

In [4]:

```
# =====
# ml_results_visualization_full.py
# -----
# Visual analytics for full 16-dataset ML comparison project
# (Classification + Regression + Clustering)
# -----
# Author: Narendra Mishra
# =====

import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# =====
# 1 Setup
# =====

# Create plots directory
os.makedirs("plots", exist_ok=True)

# Load your final CSV
df = pd.read_csv("ml_results_full_2.csv")

# Clean and prepare
df = df.fillna(0)
df["TaskType"] = df["TaskType"].str.strip().str.capitalize()

classification_df = df[df["TaskType"] == "Classification"]
regression_df = df[df["TaskType"] == "Regression"]
clustering_df = df[df["TaskType"] == "Clustering"]

print(f"✓ Loaded results: {len(df)} total rows across {df['Dataset'].nunique()} datasets")

# =====
# 2 Top model per dataset (Classification)
# =====

top_models = (
    classification_df.loc[
        classification_df.groupby("Dataset")["Accuracy"].idxmax(),
        ["Dataset", "Model", "Accuracy", "F1-Score", "Precision", "Recall"]
    ]
    .sort_values("Accuracy", ascending=False)
)
top_models.to_csv("plots/top_models_summary.csv", index=False)
print("🏆 Saved top-performing models summary → plots/top_models_summary.csv")

# =====
# 3 Classification Visuals
# =====

print("\n📊 Creating Classification Visuals...")
```

```

# Heatmap: Accuracy per model per dataset
pivot_acc = classification_df.pivot_table(
    values="Accuracy", index="Model", columns="Dataset", aggfunc=np.mean
)
plt.figure(figsize=(14, 6))
sns.heatmap(
    pivot_acc, annot=True, fmt=".2f", cmap="coolwarm", cbar_kws={"label": "Accuracy"}
)
plt.title("🔥 Classification Model Accuracy Heatmap")
plt.ylabel("Model")
plt.xlabel("Dataset")
plt.tight_layout()
plt.savefig("plots/classification_accuracy_heatmap.png", dpi=300)
plt.show()
plt.close()

# Bar plot: F1-Score comparison
plt.figure(figsize=(14, 6))
sns.barplot(
    data=classification_df,
    x="Dataset", y="F1-Score", hue="Model", palette="tab10"
)
plt.title("📈 Classification Model F1-Score Comparison")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
plt.savefig("plots/classification_f1score_comparison.png", dpi=300)
plt.show()
plt.close()

# Boxplot: Accuracy distribution by model
plt.figure(figsize=(10, 5))
sns.boxplot(data=classification_df, x="Model", y="Accuracy", palette="Set2")
plt.title("📦 Accuracy Distribution per Model (Classification)")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
plt.savefig("plots/classification_accuracy_boxplot.png", dpi=300)
plt.show()
plt.close()

# =====
# 4 Regression Visuals
# =====

print("📈 Creating Regression Visuals...")

if not regression_df.empty:
    plt.figure(figsize=(12, 6))
    sns.barplot(
        data=regression_df,
        x="Dataset", y="MSE", hue="Model", palette="cool"
    )
    plt.title("📉 Regression Model MSE Comparison (Lower = Better)")
    plt.xticks(rotation=45, ha="right")
    plt.tight_layout()
    plt.savefig("plots/regression_mse_comparison.png", dpi=300)

```

```

plt.show()
plt.close()

# Boxplot for regression
plt.figure(figsize=(10, 5))
sns.boxplot(data=regression_df, x="Model", y="MSE", palette="Blues")
plt.title("▣ MSE Distribution per Model (Regression)")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
plt.savefig("plots/regression_mse_boxplot.png", dpi=300)
plt.show()
plt.close()
else:
    print("⚠️ No regression data found in the CSV.")

# =====
# 5 Clustering Visuals
# =====

print("◆ Creating Clustering Visuals...")

if not clustering_df.empty:
    plt.figure(figsize=(8, 5))
    sns.barplot(
        data=clustering_df,
        x="Dataset", y="Accuracy", hue="Model", palette="mako"
    )
    plt.title("◆ Clustering Model Silhouette Scores (Higher = Better)")
    plt.tight_layout()
    plt.savefig("plots/clustering_silhouette_comparison.png", dpi=300)
    plt.show()
    plt.close()
else:
    print("⚠️ No clustering data found in the CSV.")

# =====
# 6 Combined Summary
# =====

print("📊 Creating Combined Summary Overview...")

# Merge averages per model across tasks
summary_stats = (
    df.groupby(["TaskType", "Model"])[["Accuracy", "F1-Score", "MSE"]]
    .mean()
    .reset_index()
)
summary_stats.to_csv("plots/model_performance_summary.csv", index=False)

plt.figure(figsize=(12, 5))
sns.barplot(
    data=summary_stats[summary_stats["TaskType"] == "Classification"],
    x="Model", y="Accuracy", palette="viridis"
)
plt.title("Overall Average Accuracy per Model (Classification)")
plt.xticks(rotation=45, ha="right")

```

```

plt.tight_layout()
plt.savefig("plots/classification_avg_accuracy.png", dpi=300)
plt.show()
plt.close()

# =====
# 7 Wrap-up
# =====

print("\n✅ All visualizations created successfully!")
print("📁 Plots saved in folder: plots/")
print("📄 Summary CSVs saved: top_models_summary.csv & model_performance_summary.csv")

```

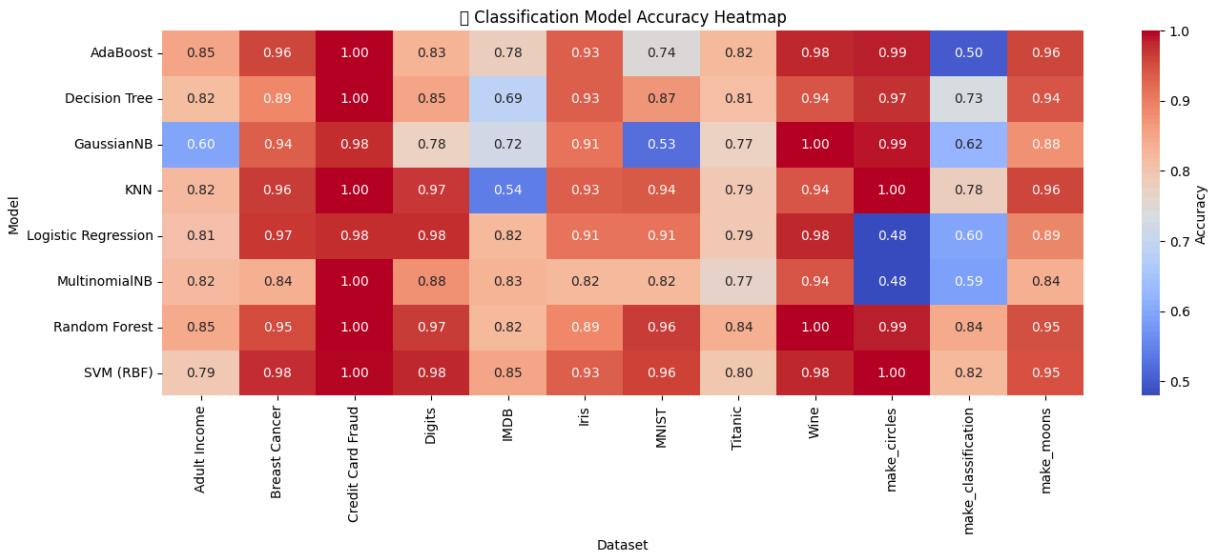
✅ Loaded results: 116 total rows across 16 datasets
 🏆 Saved top-performing models summary → plots/top_models_summary.csv

📊 Creating Classification Visuals...

```

C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:57: FutureWarning: Th
e provided callable <function mean at 0x000002591E1F6E80> is currently using DataFra
meGroupBy.mean. In a future version of pandas, the provided callable will be used di
rectly. To keep current behavior pass the string "mean" instead.
    pivot_acc = classification_df.pivot_table(
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:67: UserWarning: Glyp
h 128293 (\N{FIRE}) missing from font(s) DejaVu Sans.
    plt.tight_layout()
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:68: UserWarning: Glyp
h 128293 (\N{FIRE}) missing from font(s) DejaVu Sans.
    plt.savefig("plots/classification_accuracy_heatmap.png", dpi=300)
C:\Users\naren\AppData\Roaming\Python\Python312\site-packages\IPython\core\pylabtool
s.py:170: UserWarning: Glyph 128293 (\N{FIRE}) missing from font(s) DejaVu Sans.
    fig.canvas.print_figure(bytes_io, **kw)

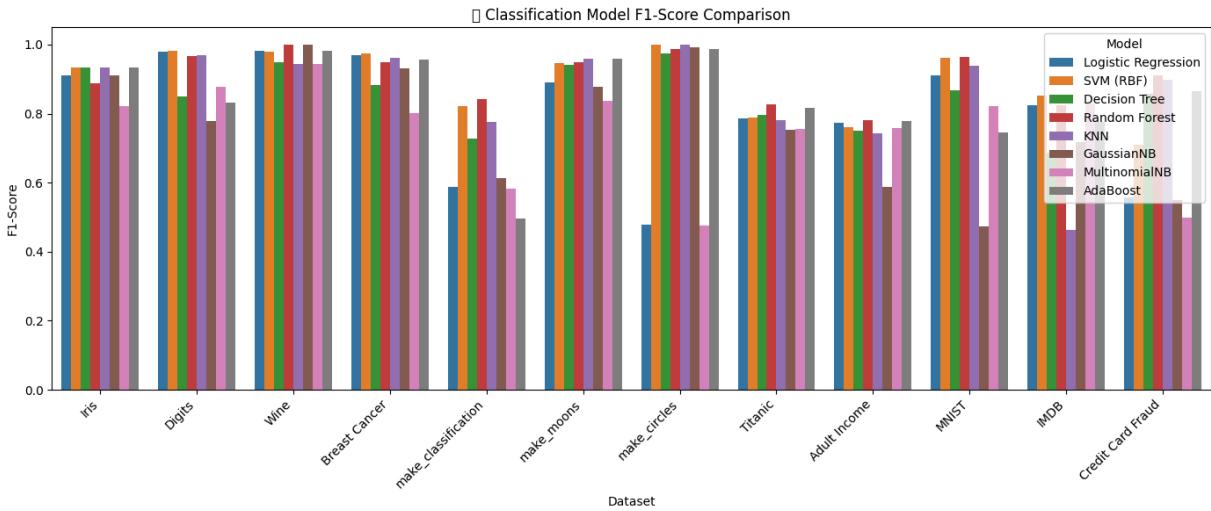
```



```

C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:80: UserWarning: Glyph 128200 (\N{CHART WITH UPWARDS TREND}) missing from font(s) DejaVu Sans.
    plt.tight_layout()
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:81: UserWarning: Glyph 128200 (\N{CHART WITH UPWARDS TREND}) missing from font(s) DejaVu Sans.
    plt.savefig("plots/classification_f1score_comparison.png", dpi=300)
C:\Users\naren\AppData\Roaming\Python\Python312\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 128200 (\N{CHART WITH UPWARDS TREND}) missing from font(s) DejaVu Sans.
    fig.canvas.print_figure(bytes_io, **kw)

```



```

C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:87: FutureWarning:

```

```

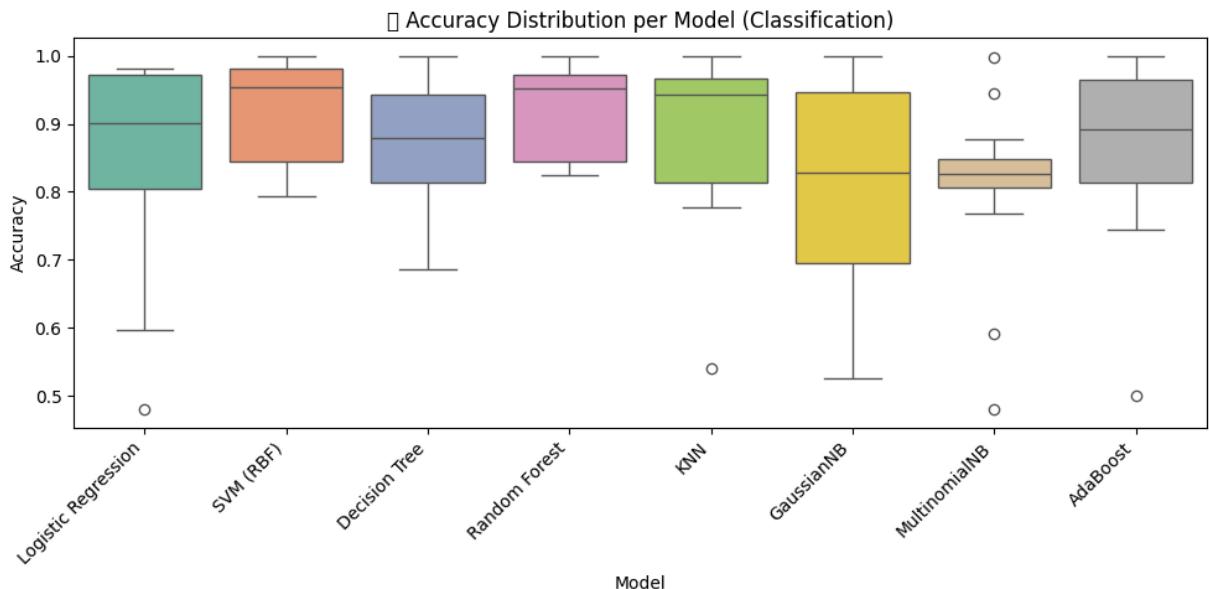
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```

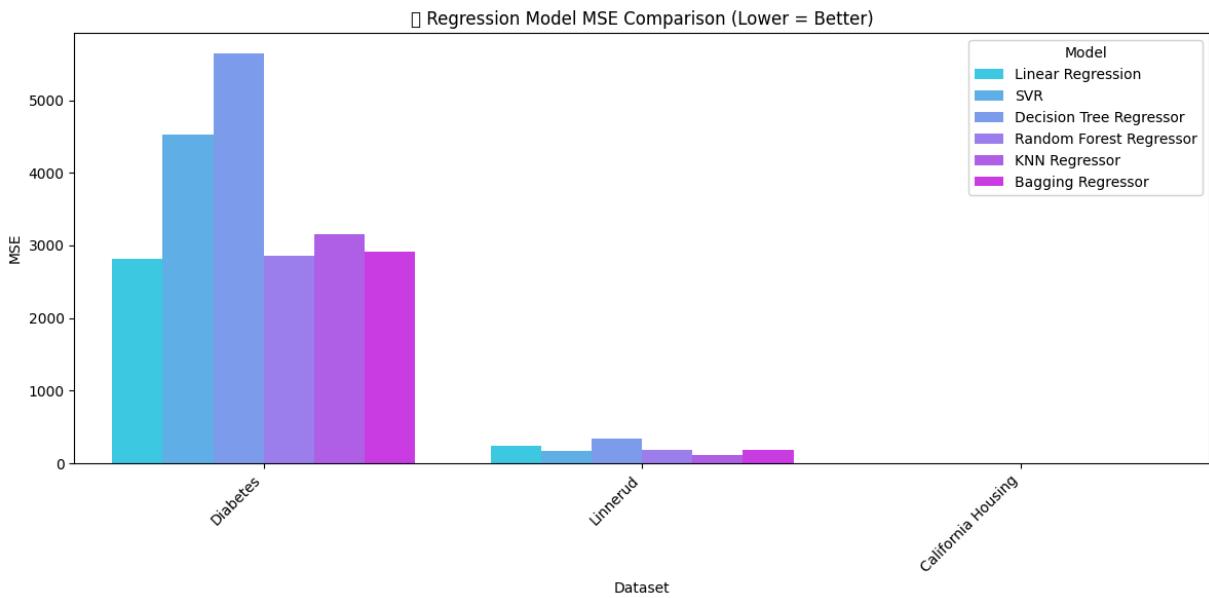
```

sns.boxplot(data=classification_df, x="Model", y="Accuracy", palette="Set2")
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:90: UserWarning: Glyph 128230 (\N{PACKAGE}) missing from font(s) DejaVu Sans.
    plt.tight_layout()
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:91: UserWarning: Glyph 128230 (\N{PACKAGE}) missing from font(s) DejaVu Sans.
    plt.savefig("plots/classification_accuracy_boxplot.png", dpi=300)
C:\Users\naren\AppData\Roaming\Python\Python312\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 128230 (\N{PACKAGE}) missing from font(s) DejaVu Sans.
    fig.canvas.print_figure(bytes_io, **kw)

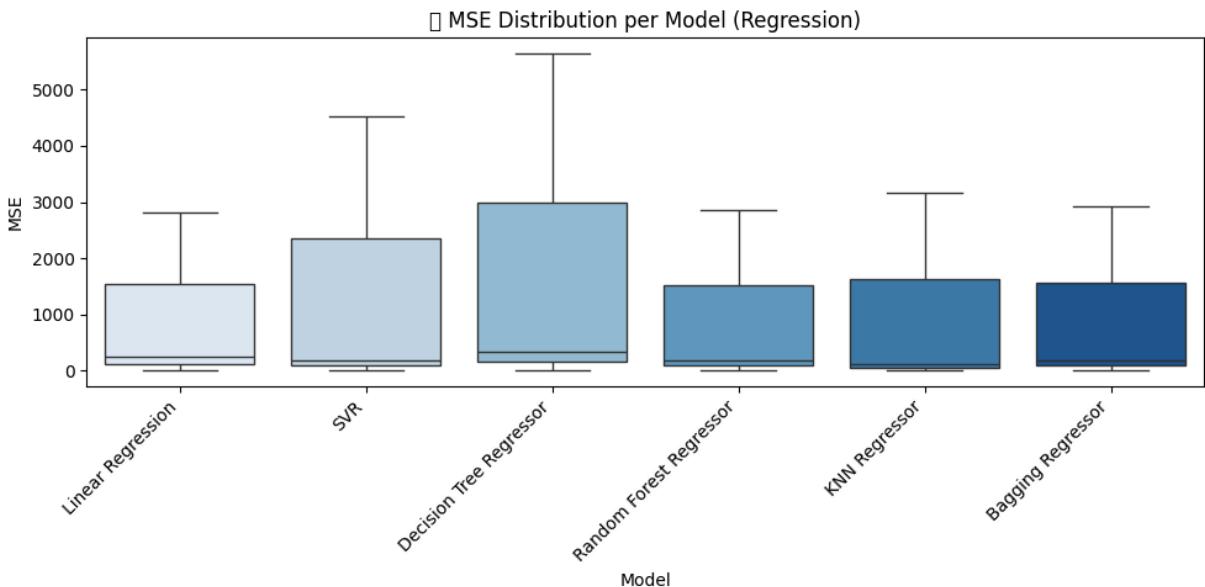
```



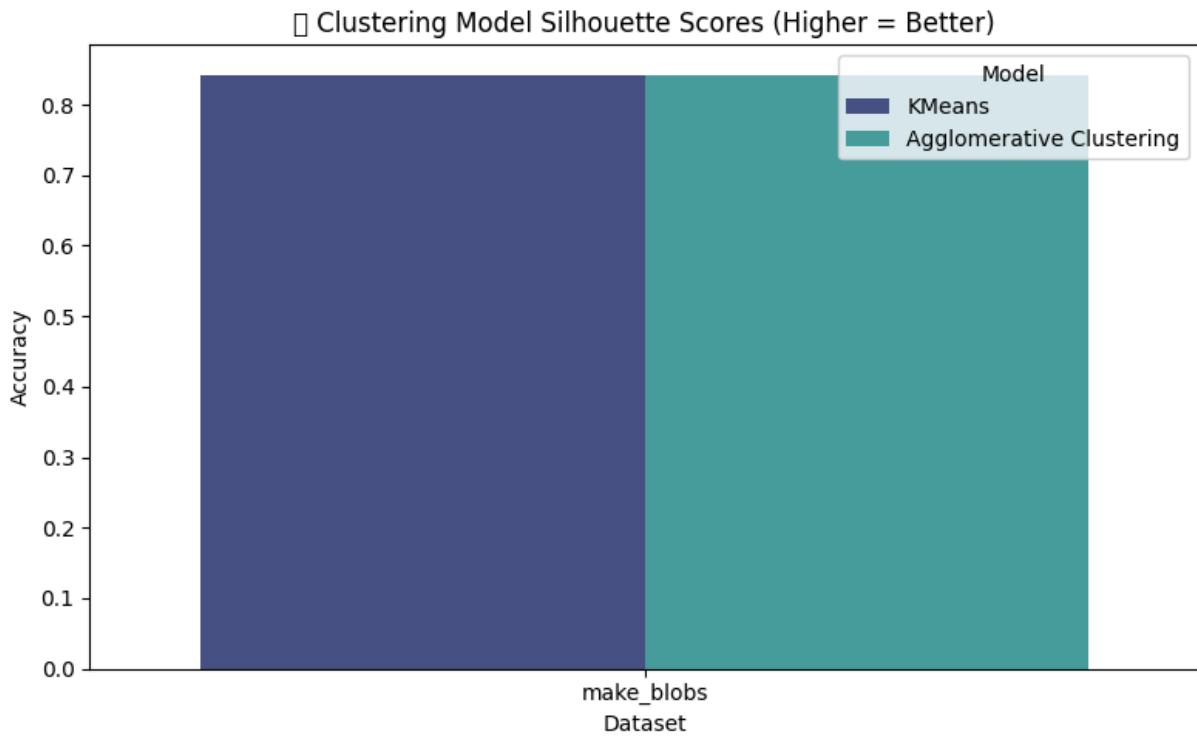
```
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:109: UserWarning: Glyph 128201 (\N{CHART WITH DOWNWARDS TREND}) missing from font(s) DejaVu Sans.
    plt.tight_layout()
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:110: UserWarning: Glyph 128201 (\N{CHART WITH DOWNWARDS TREND}) missing from font(s) DejaVu Sans.
    plt.savefig("plots/regression_mse_comparison.png", dpi=300)
[✓] Creating Regression Visuals...
C:\Users\naren\AppData\Roaming\Python\Python312\site-packages\IPython\core\pylabtools.py:170: UserWarning: Glyph 128201 (\N{CHART WITH DOWNWARDS TREND}) missing from font(s) DejaVu Sans.
    fig.canvas.print_figure(bytes_io, **kw)
```



```
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:116: FutureWarning:  
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1  
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.  
  
    sns.boxplot(data=regression_df, x="Model", y="MSE", palette="Blues")  
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:119: UserWarning: Gly  
ph 128230 (\N{PACKAGE}) missing from font(s) DejaVu Sans.  
    plt.tight_layout()  
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:120: UserWarning: Gly  
ph 128230 (\N{PACKAGE}) missing from font(s) DejaVu Sans.  
    plt.savefig("plots/regression_mse_boxplot.png", dpi=300)  
C:\Users\naren\AppData\Roaming\Python\Python312\site-packages\IPython\core\pylabtool  
s.py:170: UserWarning: Glyph 128230 (\N{PACKAGE}) missing from font(s) DejaVu Sans.  
    fig.canvas.print_figure(bytes_io, **kw)
```

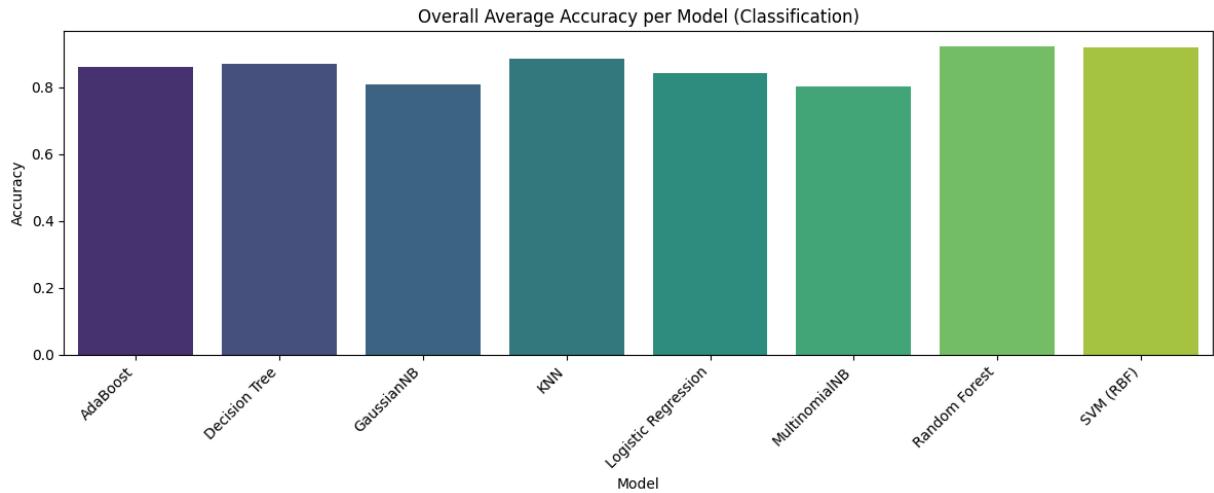


```
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:139: UserWarning: Gly  
ph 128313 (\N{SMALL BLUE DIAMOND}) missing from font(s) DejaVu Sans.  
    plt.tight_layout()  
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:140: UserWarning: Gly  
ph 128313 (\N{SMALL BLUE DIAMOND}) missing from font(s) DejaVu Sans.  
    plt.savefig("plots/clustering_silhouette_comparison.png", dpi=300)  
◆ Creating Clustering Visuals...  
C:\Users\naren\AppData\Roaming\Python\Python312\site-packages\IPython\core\pylabtool  
s.py:170: UserWarning: Glyph 128313 (\N{SMALL BLUE DIAMOND}) missing from font(s) De  
jaVu Sans.  
    fig.canvas.print_figure(bytes_io, **kw)
```



```
C:\Users\naren\AppData\Local\Temp\ipykernel_9584\2595178248.py:161: FutureWarning:  
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1  
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.
```

```
sns.barplot(  
    Creating Combined Summary Overview...
```



- All visualizations created successfully!
- Plots saved in folder: plots/
- Summary CSVs saved: top_models_summary.csv & model_performance_summary.csv

In []: