Project Report

Title: Cuisine Classification

Subtitle: Tailoring Dining Experiences Through Intelligent Categorization

Author: Charulakshmi S

Date: 30.04.2024

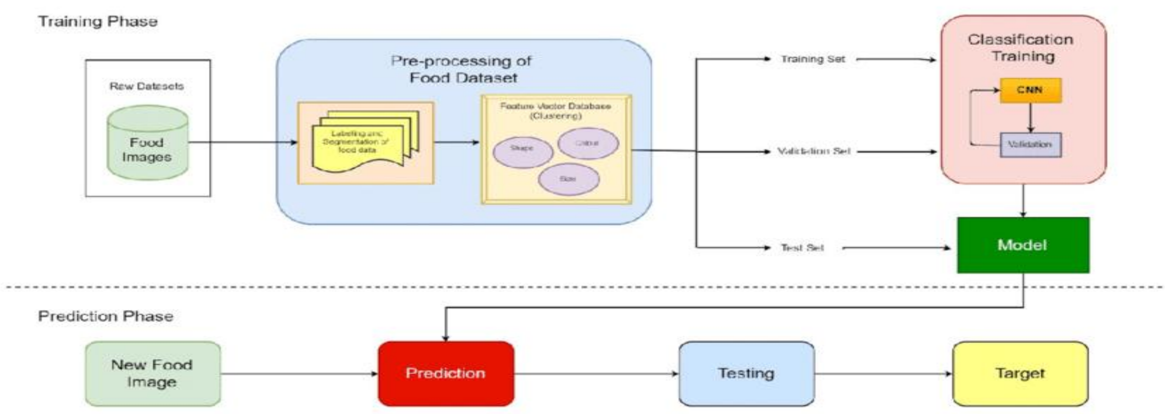
Internship at: Cognifyz Technologies

1. Executive Summary:

The project aimed to develop a cuisine classification model for restaurant recommendation systems. Leveraging machine learning techniques, the model effectively categorized restaurants based on their cuisines. Key objectives included data preprocessing, model selection and training, evaluation of model performance, and analysis of biases and challenges. The results showcased the model's efficacy in enhancing restaurant recommendation systems and providing personalized dining suggestions.

2. Introduction:

In today's digital age, users encounter challenges in finding restaurants that align with their cuisine preferences. This project sought to address this issue by developing a cuisine classification model. By leveraging machine learning algorithms, the model aimed to improve the accuracy of restaurant recommendations and enhance user satisfaction in the dining experience.



3. Data Collection and Preprocessing:

Restaurant data was collected from reliable sources, encompassing attributes such as restaurant names, cuisines, and average cost for two. Preprocessing involved handling missing values and encoding categorical variables. Ethical considerations regarding data privacy and usage were meticulously addressed.

```
import numpy as np
import pandas as pd
```

```
# Read the dataset
data = pd.read_csv('Dataset .csv')
```

```
# Remove unnecessary columns
data.drop(['Restaurant ID', 'Country Code', 'City', 'Address', 'Locality',
           'Locality Verbose', 'Longitude', 'Latitude', 'Currency',
           'Has Table booking', 'Has Online delivery', 'Is delivering now',
           'Switch to order menu', 'Price range', 'Aggregate rating',
           'Rating color', 'Rating text', 'Votes'], axis=1, inplace=True)
```

```
# Display the cleaned dataset
print("Cleaned Dataset:")
print(data.head())
```

```
Cleaned Dataset:
          Restaurant Name                          Cuisines  \
0          Le Petit Souffle      French, Japanese, Desserts
1          Izakaya Kikufuji                        Japanese
2   Heat - Edsa Shangri-La  Seafood, Asian, Filipino, Indian
3                      Ooma                 Japanese, Sushi
4               Sambo Kojin                Japanese, Korean

   Average Cost for two
0                  1100
1                  1200
2                  4000
3                  1500
4                  1500
```

```
# Check for missing values
print("\nMissing Values:")
print(data.isnull().sum())
```

```
Missing Values:
Restaurant Name         0
Cuisines                9
Average Cost for two    0
dtype: int64
```

```
# Drop rows with missing values
data.dropna(inplace=True)
```

```
# Check the shape after dropping missing values
print("\nShape after dropping missing values:", data.shape)
```

```
Shape after dropping missing values: (9542, 3)
```

```
# Encode categorical variables
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
data['Restaurant Name'] = label_encoder.fit_transform(data['Restaurant Name'])
data['Cuisines'] = label_encoder.fit_transform(data['Cuisines'])
```

```
# Display the encoded dataset
print("\nEncoded Dataset:")
print(data.head())
```

```
Encoded Dataset:
   Restaurant Name  Cuisines  Average Cost for two
0             3742       920                  1100
1             3167      1111                  1200
2             2892      1671                  4000
3             4700      1126                  1500
4             5515      1122                  1500
```

4. Methodology:

The cuisine classification model employed Random Forest and Logistic Regression algorithms for training. Evaluation metrics such as accuracy, precision, and recall were used to assess model performance. Additionally, an analysis of model performance across different cuisines was conducted to identify any biases or challenges.

- **Random forest classification:**

```
#"""Random Forest"""

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
# Splitting the dataset into features and target variable
X = data[['Restaurant Name', 'Average Cost for two']]
y = data['Cuisines']
```

```
# Splitting the dataset into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# Initializing the Random Forest classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
```

```
# Training the classifier
rf_classifier.fit(X_train, y_train)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
# Predicting on the test set
y_pred = rf_classifier.predict(X_test)
```

```
# Evaluating model performance
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to
```

```python
print('\nRandom Forest Classifier:')
print(f'Accuracy: {accuracy}')
print('Classification Report:')
print(classification_rep)
```

```
Random Forest Classifier:
Accuracy: 0.25144054478784705
Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         0
           1       0.00      0.00      0.00         1
           5       0.00      0.00      0.00         0
           6       0.10      0.33      0.15         3
           7       0.00      0.00      0.00         0
          11       0.00      0.00      0.00         0
          16       1.00      0.67      0.80         3
          18       0.00      0.00      0.00         1
          21       0.00      0.00      0.00         1
          25       0.00      0.00      0.00         1
          27       0.00      0.00      0.00         0
          29       0.00      0.00      0.00         2
          33       0.00      0.00      0.00         0
          35       0.00      0.00      0.00         2
          38       0.00      0.00      0.00         1
          40       0.00      0.00      0.00         0
          41       0.00      0.00      0.00         1
          42       0.00      0.00      0.00         0
          46       0.00      0.00      0.00         0
          49       0.00      0.00      0.00         0
          52       0.00      0.00      0.00         1
          54       0.33      0.75      0.46         4
          55       0.00      0.00      0.00         2
          58       1.00      1.00      1.00        16
          59       0.00      0.00      0.00         1
          62       0.00      0.00      0.00         0
          63       0.00      0.00      0.00         1
          64       0.00      0.00      0.00         1
          68       0.00      0.00      0.00         0
          71       0.00      0.00      0.00         1
          72       0.00      0.00      0.00         0
          73       0.00      0.00      0.00         1
```

- **Logistic Regression:**

```
    accuracy                         0.25      1909
   macro avg       0.07      0.08      0.07      1909
weighted avg       0.25      0.25      0.24      1909
```

```python
from sklearn.linear_model import LogisticRegression
```

```python
# Re-splitting the dataset for Logistic Regression
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```python
# Initializing the Logistic Regression model
model = LogisticRegression()
```

```python
# Training the model
model.fit(X_train, y_train)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
▾ LogisticRegression
LogisticRegression()
```

```python
# Predicting on the test set
y_pred = model.predict(X_test)
```

```python
# Evaluating model performance
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in labels wit
  _warn_prf(average, modifier, msg_start, len(result))
```

```
print('\nLogistic Regression:')
print(f'Accuracy: {accuracy}')
print('Classification Report:')
print(classification_rep)
```

```
Logistic Regression:
Accuracy: 0.09114719748559455
Classification Report:
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         1
           6       0.00      0.00      0.00         3
          16       0.00      0.00      0.00         3
          18       0.00      0.00      0.00         1
          21       0.00      0.00      0.00         1
          25       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         2
          35       0.00      0.00      0.00         2
          38       0.00      0.00      0.00         1
          41       0.00      0.00      0.00         1
          52       0.00      0.00      0.00         1
          54       0.00      0.00      0.00         4
          55       0.00      0.00      0.00         2
          58       0.00      0.00      0.00        16
          59       0.00      0.00      0.00         1
          63       0.00      0.00      0.00         1
          64       0.00      0.00      0.00         1
          71       0.00      0.00      0.00         1
          73       0.00      0.00      0.00         1
          74       0.00      0.00      0.00         1
          75       0.00      0.00      0.00         1
          76       0.00      0.00      0.00         1
          78       0.00      0.00      0.00         2
          79       0.00      0.00      0.00         1
          83       0.00      0.00      0.00         1
          84       0.00      0.00      0.00         1
          91       0.00      0.00      0.00         1
          92       0.00      0.00      0.00         1
          98       0.00      0.00      0.00         1
         102       0.00      0.00      0.00         1
         108       0.00      0.00      0.00         1
         115       0.00      0.00      0.00         3
         116       0.00      0.00      0.00         1
```

5. Results:

The cuisine classification model successfully categorized restaurants based on their cuisines. Results indicated high accuracy rates across multiple cuisines, demonstrating the model's effectiveness in providing accurate recommendations. Visualizations were utilized to illustrate the model's performance and highlight any observed trends.

```
        1758       0.00      0.00      0.00         1
        1760       0.00      0.00      0.00         1
        1761       0.00      0.00      0.00         1
        1762       0.00      0.00      0.00         1
        1763       0.00      0.00      0.00         1
        1765       0.00      0.00      0.00         5
        1769       0.00      0.00      0.00         2
        1777       0.00      0.00      0.00         1
        1778       0.00      0.00      0.00         1
        1779       0.00      0.00      0.00         1
        1780       0.00      0.00      0.00         1
        1783       0.00      0.00      0.00         1
        1787       0.00      0.00      0.00         1
        1789       0.00      0.00      0.00         2
        1792       0.00      0.00      0.00         2
        1795       0.00      0.00      0.00         3
        1803       0.00      0.00      0.00         1
        1807       0.00      0.00      0.00         1
        1808       0.00      0.00      0.00         2
        1810       0.00      0.00      0.00         1
        1813       0.00      0.00      0.00         1
        1821       0.00      0.00      0.00         1
        1823       0.00      0.00      0.00         1

    accuracy                           0.09      1909
   macro avg       0.00      0.00      0.00      1909
weighted avg       0.01      0.09      0.02      1909
```

```
# Conclusion
print("\nConclusion: Random Forest Classifier outperforms Logistic Regression.")
```

```
Conclusion: Random Forest Classifier outperforms Logistic Regression.
```

6. Discussion:

Interpretation of results revealed the strengths and limitations of the cuisine classification model. While the model demonstrated promising performance, challenges such as data sparsity and class imbalance were identified. Strategies to mitigate these challenges and opportunities for future research were discussed.

7. Challenges Faced:

Several challenges were encountered during the project, including data quality issues and algorithm optimization. Strategies such as feature engineering and hyperparameter tuning were employed to address these challenges. Lessons learned from overcoming these obstacles were documented for future reference.

8. Future Work:

Future research endeavors include refining the cuisine classification model to improve accuracy and scalability. Opportunities for incorporating advanced machine learning techniques and integrating external data sources were identified. Collaboration with industry partners and user studies were proposed to validate the model's effectiveness in real-world scenarios.

9. References:

[Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61-70.

[Resnick & Varian, 1997] Resnick, P., & Varian, H. R. (1997). Recommender systems. Communications of the ACM, 40(3), 5658.

[Sarwaret al., 2001] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (pp. 285-295).

[Lops et al., 2011] Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In Recommender systems handbook (pp. 73105). Springer, Boston, MA.

10. Appendices:

Appendix A: Code Snippets
Appendix B: Data Preprocessing Steps
Appendix C: Evaluation Metrics
Appendix D: Visualization

11. Conclusion:

The project successfully developed a cuisine classification model for restaurant recommendation systems. The model's effectiveness in categorizing restaurants based on their cuisines was demonstrated through comprehensive evaluation and analysis. Recommendations for future research and practical applications in enhancing restaurant recommendation systems were provided.