cs-compare

May 20, 2023

[30]:
```
import pandas as pd

# Load the evaluation results from each notebook
evaluation_files = ['evaluation_results_NB.csv', 'evaluation_results_KNN.csv',␣
 ↪'evaluation_results_NT.csv', 'evaluation_results_DT.csv',␣
 ↪'evaluation_results_RF.csv', 'evaluation_results_SVM.
 ↪csv','evaluation_results_LR.csv']

# Create an empty DataFrame to store the results
results_table = pd.DataFrame(columns=['Model', 'Accuracy', 'Precision',␣
 ↪'Recall', 'F1-score'])
```

[31]:
```
print(results_table)
```

```
Empty DataFrame
Columns: [Model, Accuracy, Precision, Recall, F1-score]
Index: []
```

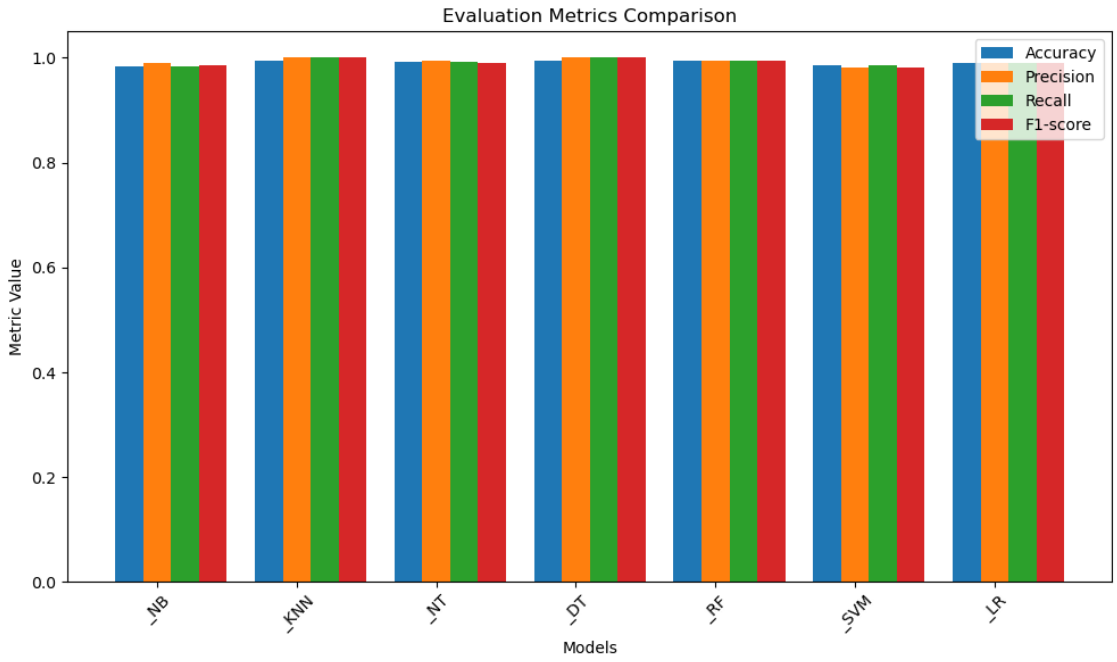
[32]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Load the evaluation results from each notebook
evaluation_files = ['evaluation_results_NB.csv', 'evaluation_results_KNN.csv',␣
 ↪'evaluation_results_NT.csv',
                    'evaluation_results_DT.csv', 'evaluation_results_RF.csv',␣
 ↪'evaluation_results_SVM.csv',
                    'evaluation_results_LR.csv']

# Create an empty DataFrame to store the results
results_table = pd.DataFrame(columns=['Model', 'Accuracy', 'Precision',␣
 ↪'Recall', 'F1-score'])

# Create empty lists to store the metrics for plotting
models = []
accuracies = []
precisions = []
```

```
recalls = []
f1_scores = []

# Iterate over the evaluation result files
for file in evaluation_files:
    # Extract the model name from the file name
    model_name = file.split('_results')[1].split('.csv')[0]

    # Read the evaluation results from the CSV file
    evaluation_data = pd.read_csv(file)

    # Extract the evaluation metrics
    accuracy = evaluation_data['Accuracy'].values[0]
    precision = evaluation_data['Precision'].values[0]
    recall = evaluation_data['Recall'].values[0]
    f1_score = evaluation_data['F1-score'].values[0]

    # Append the metrics to the lists
    models.append(model_name)
    accuracies.append(accuracy)
    precisions.append(precision)
    recalls.append(recall)
    f1_scores.append(f1_score)

    # Append the metrics to the results table
    results_table = results_table.append({'Model': model_name, 'Accuracy':␣
 ↪accuracy, 'Precision': precision,
                                          'Recall': recall, 'F1-score':␣
 ↪f1_score}, ignore_index=True)

# Plot the evaluation metrics
metrics = ['Accuracy', 'Precision', 'Recall', 'F1-score']
num_models = len(models)
width = 0.2
ind = np.arange(num_models)

plt.figure(figsize=(10, 6))

for i, metric in enumerate(metrics):
    plt.bar(ind + i * width, results_table[metric], width, label=metric)

plt.xlabel('Models')
plt.ylabel('Metric Value')
plt.title('Evaluation Metrics Comparison')
plt.xticks(ind + width, models, rotation=45)
plt.legend()
plt.tight_layout()
```

```
plt.show()

# Compare the accuracy of each model
max_accuracy = results_table['Accuracy'].max()
max_precision = results_table['Precision'].max()
max_recall = results_table['Recall'].max()
max_f1 = results_table['F1-score'].max()

best_model = results_table.loc[results_table['Accuracy'] == max_accuracy,␣
↪'Model'].values[0]

# Print the results table
print(results_table)
print("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
print("Best Model:", best_model)
print("Best Accuracy:", max_accuracy)
print("Best Precision:", max_precision)
print("Best Recall:", max_recall)
print("Best F1-Score:", max_f1)
```

```
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
```

```
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
C:\Users\pappu\AppData\Local\Temp\ipykernel_4712\1721995833.py:42:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  results_table = results_table.append({'Model': model_name, 'Accuracy':
accuracy, 'Precision': precision,
```



	Model	Accuracy	Precision	Recall	F1-score
0	_NB	0.983787	0.989495	0.983787	0.985642
1	_KNN	0.994241	1.000000	1.000000	1.000000
2	_NT	0.991737	0.994470	0.991737	0.990871
3	_DT	0.994335	1.000000	1.000000	1.000000
4	_RF	0.995368	0.995090	0.995368	0.995182
5	_SVM	0.986478	0.981538	0.986478	0.980242
6	_LR	0.990829	0.989645	0.990829	0.989906

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Best Model: _RF
Best Accuracy: 0.9953676171398166
Best Precision: 1.0
Best Recall: 1.0
Best F1-Score: 1.0
```

[ ]: