RECIPE RECOMMENDATION SYSTEM

NLP Group 1:

Wen-chi Tseng
Vincent Lee
Hao-Yun Tseng
Rita Rong Nie

CONTENT

PROJECT OVERVIEW

- Growing demand for personalized online recipe recommendations
- Significance of matching user preferences (taste, ingredients, time) with suitable recipes
- Objective: Develop a system to recommend recipes based on user preferences.

DATASET

Sourced from ~125,000 online recipes, including titles, ingredients, instructions, advertisement and images

Dataset: https://eightportions.com/datasets/Recipes/

PROBLEM STATEMENT

- Challenge: Recommending recipes that align with user-defined taste, preferred ingredients, and cooking time constraints
- Goal: To create a recommendation system that understands and adapts to individual user preferences

DATA PRE-PROCESSING

1. Data Cleaning

- Remove Advertisements
- Data Transformation
- Removed duplicated and null data
- Now we have 82,267 recipes ready to use

```python
# Open the JSON file in read mode
with open('recipes_raw_nosource_ar.json', 'r') as json_file:
    data = json.load(json_file)

# Remove "ADVERTISEMENT" from the JSON data
def remove_advertisement(obj):
    if isinstance(obj, str):
        return obj.replace(" ADVERTISEMENT", "")
    elif isinstance(obj, list):
        return [remove_advertisement(item) for item in obj]
    elif isinstance(obj, dict):
        return {key: remove_advertisement(value) for key, value in obj.items()}
    else:
        return obj

modified_data = remove_advertisement(data)

# Write the modified data back to the JSON file
with open('recipes_raw_nosource_ar_modified.json', 'w') as json_file:
    json.dump(modified_data, json_file, indent=4)
```

```python
# Load the modified json file
df = pd.DataFrame(modified_data)
df = df.transpose()
df.reset_index(drop=True, inplace=True)


df.head()
```

	title	ingredients	instructions	picture_link
0	Slow Cooker Chicken and Dumplings	[4 skinless, boneless chicken breast halves, 2...	Place the chicken, butter, soup, and onion in ...	55lznCYBbs2mT8BTx6BTkLhynGHzM.S
1	Awesome Slow Cooker Pot Roast	[2 (10.75 ounce) cans condensed cream of mushr...	In a slow cooker, mix cream of mushroom soup, ...	QyrvGdGNMBA2lDdciY0FjKu.77MM0Oe
2	Brown Sugar Meatloaf	[1/2 cup packed brown sugar, 1/2 cup ketchup, ...	Preheat oven to 350 degrees F (175 degrees C)....	LVW1DI0vtlCrpAhNSEQysE9i/7rJG56
3	Best Chocolate Chip Cookies	[1 cup butter, softened, 1 cup white sugar, 1 ...	Preheat oven to 350 degrees F (175 degrees C)....	0SO5kdWOV94j6EfAVwMMYRM3yNN8eRi
4	Homemade Mac and Cheese Casserole	[8 ounces whole wheat rotini pasta, 3 cups fre...	Preheat oven to 350 degrees F. Line a 2-quart ...	YCnbhplMgiraW4rUXcybgSEZinSgljm

2. Feature Extraction

- Lemmanization and spaCy for Regular Expressions
- Extracting Cooking Times

```
<class 'pandas.core.frame.DataFrame'>
Index: 39522 entries, 0 to 39801
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   title         39522 non-null  object
 1   ingredients   39522 non-null  object
 2   instructions  39522 non-null  object
 3   picture_link  39522 non-null  object
 4   cooking time  39522 non-null  object
dtypes: object(5)
memory usage: 1.8+ MB
None
```

```
                                    title  \
0          Slow Cooker Chicken and Dumplings
1              Awesome Slow Cooker Pot Roast
2                       Brown Sugar Meatloaf
3                Best Chocolate Chip Cookies
4          Homemade Mac and Cheese Casserole
...                                      ...
39797                Thai-Indian Veggie Soup
39798  Coconut Milk-Free Panang Curry Chicken
39799                      Cooked Cold Salad
39800                Easy Eggnog Creme Brulee
39801            Super Power Stovetop Granola

                                             ingredients  \
0      [4 skinless, boneless chicken breast halves, 2...
1      [2 (10.75 ounce) cans condensed cream of mushr...
2      [1/2 cup packed brown sugar, 1/2 cup ketchup, ...
3      [1 cup butter, softened, 1 cup white sugar, 1 ...
4      [8 ounces whole wheat rotini pasta, 3 cups fre...
...                                                  ...
39797  [2 teaspoons olive oil, 1/4 cup minced fresh g...
39798  [2 cups light cream, 1/4 teaspoon coconut extr...
39799  [3 tablespoons bacon grease, 2 cups shredded B...
39800  [4 egg yolks, 1 tablespoon white sugar, 1 cup ...
39801  [1/4 cup canola oil, 3 cups quick-cooking oats...
...
39800                       [1 hour, 4 hours, 1 minute]
39801  [7 minutes, 3 minutes, 5 minutes, 5 minutes, 5...
```

MODEL BUILDING Part 1

- Used Pipeline to find the optimal parameters

```
# Define TF-IDF parameters grid
param_grid = {
    'tfidf__max_features': [1000, 5000, 10000],
    'tfidf__ngram_range': [(1, 1), (1, 2), (1, 3)],
    'svd__n_components': [50, 100, 200]  # Truncated SVD parameters
}

# Define pipeline
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('svd', TruncatedSVD())
])

# Perform Grid Search with Cross-Validation
grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='accuracy')
grid_search.fit(combined_text)

# Get best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Apply best parameters to the pipeline
pipeline.set_params(**best_params)

# Fit the pipeline
pipeline.fit(combined_text)

# Extract the TF-IDF matrix and Truncated SVD matrix
tfidf_matrix = pipeline.named_steps['tfidf'].transform(combined_text)
svd_matrix = pipeline.named_steps['svd'].transform(tfidf_matrix)
```

- TF-IDF Vectorization

Vectors: Titles, Ingredients, Instructions, Cooking Times transformed into numerical vectors.

```
# Feature Extraction using TF-IDF
tfidf = TfidfVectorizer(stop_words='english', max_features=1000)
tfidf_matrix = tfidf.fit_transform(combined_text)

# Dimensionality Reduction using TruncatedSVD
svd = TruncatedSVD(n_components=50)
svd_matrix = svd.fit_transform(tfidf_matrix)
```

```
# Cosine Similarity Calculation
def get_recipe_recommendations(query, tfidf_vectorizer, svd_model, data):
    # Preprocess the query
    query = preprocess_text(query)
    query_vector = tfidf_vectorizer.transform([query])
    query_svd = svd_model.transform(query_vector)

    # Calculate cosine similarity
    similarity_scores = cosine_similarity(query_svd, svd_matrix)

    # Get index of the most similar recipe
    top_recipe_index = np.argmax(similarity_scores)

    # Return the most similar recipe
    return data.iloc[top_recipe_index]
```

MODEL BUILDING Part 2

- Word2Vec

  Trained a Word2Vec model on preprocessed text data to learn word embeddings.

  Word2Vec represents words in a high-dimensional vector space where words with similar contexts are closer to each other.

- Recipe Embeddings

  Used Embedding to match words with similar meanings in recipes.

  Compute a vector representation for each recipe by averaging the word embeddings of its constituent words. This provides a numerical representation of the recipe's content.

```
# Train Word2Vec model
word2vec_model = Word2Vec(sentences=df['preprocessed_text'], vector_size=100, window=5, min_count=1, workers=4)

# Function to generate recipe embeddings
def generate_recipe_embedding(tokens, model):
    embeddings = [model.wv[word] for word in tokens if word in model.wv]
    if embeddings:
        return np.mean(embeddings, axis=0)
    else:
        return np.zeros(model.vector_size)

# Generate recipe embeddings for all recipes
df['recipe_embedding'] = df['preprocessed_text'].apply(lambda x: generate_recipe_embedding(x, word2vec_model))
```

CHATBOT TEST – TF-IDF SVD

I want to make a…🤔

Lunch with chicken breast and broccoli under 20 minutes.

Title: Salsa Simmered Chicken

Ingredients:

('6 skinless, boneless chicken breast halves', 'salt and ground black pepper to taste', '2 tablespoons olive oil', '1 (15 ounce) jar mild picante salsa', '1 (14 ounce) can chicken broth', ' ')

Instructions:

Season chicken breasts all over with salt and black pepper. Heat oil in a large skillet over medium-high heat. Cook chicken breasts in hot oil until browned, 4 to 6 minutes per side. Pour salsa and chicken broth over chicken, bring mixture to a boil, reduce heat to medium-low, and simmer for 15 minutes. Turn chicken over, stir salsa mixture, and continue to simmer until chicken is tender and sauce is slightly thickened, about 15 minutes more. An instant-read thermometer inserted into the center of the chicken should read at least 165 degrees F (74 degrees C).

Time Duration: 6 minutes, 15 minutes, 15 minutes

CHATBOT TEST – Word2Vec

I want to make a…🤔

Lunch with chicken breast and broccoli under 20 minutes.

Title: Stuffed Tomato Basil Chicken

Ingredients:

('4 (6 ounce) boneless, skinless chicken breasts', '1/2 (12 ounce) bottle garlic and herb marinade', '16 fresh basil leaves', '1 large tomato, thinly sliced', '4 slices provolone cheese', '12 slices bacon', '1/4 cup freshly grated Parmesan', ' ')

Instructions:

Place chicken breasts on a cutting board. With a sharp knife, slice chicken breasts horizontally, without slicing them completely in half. Open the chicken breasts like a book. Place chicken and marinade into a large resealable plastic bag. Refrigerate for 30 minutes. Preheat oven to 500 degrees F (260 degrees C). Place opened chicken breasts on a broiler pan. Place 4 basil leaves on the bottom half of each chicken breast. Top each with 2 or 3 tomato slices and 1 slice of cheese, and fold over top half of chicken (if necessary, fasten with toothpicks). Wrap 3 slices bacon around each chicken breast. Cook in preheated oven for 15 minutes. Turn chicken, and cook 15 minutes more. Remove from oven, and sprinkle chicken with Parmesan. Return to oven, and cook until cheese is melted, about 2 to 3 minutes.

Time Duration: 30 minutes, 15 minutes, 15 minutes, 3 minutes

CONCLUSION

What We Did:

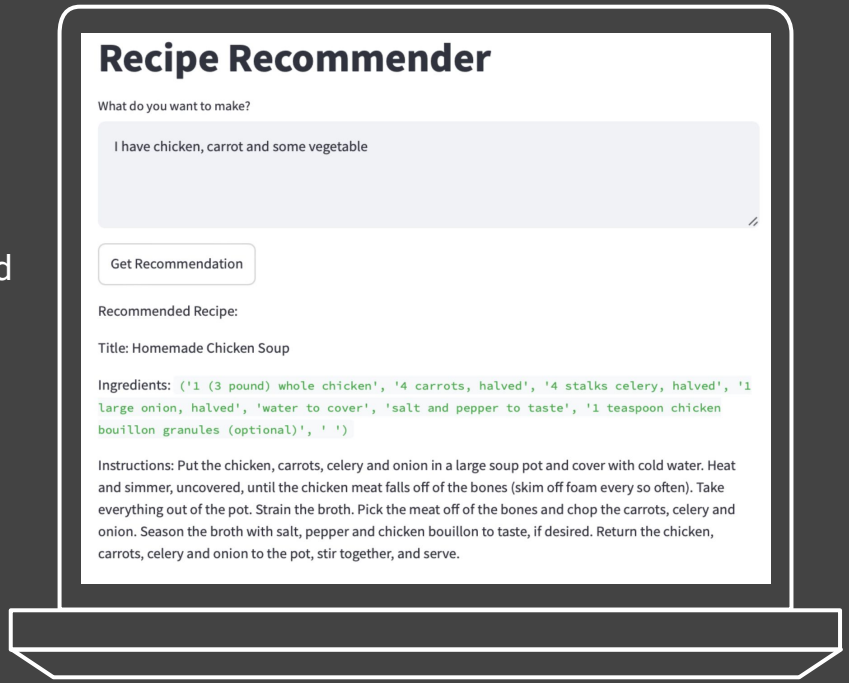
- Data Preprocessing:
  - Cleaned ~125,000 recipes, eliminated ads
- NLP & ML:
  - Applied TF-IDF, Truncated SVD, Word2Vec, Word Embedding, and Cosine Similarity
- User Recommendations:
  - Used weighted scoring for precise matches

Challenge:

- Managed data diversity and volume, fine-tuned weighting, and improved relevance

Future Directions:

- Aim for more personalized recommendations, expand recipe diversity, and enhance UI/UX



Thank You