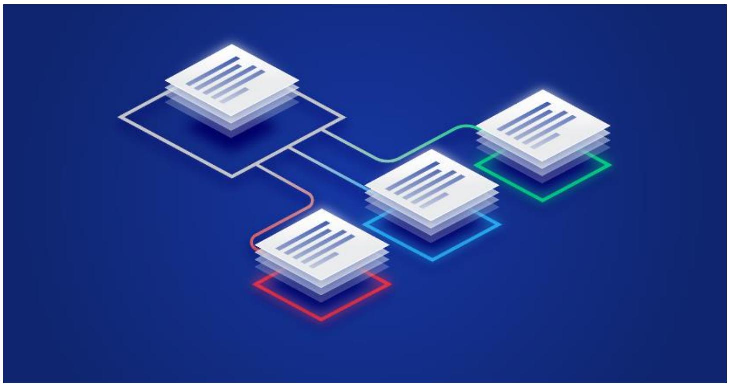


Food Cuisine

(Text Classifier)

Machine Learning text classifier model to automatically predict the cuisine of any food item.

Problem Identification

- We have a dataset of over 500,000 menu items served by 5,000+ restaurants across cities like Hyderabad, Bangalore, Chennai, etc.
- Consider the problem of predicting the cuisine of a restaurant menu item directly using its listed name on the restaurant's front end.
- Our aim is to identify the best classification technique that can help food databases easily categorize items into specific cuisine type to overcome labour intensive data entry.
- We intend to build a robust machine learning model to automatically predict a cuisine category for any food item inputted. This will help with problems like user taste profiling, recommender systems based on user segmentation and restaurant tagging. data

Problem Statement

Create a corpus of cuisines and classify food items into their respective category. Further, generate training data and build a robust Machine Learning text classifier that inputs a food item name and predicts a suitable Cuisine type that the item belongs to.

Data Used

- Menu Item names – 500,000+ data points
  - **name –** menu item name used as model input
  - **description –** menu item description (available for a few)
- Cuisine Corpus – scraped from Wikipedia and re-structured for our use-case

Data Wrangling

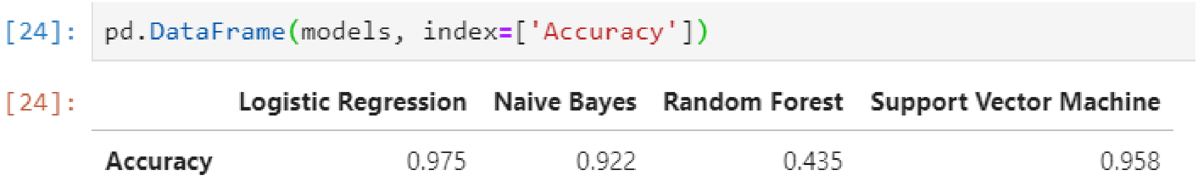
- Data pre-processing is started by specifying the multiple classes the data would be categorized into, i.e., 12 unique cuisines.
- Develop a basic corpus of various food items and ingredients under each cuisine. This data was referred from multiple data sources like Wikipedia, Kaggle, etc.
- After running a brief analysis on our item names, we deduce that in most of the data points, the last word in any item name is the key feature to predict which cuisine it belongs to.

- We used the NLTK library's word-tokenizer function to extract the last word from names, and eliminated stopwords that could impact our model's performance.
    - During the model building phase, it was noted that the elimination of certain stopwords was resulting in a greatly improved accuracy.
- The last step before training our model, is to convert the modified names from string values to a machine level sparse matrix using the TF-IDF vectorizer.

Approach and Machine Learning Models

- As the Cuisine labels to be predicted were predefined, various Supervised Text Classification models were identified to run our test cases and eventually select the best performing one.
- Data was split into training and testing datasets in an 80/20 ratio.
- Pipeline was designed that first passes data to the TF-IDF vectorizer followed by the specified machine learning algorithm.
- ML Models used:
    - Multinomial Naïve Bayes
    - Logistic Regression
    - Random Forest
    - Support Vector Machine

- Model Performance:



```
[24]: pd.DataFrame(models, index=['Accuracy'])
```

[24]:

	Logistic Regression	Naive Bayes	Random Forest	Support Vector Machine
Accuracy	0.975	0.922	0.435	0.958

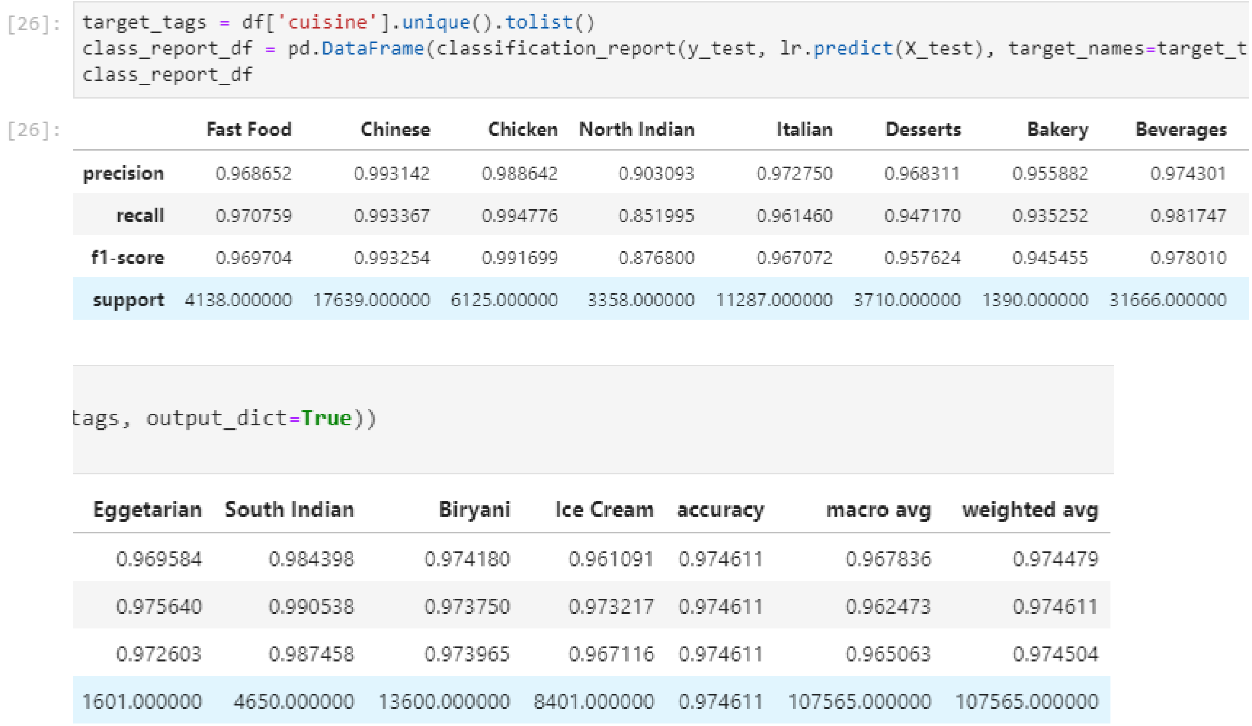
Final Feature Representation

Model chosen: Logistic Regression

- **Accuracy: 97.5%**
- Precision: 96.25 %
- Recall: 96.78 %
- **<u>Hyperparameter tuning</u>**:
    - Regularization Strength (C) : 1e5 (~100000)

Future Scope

Classification report of each category for the Logistic Regression Model



```
[26]: target_tags = df['cuisine'].unique().tolist()
      class_report_df = pd.DataFrame(classification_report(y_test, lr.predict(X_test), target_names=target_tags, output_dict=True))
      class_report_df
```

[26]:

	Fast Food	Chinese	Chicken	North Indian	Italian	Desserts	Bakery	Beverages	Eggetarian	South Indian	Biryani	Ice Cream	accuracy	macro avg	weighted avg
precision	0.968652	0.993142	0.988642	0.903093	0.972750	0.968311	0.955882	0.974301	0.969584	0.984398	0.974180	0.961091	0.974611	0.967836	0.974479
recall	0.970759	0.993367	0.994776	0.851995	0.961460	0.947170	0.935252	0.981747	0.975640	0.990538	0.973750	0.973217	0.974611	0.962473	0.974611
f1-score	0.969704	0.993254	0.991699	0.876800	0.967072	0.957624	0.945455	0.978010	0.972603	0.987458	0.973965	0.967116	0.974611	0.965063	0.974504
support	4138.000000	17639.000000	6125.000000	3358.000000	11287.000000	3710.000000	1390.000000	31666.000000	1601.000000	4650.000000	13600.000000	8401.000000	0.974611	107565.000000	107565.000000

- From the above multi-class classification report of our model, we can see that the cuisine category ‘North Indian’ hasn’t been performing well in our text classifier. We may be able to improve the results under that category by tuning the TF-IDF vectorizer by applying a Grid Search or hit-and-run method.
- Using the cuisine data, we can generate real-time taste profile of people that can assist user segmentation.
- Trend analysis can be performed using the cuisine data, and that could contribute to a user recommender system.

Conclusion

We successfully processed the raw data to clean and transform it into training data for feeding to the text classifier. We built various high performing machine learning models. After comparing their performances, we eventually selected the best model as Logistic Regression with a predictive accuracy of 97%.