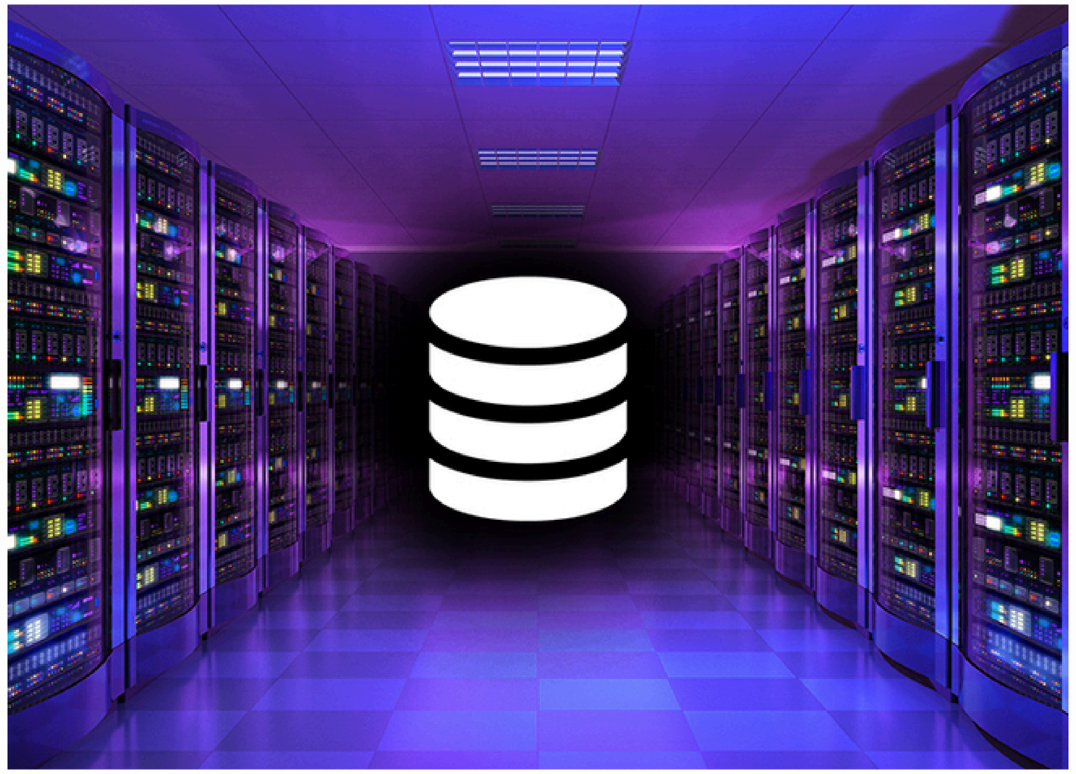


DATABASE-DRIVEN SENTIMENT ANALYSIS

A MACHINE LEARNING APPROACH TO IDENTIFYING MALICIOUS YELP REVIEWS

NISHANTH NANDAKUMAR
JONGKYU LEE
DINESH RAJA NATARAJAN

INTRODUCTION

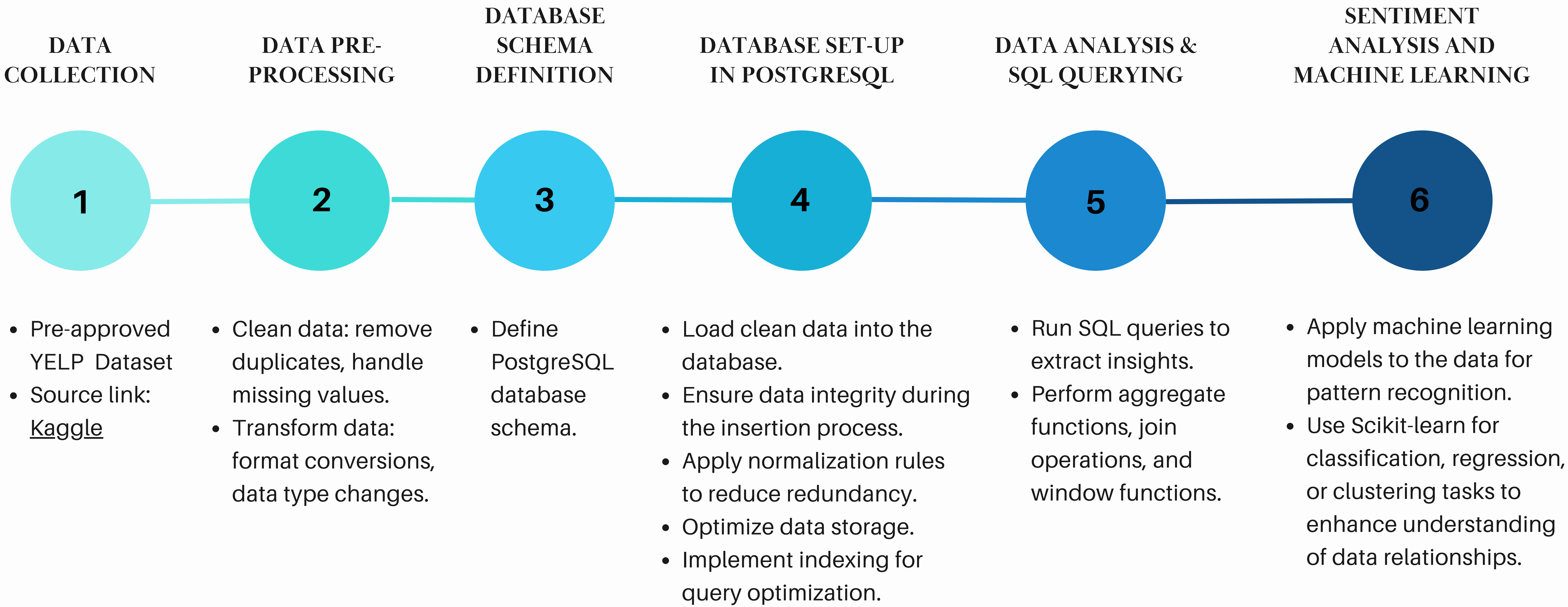
Context: In today's digital age, online reviews significantly impact consumer behavior and business reputation. However, the prevalence of fake or malicious reviews can undermine trust in these digital platforms.

Challenge: Differentiating between genuine and malicious reviews remains a complex issue due to the subtle nuances in language and the vast volume of data.

Project Objective: A Database Management and Machine Learning Approach to Identifying Authentic Yelp Reviews to develop a robust system that uses advanced database management techniques to accurately identify authentic reviews on Yelp. The use of machine learning is illustrative, supporting the primary focus on database management for this class.

Importance: By improving the accuracy of review authenticity, we aim to enhance user trust in Yelp’s review system, thereby supporting better decision-making for consumers and fairer conditions for businesses.

APPROACH



1 DATA COLLECTION

- Pre-approved YELP Dataset
- Source link: Kaggle

2 DATA PRE-PROCESSING

- Clean data: remove duplicates, handle missing values.
- Transform data: format conversions, data type changes.

3 DATABASE SCHEMA DEFINITION

- Define PostgreSQL database schema.

4 DATABASE SET-UP IN POSTGRESQL

- Load clean data into the database.
- Ensure data integrity during the insertion process.
- Apply normalization rules to reduce redundancy.
- Optimize data storage.
- Implement indexing for query optimization.

5 DATA ANALYSIS & SQL QUERYING

- Run SQL queries to extract insights.
- Perform aggregate functions, join operations, and window functions.

6 SENTIMENT ANALYSIS AND MACHINE LEARNING

- Apply machine learning models to the data for pattern recognition.
- Use Scikit-learn for classification, regression, or clustering tasks to enhance understanding of data relationships.

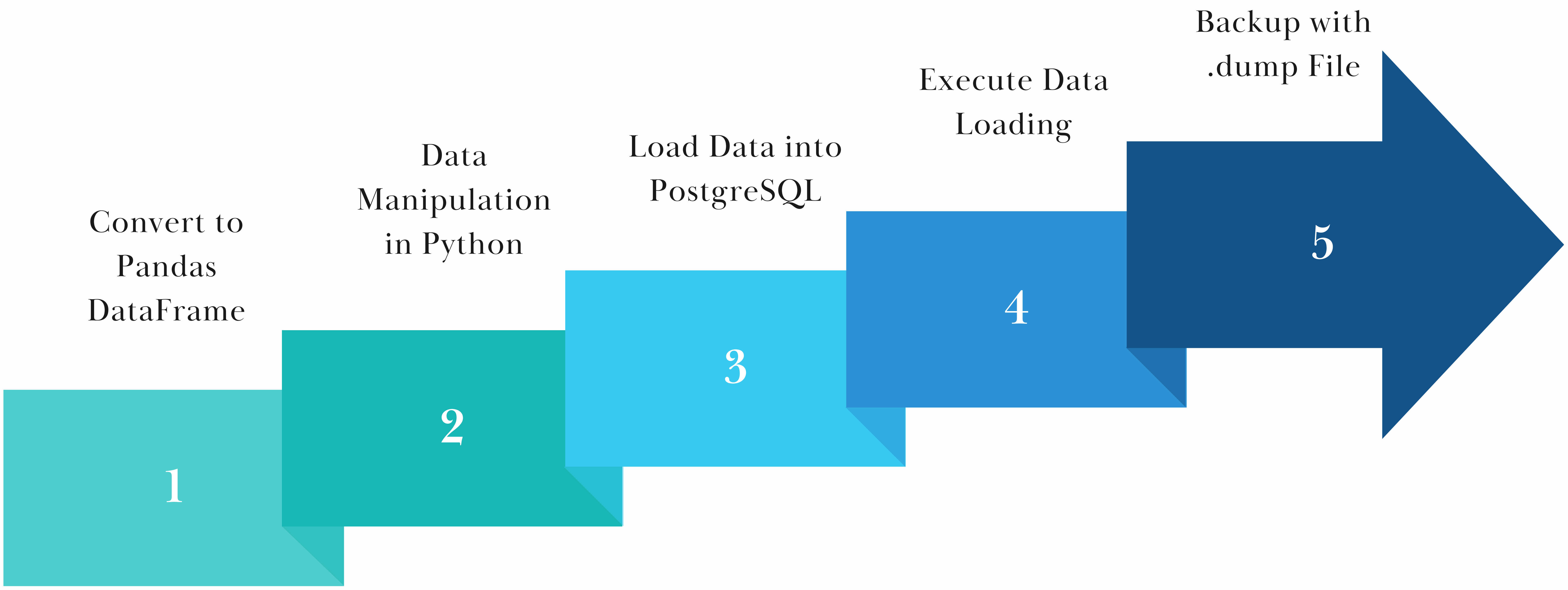
WHY POSTGRESQL?

Relational database management is done in SQL.
Its strong syntax, which accurately defines and manipulates data, helps it effectively handle structured data. SQL lets users get data, update records, and manage database structures using simple instructions.

- **Full-Text Search**: PostgreSQL has full-text search capabilities, which are essential for analyzing and searching through large volumes of text-based reviews.
- **Procedural Languages**: Supports various procedural languages like PL/pgSQL, which allows for writing complex stored procedures and functions for data analysis.
- **Extensibility**: It can be extended with custom functions developed in different programming languages such as C, Perl and Python (in our case) allowing for custom analytics.
- **Robust Security**: It offers strong security features to protect sensitive data.
- **Machine Learning Integration**: It can be integrated with machine learning tools, enabling the application of algorithms directly within the database for identifying patterns indicative of malicious content.
- **Open Source**: Being open-source, it has a wealth of plugins and tools, which is useful for our sentiment analysis on the Yelp Data.

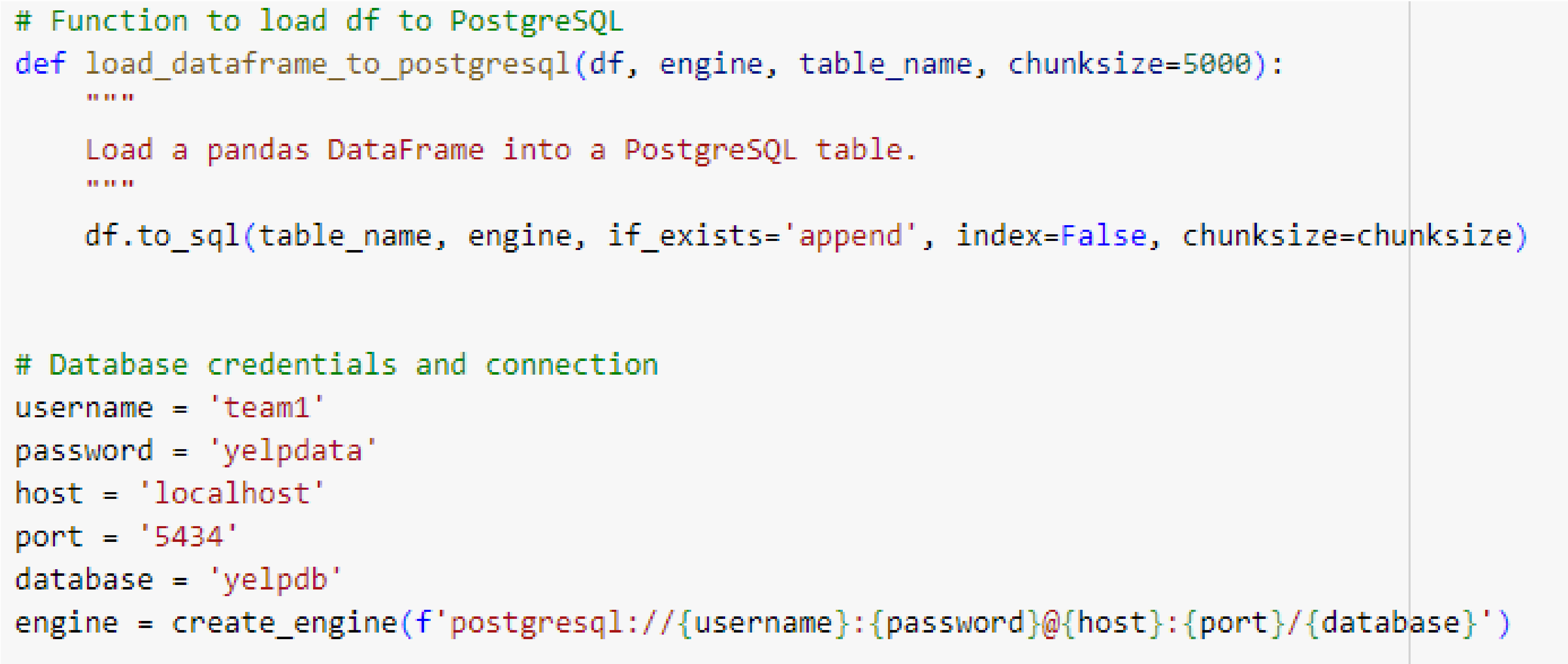
UPLOADING DATA INTO POSTGRESQL

> FOR DATA STORAGE AND MANAGEMENT

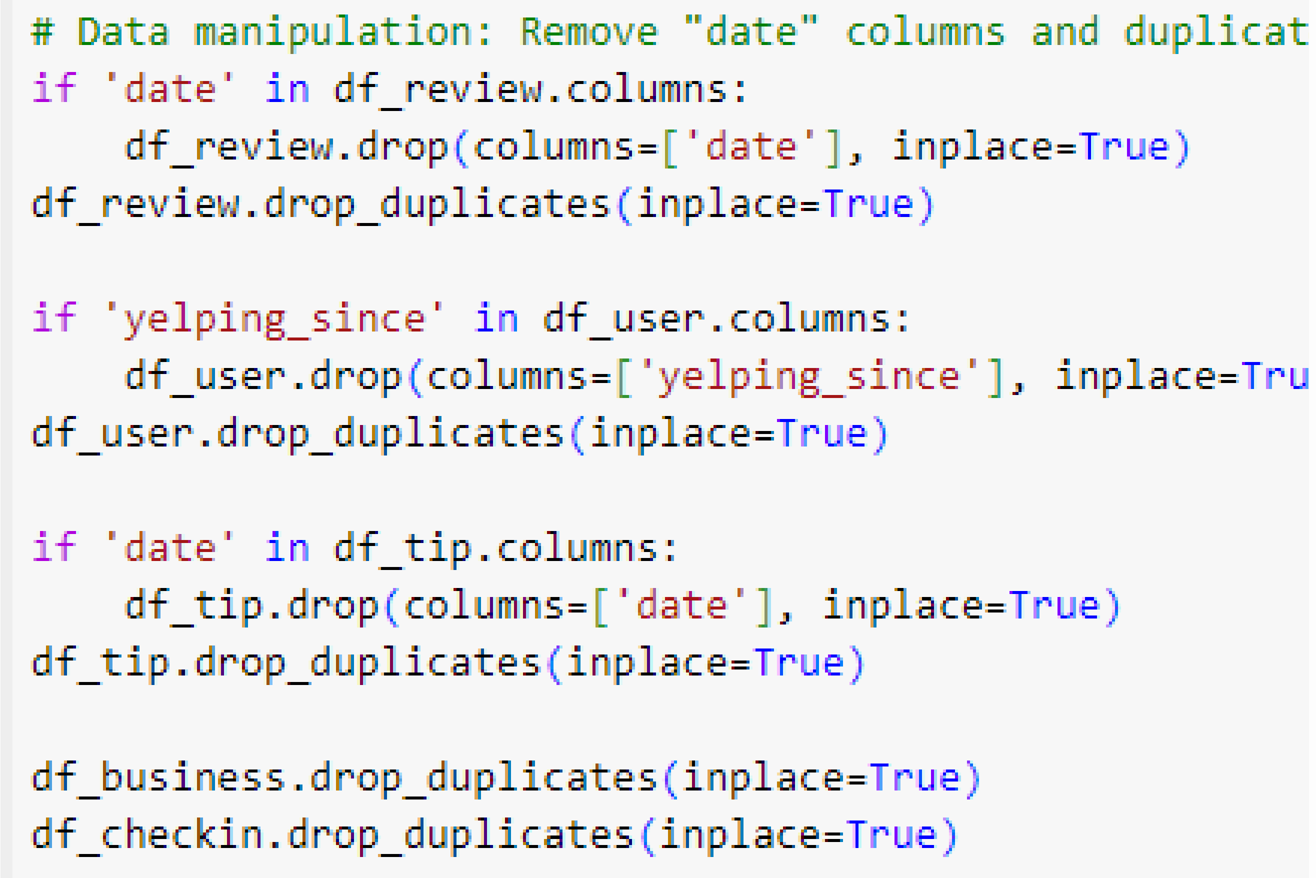


1. Convert to Pandas DataFrame
2. Data Manipulation in Python
3. Load Data into PostgreSQL
4. Execute Data Loading
5. Backup with .dump File

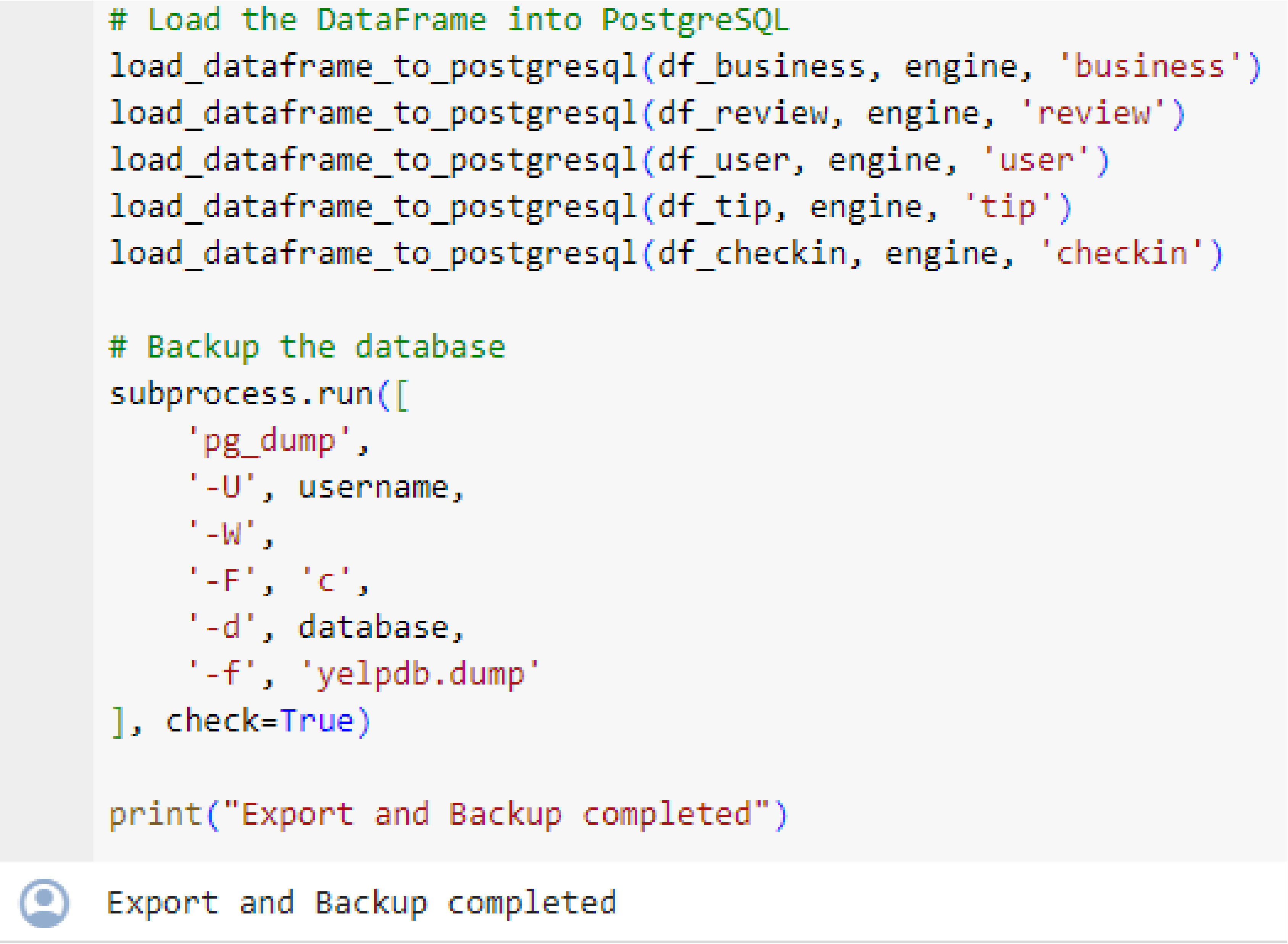
UPLOADING DATA INTO POSTGRESQL > CODE SNIPPETS



```python
# Function to load df to PostgreSQL
def load_dataframe_to_postgresql(df, engine, table_name, chunksize=5000):
    """
    Load a pandas DataFrame into a PostgreSQL table.
    """
    df.to_sql(table_name, engine, if_exists='append', index=False, chunksize=chunksize)


# Database credentials and connection
username = 'team1'
password = 'yelpdata'
host = 'localhost'
port = '5434'
database = 'yelpdb'
engine = create_engine(f'postgresql://{username}:{password}@{host}:{port}/{database}')
```



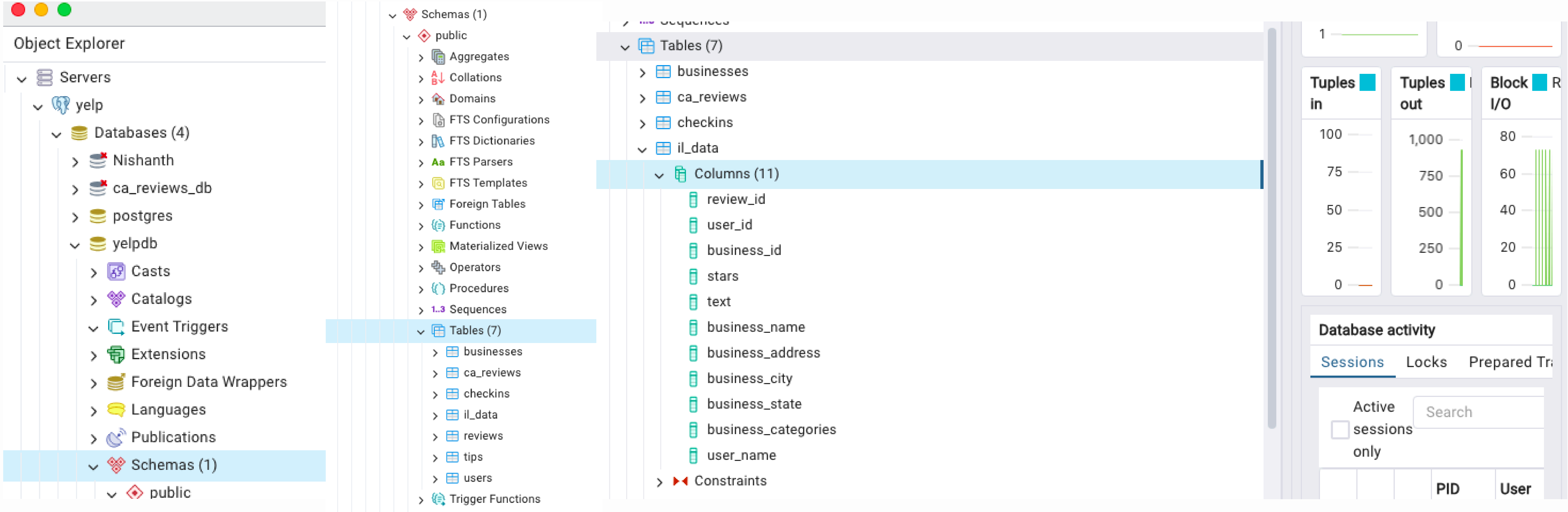
```python
# Data manipulation: Remove "date" columns and duplicat
if 'date' in df_review.columns:
    df_review.drop(columns=['date'], inplace=True)
df_review.drop_duplicates(inplace=True)

if 'yelping_since' in df_user.columns:
    df_user.drop(columns=['yelping_since'], inplace=Tru
df_user.drop_duplicates(inplace=True)

if 'date' in df_tip.columns:
    df_tip.drop(columns=['date'], inplace=True)
df_tip.drop_duplicates(inplace=True)

df_business.drop_duplicates(inplace=True)
df_checkin.drop_duplicates(inplace=True)
```



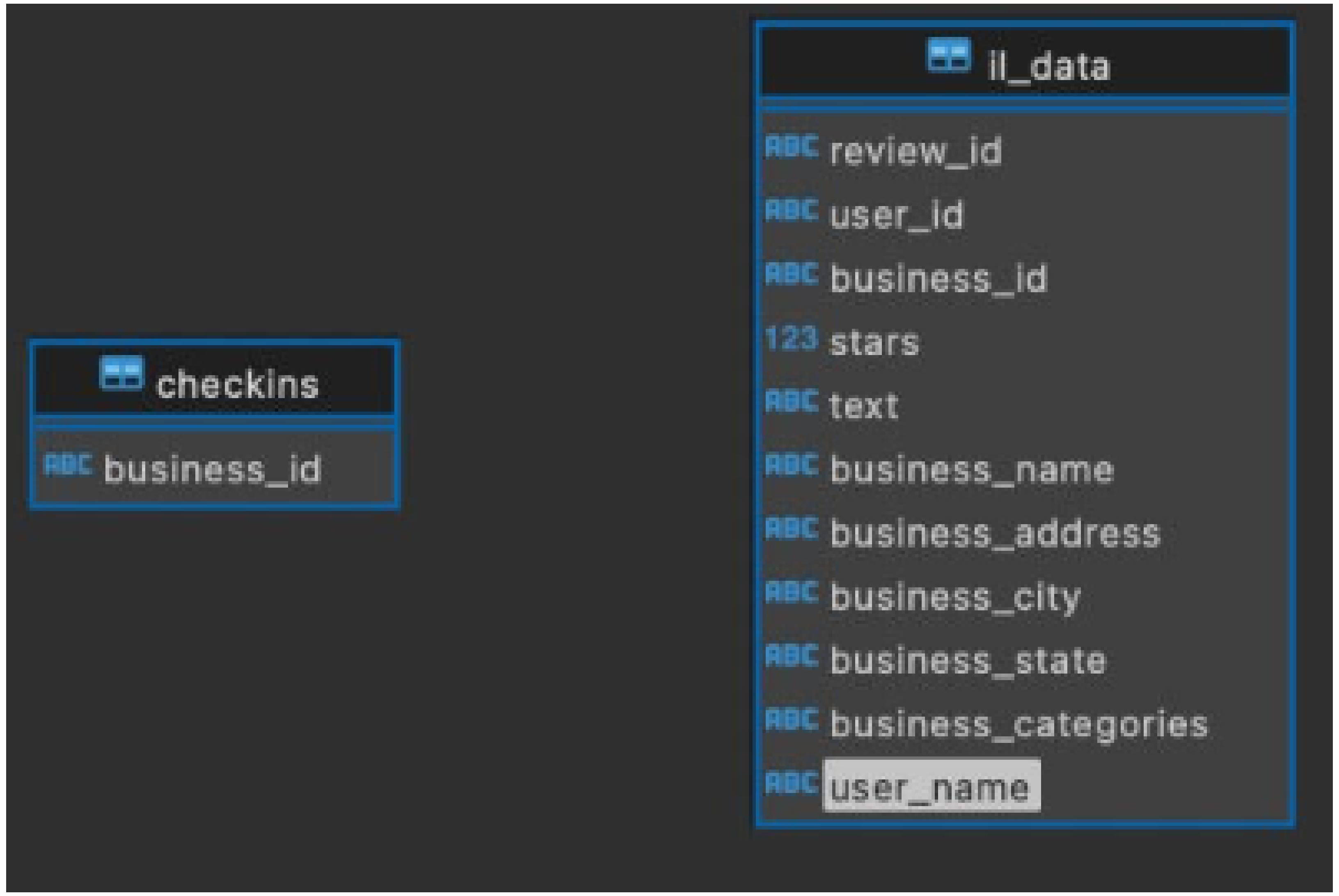
```python
# Load the DataFrame into PostgreSQL
load_dataframe_to_postgresql(df_business, engine, 'business')
load_dataframe_to_postgresql(df_review, engine, 'review')
load_dataframe_to_postgresql(df_user, engine, 'user')
load_dataframe_to_postgresql(df_tip, engine, 'tip')
load_dataframe_to_postgresql(df_checkin, engine, 'checkin')

# Backup the database
subprocess.run([
    'pg_dump',
    '-U', username,
    '-W',
    '-F', 'c',
    '-d', database,
    '-f', 'yelpdb.dump'
], check=True)

print("Export and Backup completed")
```

```
Export and Backup completed
```

UPLOADING DATA INTO POSTGRESQL > ON POSTGRESQL



ENTITY RELATIONAL(ER) DIAGRAM FROM DBEAVER

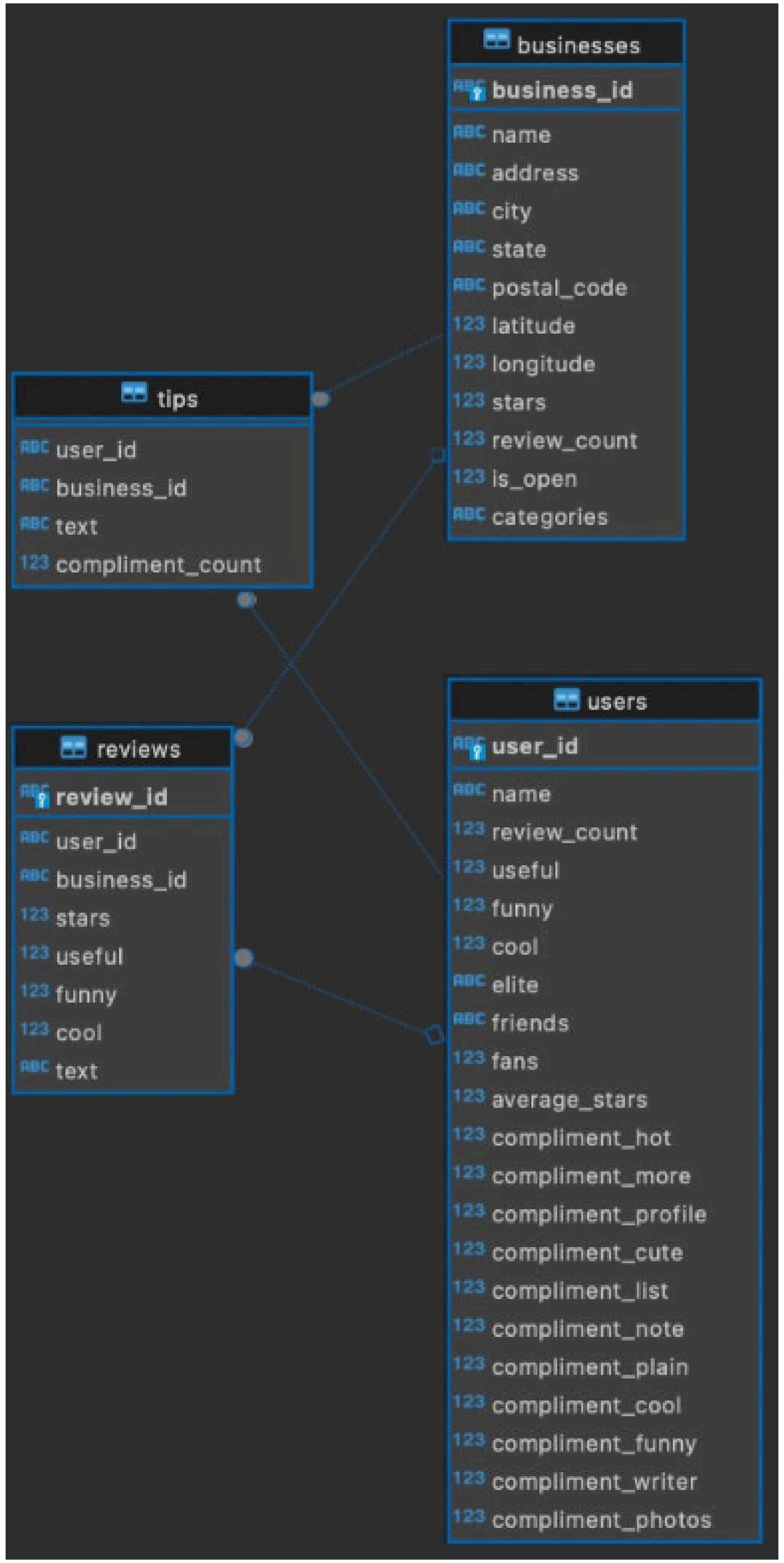


checkins

- ABC business_id

il_data

- ABC review_id
- ABC user_id
- ABC business_id
- 123 stars
- ABC text
- ABC business_name
- ABC business_address
- ABC business_city
- ABC business_state
- ABC business_categories
- ABC user_name



businesses

- ABC business_id
- ABC name
- ABC address
- ABC city
- ABC state
- ABC postal_code
- 123 latitude
- 123 longitude
- 123 stars
- 123 review_count
- 123 is_open
- ABC categories

tips

- ABC user_id
- ABC business_id
- ABC text
- 123 compliment_count

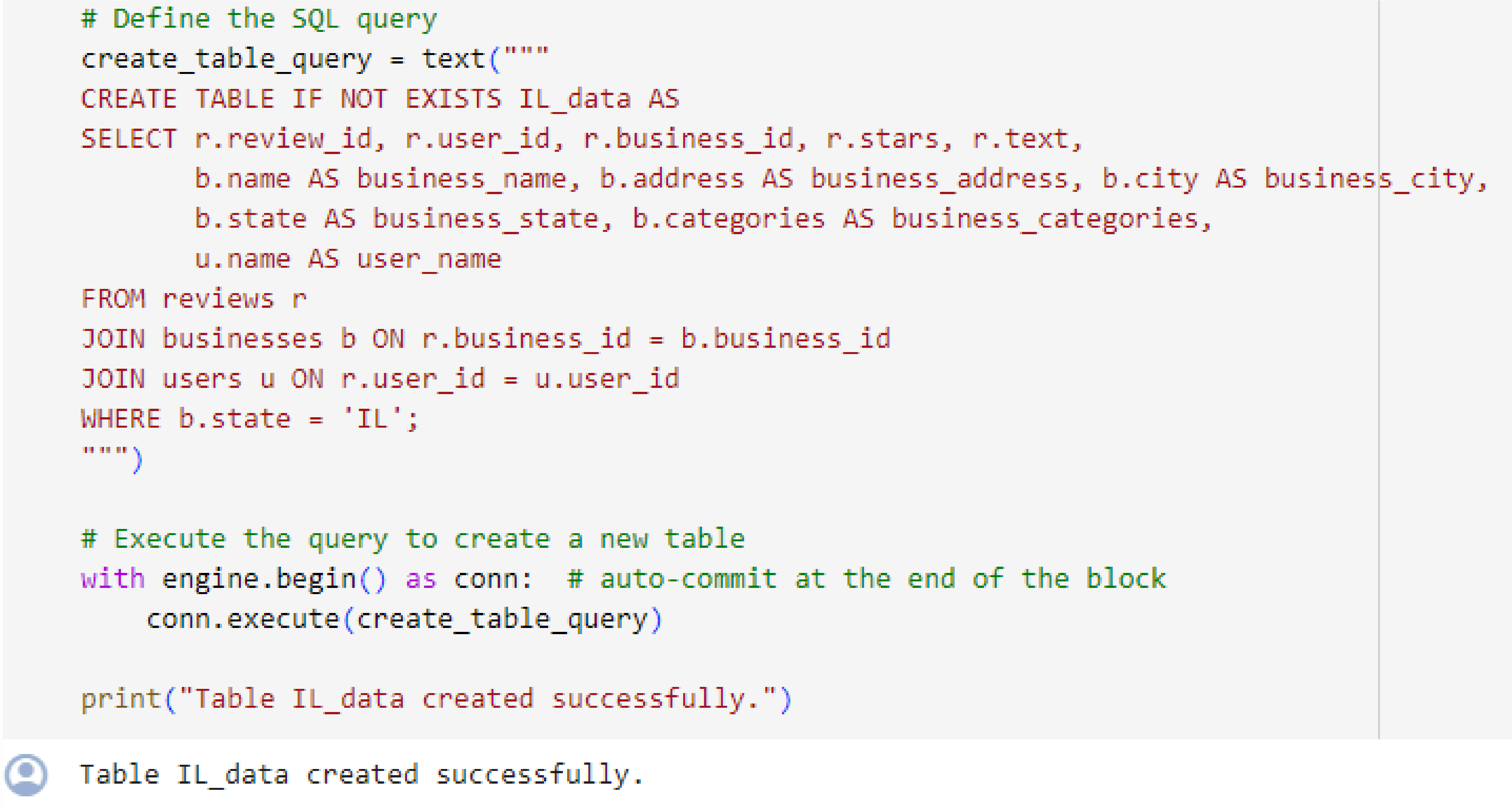
reviews

- ABC review_id
- ABC user_id
- ABC business_id
- 123 stars
- 123 useful
- 123 funny
- 123 cool
- ABC text

users

- ABC user_id
- ABC name
- 123 review_count
- 123 useful
- 123 funny
- 123 cool
- ABC elite
- ABC friends
- 123 fans
- 123 average_stars
- 123 compliment_hot
- 123 compliment_more
- 123 compliment_profile
- 123 compliment_cute
- 123 compliment_list
- 123 compliment_note
- 123 compliment_plain
- 123 compliment_cool
- 123 compliment_funny
- 123 compliment_writer
- 123 compliment_photos

CREATING A TABLE FOR ILLINOIS DATA IN SQL



```
# Define the SQL query
create_table_query = text("""
CREATE TABLE IF NOT EXISTS IL_data AS
SELECT r.review_id, r.user_id, r.business_id, r.stars, r.text,
       b.name AS business_name, b.address AS business_address, b.city AS business_city,
       b.state AS business_state, b.categories AS business_categories,
       u.name AS user_name
FROM reviews r
JOIN businesses b ON r.business_id = b.business_id
JOIN users u ON r.user_id = u.user_id
WHERE b.state = 'IL';
""")

# Execute the query to create a new table
with engine.begin() as conn:  # auto-commit at the end of the block
    conn.execute(create_table_query)

print("Table IL_data created successfully.")
```

```
Table IL_data created successfully.
```

LIMITATIONS AND PROBLEMS FACED

1. Data Scope and Size Limitations

- Computational Limits: Full dataset processing is resource-intensive, limiting real-time analysis.
- Geographic Focus: Limited to Illinois due to dataset size (*total 8GB of data*).

2. Machine Learning Model Constraints

- Model Simplicity: Models were basic, focusing on educational and demonstrative purposes.
- Feature Limitation: Use of advanced features and techniques was restricted by the project's educational scope.

3. Technical Challenges

- Tool Integration: Initially faced difficulties integrating SQL database operations with Python-based machine learning.
- Performance Optimization: Challenges in balancing model accuracy with computational efficiency.

REVIEW ANALYSIS WITH SENTIMENT AND QUALITY METRICS

1. Sentiment Extremity Calculation:

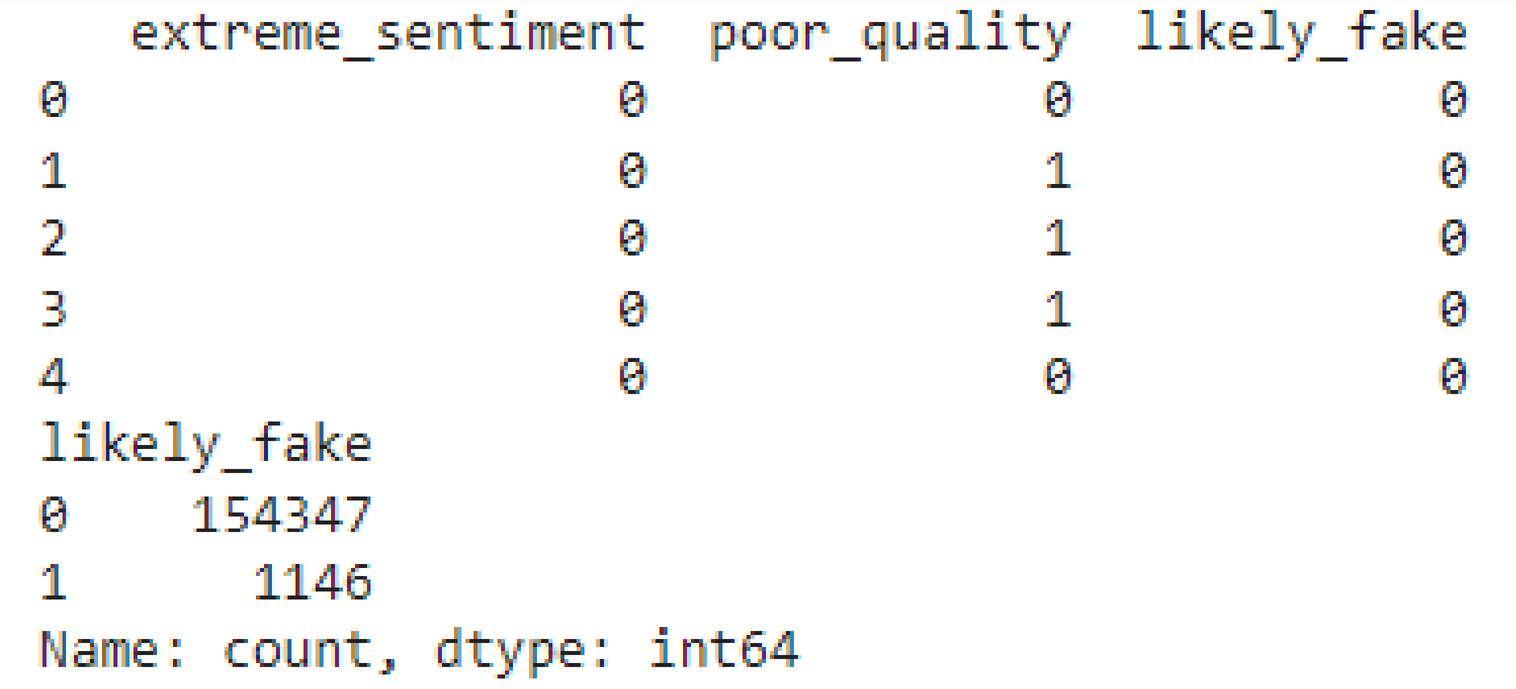
- Uses TextBlob to assess sentiment polarity and subjectivity.
- Assumptions: Flags reviews as extreme if polarity is greater than 0.8 or less than -0.8 combined with subjectivity over 0.5, or if a 5-star review has fewer than 5 words.

2. Review Quality Assessment:

- Assumptions: Identifies poor quality based on text length under 50 characters, word diversity less than 70% of total words, subjectivity over 0.8, or excessive punctuation.

3. Calculation of likely_fake:

- Combines flags from sentiment extremity and quality assessments.
- Assumptions: Reviews are marked as likely_fake if they score 2 or more based on these criteria, suggesting multiple signs of being fake or of low quality.



```
   extreme_sentiment  poor_quality  likely_fake
0                  0             0            0
1                  0             1            0
2                  0             1            0
3                  0             1            0
4                  0             0            0
likely_fake
0    154347
1      1146
Name: count, dtype: int64
```

DEVELOPMENT OF RANDOM FOREST CLASSIFIER

Model Overview

- Utilizes a Random Forest Classifier within a machine learning pipeline to identify reviews likely to be fake.

Model Pipeline Components

- Text Processing: Uses TfidfVectorizer with a maximum of 1000 features.
- Numerical Data Scaling: Applies StandardScaler to features like stars, extreme sentiment, and poor quality.
- Class Imbalance Handling: Integrates SMOTE for balancing the classes in training data.

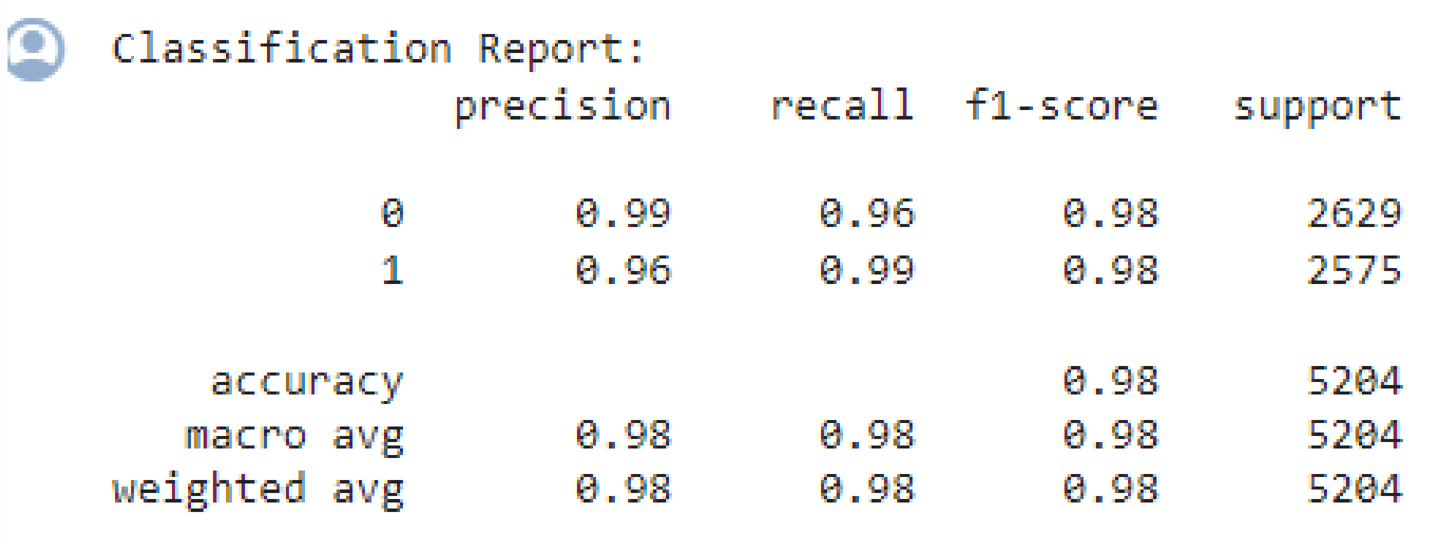
Random Forest Configuration

- Class Weight: Balanced
- Max Depth: 10
- Min Samples Split: 10
- Min Samples Leaf: 4
- Max Features: Square root of the number of features

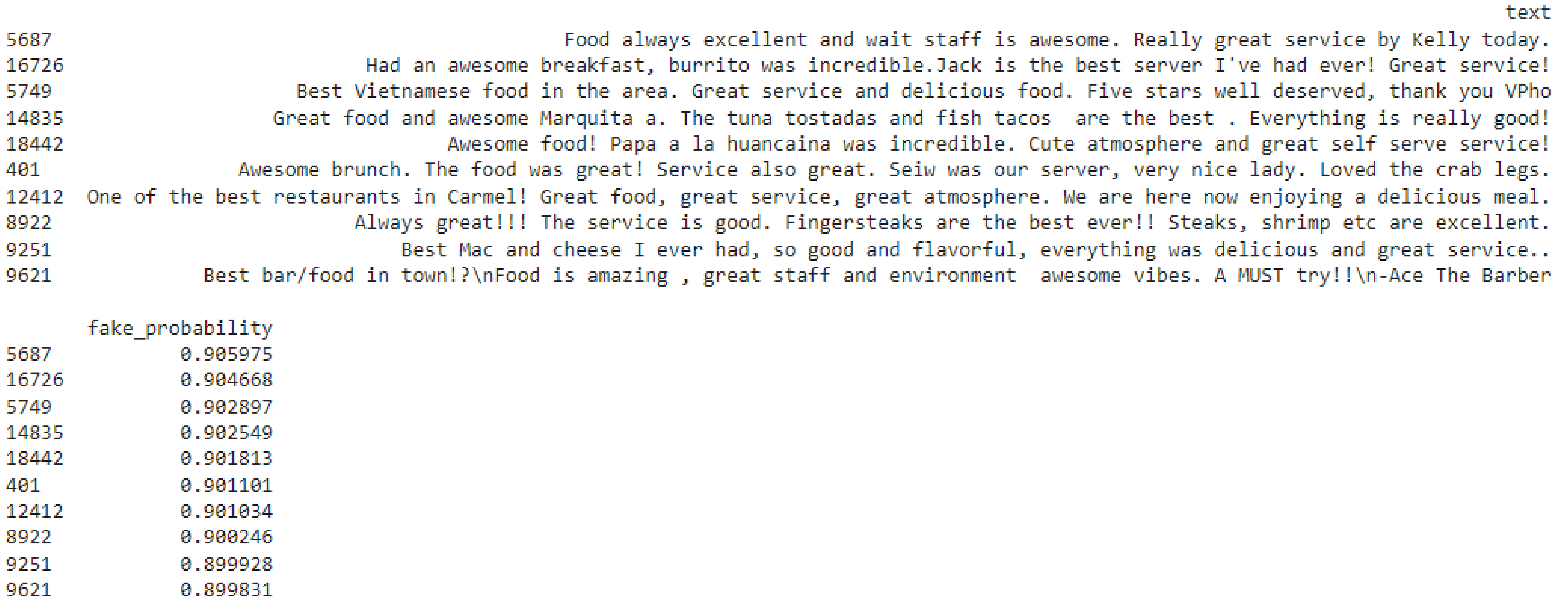
Key Takeaway

- The developed Random Forest Classifier effectively identifies potentially fake reviews by leveraging both textual and numerical data, offering a robust tool for enhancing review authenticity analysis.

RANDOM FOREST CLASSIFIER FINDINGS



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

           0       0.99      0.96      0.98      2629
           1       0.96      0.99      0.98      2575

    accuracy                           0.98      5204
   macro avg       0.98      0.98      0.98      5204
weighted avg       0.98      0.98      0.98      5204
```



```
                                                                                                                   text
5687                                              Food always excellent and wait staff is awesome. Really great service by Kelly today.
16726                                        Had an awesome breakfast, burrito was incredible.Jack is the best server I've had ever! Great service!
5749                                       Best Vietnamese food in the area. Great service and delicious food. Five stars well deserved, thank you VPho
14835                                    Great food and awesome Marquita a. The tuna tostadas and fish tacos  are the best . Everything is really good!
18442                                             Awesome food! Papa a la huancaina was incredible. Cute atmosphere and great self serve service!
401                                    Awesome brunch. The food was great! Service also great. Seiw was our server, very nice lady. Loved the crab legs.
12412  One of the best restaurants in Carmel! Great food, great service, great atmosphere. We are here now enjoying a delicious meal.
8922                                           Always great!!! The service is good. Fingersteaks are the best ever!! Steaks, shrimp etc are excellent.
9251                                               Best Mac and cheese I ever had, so good and flavorful, everything was delicious and great service..
9621                                  Best bar/food in town!?\nFood is amazing , great staff and environment  awesome vibes. A MUST try!!\n-Ace The Barber

       fake_probability
5687           0.905975
16726          0.904668
5749           0.902897
14835          0.902549
18442          0.901813
401            0.901101
12412          0.901034
8922           0.900246
9251           0.899928
9621           0.899831
```

DEVELOPMENT OF LSTM MODEL

Model Overview

- Develops a Long Short-Term Memory (LSTM) classifier to identify fake reviews by analyzing both textual and numerical features.

Model Components

- Text Processing: Utilizes a Tokenizer to convert text to sequences and pads them to a uniform length for LSTM processing.
- Numerical Data Processing: Standardizes features such as star ratings and sentiment indicators using StandardScaler.

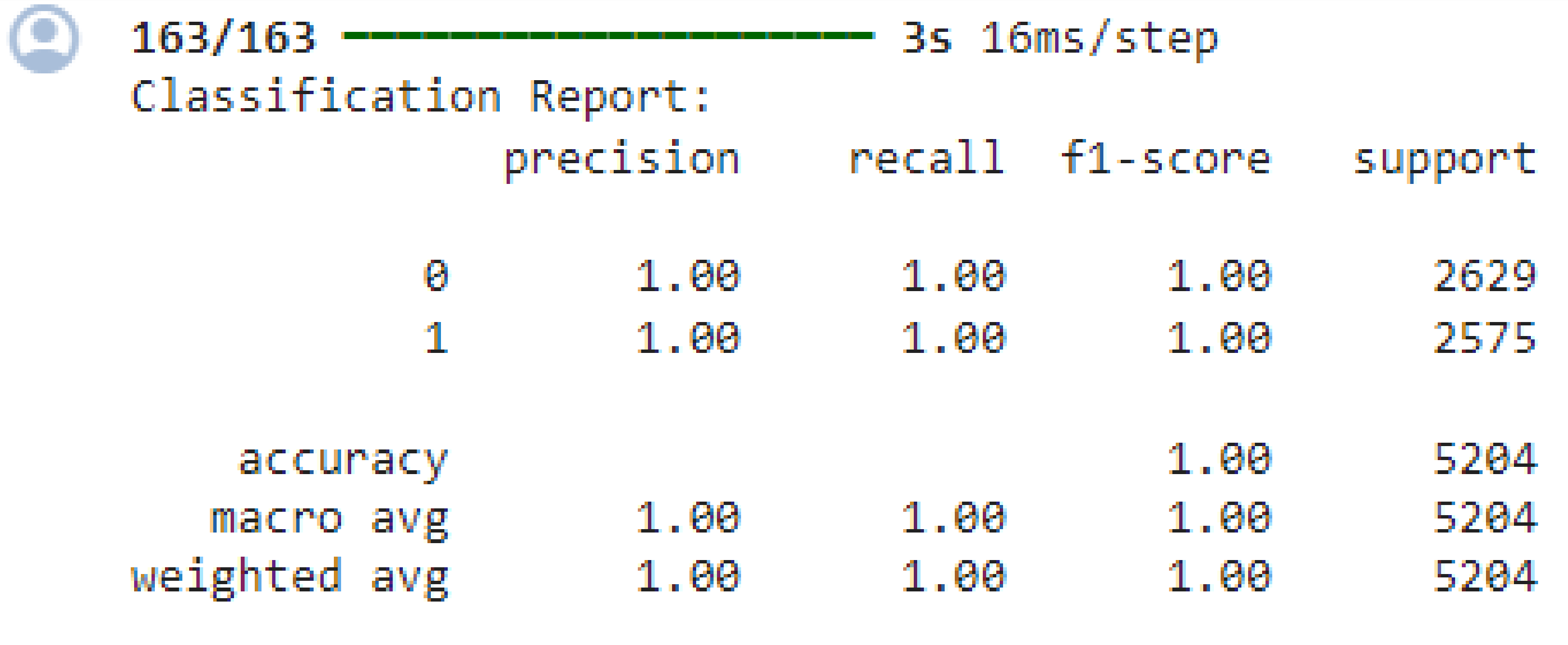
LSTM Configuration

- Architecture: Consists of an Embedding layer, a Spatial Dropout layer, an LSTM layer, and a Dense output layer.
- Parameters: Embedding dimension of 128, LSTM units of 100, and dropout rates of 0.2 for both LSTM and recurrent units.
- Optimization: Uses Adam optimizer and binary cross-entropy loss function, tracking accuracy, precision, and recall.

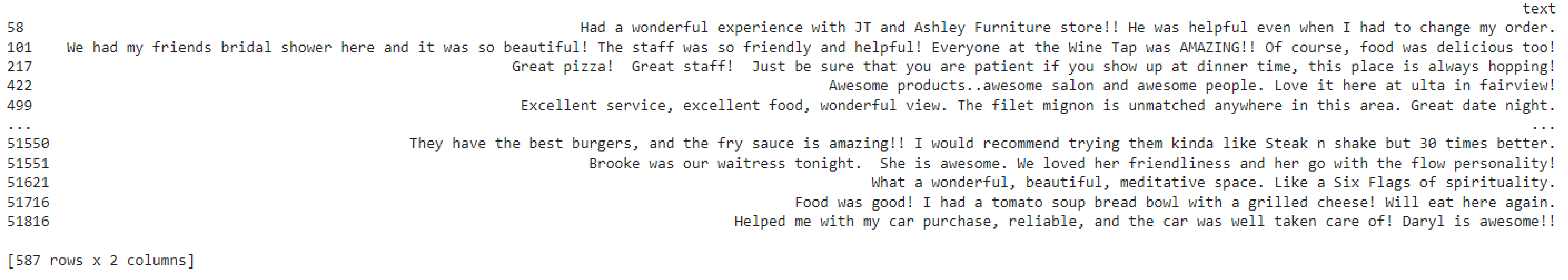
Key Takeaway

- The LSTM classifier robustly handles sequential text data and additional features to effectively pinpoint likely fake reviews, proving crucial for maintaining integrity in review platforms.

LSTM MODEL FINDINGS



```
163/163 ━━━━━━━━━━━━━━━━━━━━ 3s 16ms/step
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2629
           1       1.00      1.00      1.00      2575

    accuracy                           1.00      5204
   macro avg       1.00      1.00      1.00      5204
weighted avg       1.00      1.00      1.00      5204
```



```
                                                                                                                                                                   text
58                                                                 Had a wonderful experience with JT and Ashley Furniture store!! He was helpful even when I had to change my order.
101    We had my friends bridal shower here and it was so beautiful! The staff was so friendly and helpful! Everyone at the Wine Tap was AMAZING!! Of course, food was delicious too!
217                                                         Great pizza!  Great staff!  Just be sure that you are patient if you show up at dinner time, this place is always hopping!
422                                                                                       Awesome products..awesome salon and awesome people. Love it here at ulta in fairview!
499                                                          Excellent service, excellent food, wonderful view. The filet mignon is unmatched anywhere in this area. Great date night.
...                                                                                                                                                                         ...
51550                                               They have the best burgers, and the fry sauce is amazing!! I would recommend trying them kinda like Steak n shake but 30 times better.
51551                                                                    Brooke was our waitress tonight.  She is awesome. We loved her friendliness and her go with the flow personality!
51621                                                                                                  What a wonderful, beautiful, meditative space. Like a Six Flags of spirituality.
51716                                                                                         Food was good! I had a tomato soup bread bowl with a grilled cheese! Will eat here again.
51816                                                                                  Helped me with my car purchase, reliable, and the car was well taken care of! Daryl is awesome!!

[587 rows x 2 columns]
```

INTERACTIVE DEMONSTRATION OF LSTM FAKE REVIEW DETECTION

Demo Components

- User Interaction: Allows users to enter their own review text.
- Sentiment and Quality Analysis: Automatically calculates sentiment extremity and review quality.
- Data Preprocessing: Applies tokenization, padding, and scaling to prepare user input for prediction.
- Prediction Execution: Uses the trained LSTM model to determine if the review is likely fake.
- Result Display: Outputs whether the entered review is classified as 'Fake' or 'Not Fake'.

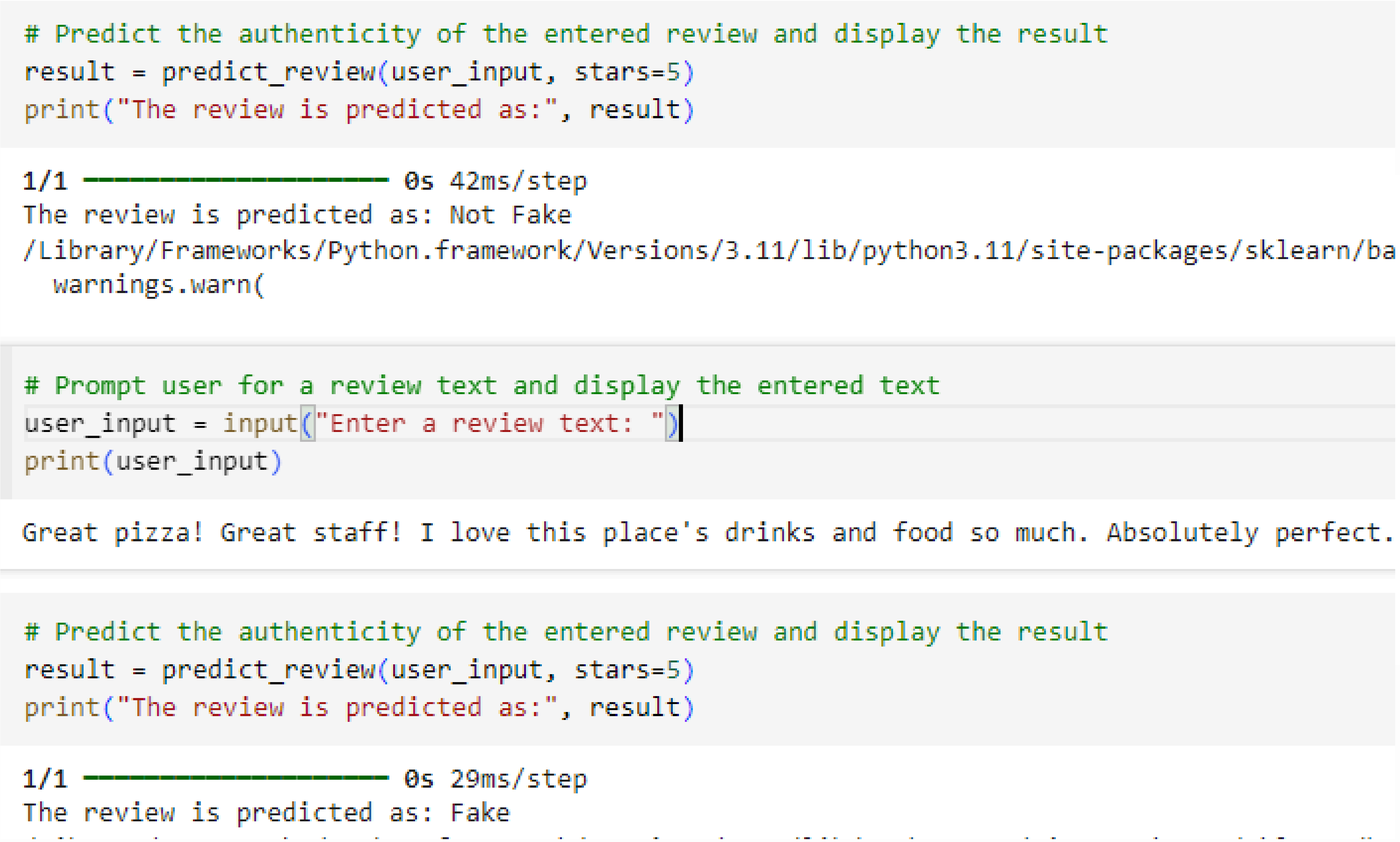
Sample Interaction

1. User is prompted to input a review.
2. System processes the review through the LSTM model.
3. The system displays the prediction outcome.

Extended Functionality

- Review Retrieval: Fetches reviews from the database for specified states to analyze and predict.
- Fake Review Analysis: Identifies and displays reviews predicted as fake.
- Word Cloud Generation: Visualizes common words from fake reviews to highlight prevalent themes or expressions.

INTERACTIVE DEMONSTRATION OF LSTM FAKE REVIEW DETECTION



```
# Predict the authenticity of the entered review and display the result
result = predict_review(user_input, stars=5)
print("The review is predicted as:", result)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
The review is predicted as: Not Fake
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-packages/sklearn/ba
  warnings.warn(
```

```
# Prompt user for a review text and display the entered text
user_input = input("Enter a review text: ")
print(user_input)
```

```
Great pizza! Great staff! I love this place's drinks and food so much. Absolutely perfect.
```

```
# Predict the authenticity of the entered review and display the result
result = predict_review(user_input, stars=5)
print("The review is predicted as:", result)
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 29ms/step
The review is predicted as: Fake
```

great great
service service
food food
best best
delicious delicious
place place
good good
awesome awesome
excellent excellent
amazing amazing
wonderful wonderful
always always
customer customer
atmosphere atmosphere
everything everything
pizza pizza
time time
staff staff
restaurant restaurant
selection selection
drinks drinks
terrible terrible
coffee coffee
try try
cheese cheese
menu menu
location location
breakfast breakfast
one one
ordered ordered
dinner dinner
family family
loved loved
never never
really really
recommend recommend
nice nice
go go
thank thank
well well
went went
experience experience
will will
horrible horrible
happy happy
order order
every every
bar bar
got got
lunch lunch
perfect perfect
must must
love love
town town
made made
beautiful beautiful
even even
beer beer
spot spot
people people
worst worst
salad salad
work work
job job
make make
us us

CONCLUSION AND FUTURE DIRECTIONS

Database Management Integration

- PostgreSQL Utilization: Managed large datasets with PostgreSQL, employing a specialized schema for Yelp data to support complex SQL queries.
- Localized Data Analysis: Developed a PostgreSQL table for Illinois data to enhance geographical analysis and database efficiency.
- Database Operations: Executed essential operations such as data insertion, updates, and backups using pg_dump for data integrity.

Machine Learning Integration

- Model Development: Created models to analyze textual and numerical data from the database, effectively predicting fake reviews.
- Real-Time Analysis: Integrated models within the database framework to dynamically classify reviews from PostgreSQL, showcasing real-world machine learning applications in database management.

Enhancing Project Capabilities

Advanced Model Development:

- Upgrade to more sophisticated machine learning algorithms to handle larger datasets and enhance insight extraction.
- Introduce an authentic fake review dataset for accurate model training and validation.

THANK YOU

ANY QUESTIONS?