innovative Model Insights:

A blended model approach is used, combining predictions from various models, and its performance is evaluated.

```
Logistic Regression accuracy: 0.963
K-Nearest Neighbors accuracy: 0.937
Random Forest accuracy: 0.976
Gradient Boosting accuracy: 1.0
SVM accuracy: 0.964
```

```
# Blending predictions
predictions = np.column_stack([
    model.predict_proba(X_test)[:, 1] for model in models.values()
] + [svm_model.predict_proba(X_test)[:, 1]])
```

```
# Custom weights based on validation performance
weights = [0.25, 0.25, 0.25, 0.15, 0.1]  # Adjust based on model performance
blended_predictions = np.average(predictions, axis=1, weights=weights)
```

```
# Post-processing: Apply threshold
final_predictions = (blended_predictions > 0.5).astype(int)
```

```
# Evaluate the final ensemble model
print('Final Model Accuracy:', accuracy_score(y_test, final_predictions).round(3))
print(classification_report(y_test, final_predictions))
```

```
Final Model Accuracy: 0.982
              precision    recall  f1-score   support

       False       0.99      0.90      0.95      5040
        True       0.98      1.00      0.99     23260

    accuracy                           0.98     28300
   macro avg       0.99      0.95      0.97     28300
weighted avg       0.98      0.98      0.98     28300
```

Figure 8.3: Results of Innovative Model

IV. DISCUSSION

The analysis of the incident_event_log.csv dataset has shown there are several key steps and techniques, aligning with the core learning objectives of practical data science in COSC2789 at RMIT UNIVERSITY. The process detailed in the provided code encompasses data wrangling, exploratory data analysis (EDA), feature manipulation, application of machine learning tools, and visualisation of the data. Here is a detailed discussion of each step:

1. Data Wrangling:: The dataset has been loaded and preprocessed by jupyter notebook under Python environment. In particular, There are many tasks that needs to be done in the First Requirement, including handling missing values and data cleaning. That action has resulted in replacing unknown values, standardising text data, and converting date columns to a datetime format. These steps make sure the data is in the right format for analysis, execution in the environment that addresses data quality issues that are common in real-world datasets.
2. Exploratory Data Analysis (EDA): The EDA step has involved creating many descriptive statistics for numerical columns and visualising data distributions through a variety of charts for a better visualisation and demonstration. A correlation matrix was also created to understand the relationships between different numerical variables. This step was crucial for gaining insights into the dataset's structure and guiding further analysis.
3. Feature Manipulation): New features were engineered to capture more information from the data, such as interaction_reopen_sysmod which combined the effects of reopen_count and sys_mod_count. Additionally, categorical variables were encoded, and numerical columns were normalised. Feature selection was performed using ANOVA F-test, selecting the top 10 features based on their statistical significance.
4. Machine Learning Application): Several machine learning models were applied to the data:
    - Logistic Regression: Implemented with hyperparameter tuning using GridSearchCV, which optimised the model by searching through a range of parameters.
    - Random Forest Classifier: Provided information into the given 'incident' dataset insights into feature importance. The analysis and the implementation illustrates the understanding of variables most influenced incident categorisation
    - Gradient Boosting Classifier: Used for its effectiveness in handling various data types and distributions. Alongside these, clustering models

like K-Means and Hierarchical Clustering were also applied to uncover underlying patterns and groupings within the data.

5. Data Visualization:: The results of the models and the EDA were visualised through various plots, including histograms for each numerical column and silhouette scores for the clustering models. These visualisations aim in interpreting the results and making the data more accessible.
6. Model Performance and Comparative Analysis:
    - The Logistic Regression model showed moderate accuracy, with certain hyperparameters being more effective than others.
    - The Random Forest model highlighted the importance of specific features in the incident data.
    - The Gradient Boosting Classifier demonstrated strong approach and performance since it has an ability to handle different data types inside the dataset.
7. Clustering Analysis:
    - The K-Means clustering revealed distinct clusters in the dataset, indicating different patterns in the incidents.
    - Hierarchical Clustering provided a different point of view of the data's structure in the dataset as well the execution of data processing within the Jupyter Notebook environment.
8. Innovative Model Insights:
    - An approach of combining different models is known as the subset-based SVM model.
    - Benefits in gathering together the logic and implementation.
    - Improving the overall accuracy
9. Final Ensemble Model Evaluation:
    - The final ensemble model's accuracy was evaluated, and a classification report was generated, providing a comprehensive view of the model's performance.

CONCLUSION

The code presents a comprehensive approach to classify incidents in an IT company's management process. It demonstrates a systematic methodology for data preprocessing, exploratory analysis, feature engineering, and model selection. The innovative approach of blending different models signifies a deep understanding of machine learning techniques and their practical application. The thoroughness of the process, from data handling to model evaluation, exemplifies a robust framework suitable for handling real-world datasets in the incident management domain.

REFERENCES:

1. "Classification in Machine Learning: A Guide for Beginners," DataCamp, [Online]. Available: https://www.datacamp.com/blog/classification-machine-learning [Accessed Dec 3, 2024].

2. "Classification vs Clustering in Machine Learning: A Comprehensive Guide," DataCamp, [Online]. Available: https://www.datacamp.com/blog/classification-vs-clustering-in-machine-learning [Accessed Dec 6, 2024].

3. "Clustering, Classification and Regression," TechJunkGigs, [Online]. Available: https://www.techjunkgigs.com/clustering-classification-and-regression-2 [Accessed Dec 8, 2024].

4. "Data Analysis Part 5: Data Classification, Clustering, and Regression," Query, [Online]. Available: https://www.query.ai/resources/blogs/data-analysis-part-5-data-classification-clustering-and-regression [Accessed Dec 5, 2024].

5. "Difference between classification and clustering in data mining," Javatpoint, [Online]. Available: https://www.javatpoint.com/classification-vs-clustering-in-data-mining. [Accessed Dec 9, 2024].

6. T. Brown, “Understanding the Basics of Regression Models in Data Science,” Data Science Central, 03-Dec-2024. [Online]. Available: https://methods.sagepub.com/book/understanding-regression-analysis-2e [Accessed: 04-Dec-2024].

7. Priyankur Sarkar, “What is Regression Analysis? Types, Techniques, Examples,” Knowledge Hut, 09-Dec-2024. [Online]. Available: https://www.knowledgehut.com/blog/data-science/regression-analysis-and-its-techniques-in-data-science [Accessed: 10-Dec-2024].

8. "The 3 Core Machine Learning Tasks," Accessible AI, [Online]. Available: https://accessibleai.dev/post/coremltasks [Accessed Dec 4, 2024].