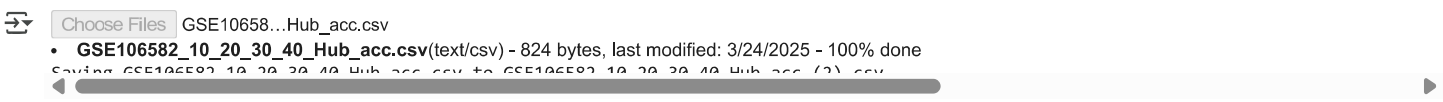
```
!pip install pandas matplotlib seaborn
```

```
['Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)',
 'Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)',
 'Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.13.2)',
 'Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.0.2)',
 'Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2)',
 'Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)',
 'Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)',
 'Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)',
 'Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)',
 'Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)',
 'Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)',
 'Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)',
 'Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)',
 'Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1)',
 'Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)']
```

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as mtick
from google.colab import files
from google.colab import drive
```

```
files.download("accuracy_comparison_GSE21815.png")
```

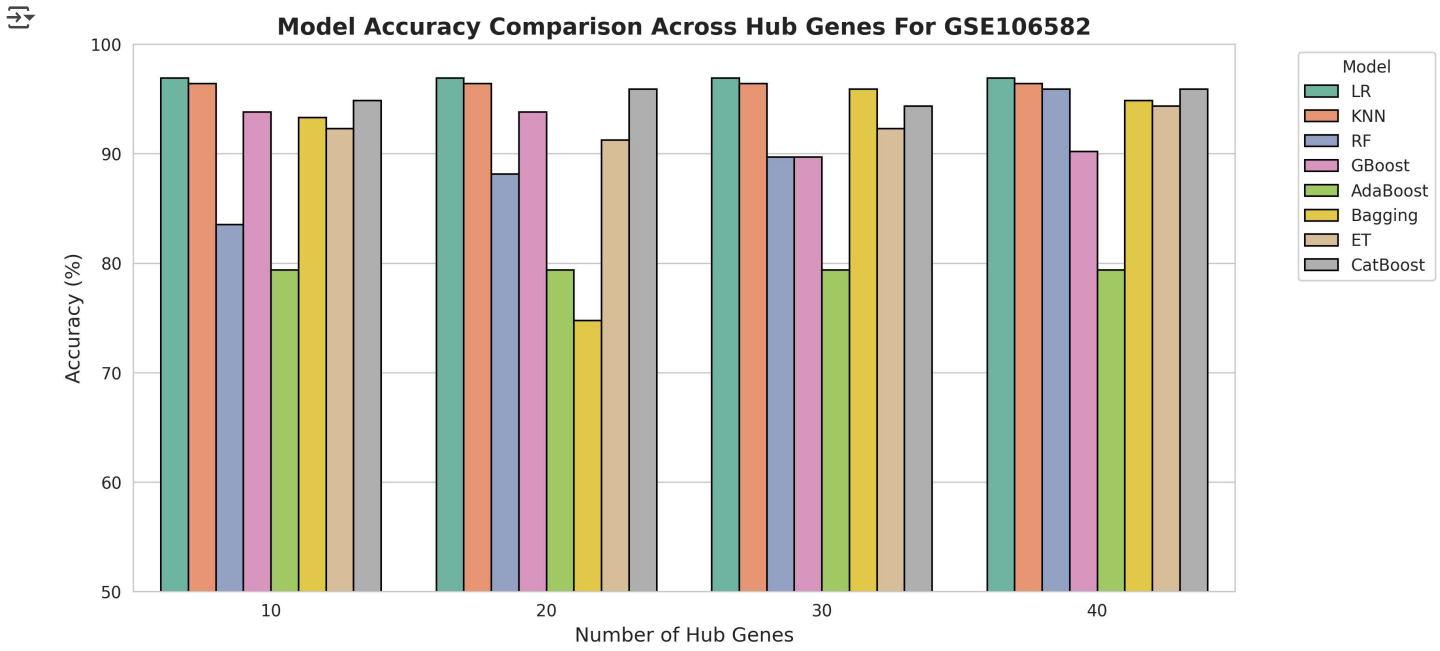
```
uploaded = files.upload()
```




```
df_2 = pd.read_csv("GSE106582_10_20_30_40_Hub_acc.csv")
print(df_2.head())
```

```
   Hub Genes     Model    Metric  Value
0         10        LR  Accuracy  96.91
1         10       KNN  Accuracy  96.39
2         10        RF  Accuracy  83.51
3         10    GBoost  Accuracy  93.81
4         10  AdaBoost  Accuracy  79.38
```

```
sns.set_style("whitegrid")
sns.set_palette("Set2")

plt.figure(figsize=(12, 6), dpi=300)

ax = sns.barplot(x="Hub Genes", y="Value", hue="Model", data=df_2[df_2["Metric"] == "Accuracy"], edgecolor='black')

plt.title("Model Accuracy Comparison Across Hub Genes For GSE106582", fontsize=14, fontweight='bold')
plt.xlabel("Number of Hub Genes", fontsize=12)
plt.ylabel("Accuracy (%)", fontsize=12)
plt.ylim(50, 100)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.legend(title="Model", fontsize=10, bbox_to_anchor=(1.05, 1), loc='upper left')

plt.savefig("accuracy_comparison_GSE106582.png", dpi=300, bbox_inches='tight')

plt.show()
```




```
files.download("accuracy_comparison_GSE106582.png")
```

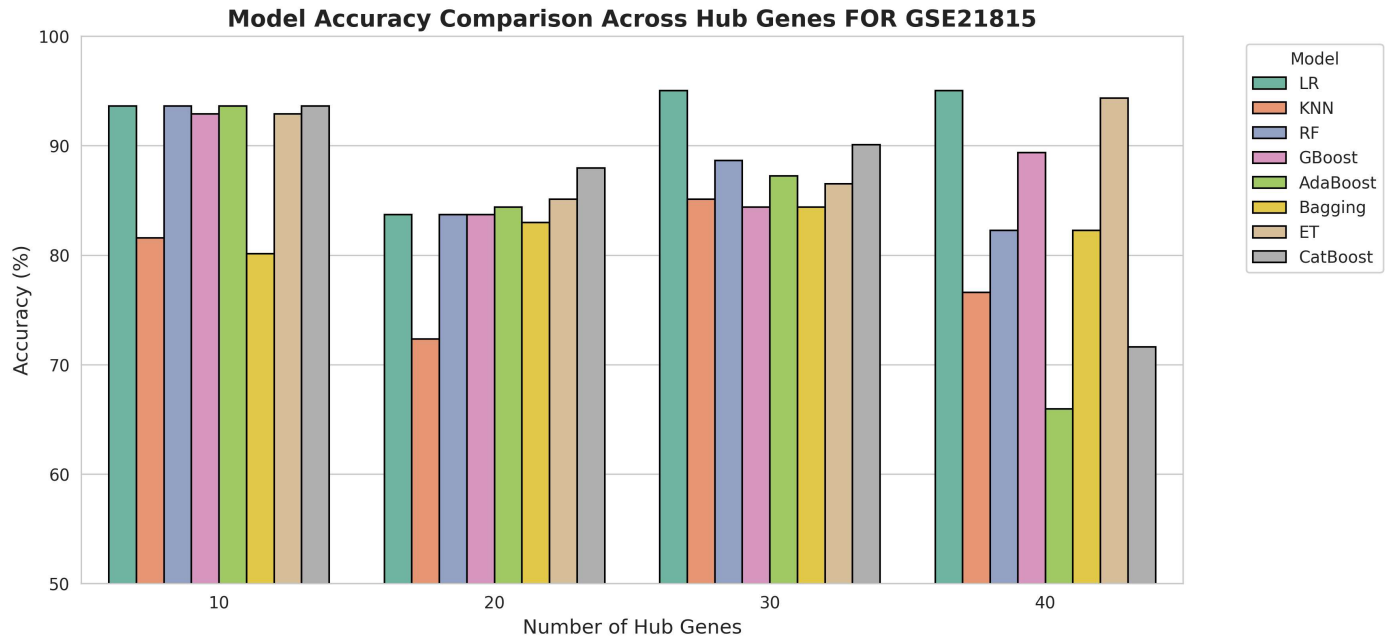
```
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable

```
df_1 = pd.read_csv("GSE21815_10_20_30_40_Hub_acc.csv")
print(df_1.head())
```

```
   Hub Genes     Model    Metric  Value
0         10        LR  Accuracy  93.62
1         10       KNN  Accuracy  81.56
2         10        RF  Accuracy  93.62
3         10    GBoost  Accuracy  92.91
4         10  AdaBoost  Accuracy  93.62
```

```
sns.set_style("whitegrid")
sns.set_palette("Set2")

plt.figure(figsize=(12, 6), dpi=300)

ax = sns.barplot(x="Hub Genes", y="Value", hue="Model", data=df_1[df_1["Metric"] == "Accuracy"], edgecolor='black')

plt.title("Model Accuracy Comparison Across Hub Genes FOR GSE21815", fontsize=14, fontweight='bold')
plt.xlabel("Number of Hub Genes", fontsize=12)
plt.ylabel("Accuracy (%)", fontsize=12)
plt.ylim(50, 100)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.legend(title="Model", fontsize=10, bbox_to_anchor=(1.05, 1), loc='upper left')

plt.savefig("accuracy_comparison_GSE21815.png", dpi=300, bbox_inches='tight')

plt.show()
```



Key Observations:

For GSE21815:

1. 10 hub genes: Most models perform well, with several models achieving accuracy above 90%.
2. 20 hub genes: A performance drop is visible for some models (e.g., KNN).
3. 30 hub genes: Accuracy increases again, and some models outperform the 10-hub gene scenario.
4. 40 hub genes: Performance is inconsistent, with some models achieving high accuracy (above 90%) while others drop significantly.

For GSE106582:

1. 10 hub genes: Strong performance, with most models achieving accuracy above 90%.
2. 20 hub genes: Slight fluctuation, but still maintains good accuracy for most models.
3. 30 hub genes: Appears to be the best performing across almost all models.
4. 40 hub genes: Some models drop in accuracy compared to 30 hub genes.

Conclusion: ✅ Best Number of Hub Genes: 30

1. Both datasets (GSE21815 and GSE106582) show high accuracy for most models at 30 hub genes.
2. Performance is more consistent across different models at 30 hub genes.
3. 40 hub genes show more variation in performance—some models perform very well, but others drop.

- If your goal is to find a balance where most models perform well, 30 hub genes seems to be the best choice.

Why is 30 Hub Genes the Best Choice?

After analyzing the accuracy of different machine learning models across two independent datasets (GSE21815 and GSE106582), 30 hub genes consistently showed the best overall performance.

Key Reasons:

1. Higher Accuracy Across Models

- When using 30 hub genes, most models achieved high accuracy (above 90%) in both datasets.
- This indicates that 30 genes provide sufficient information for effective classification.

2. More Consistent Performance
    - Some hub gene counts (10 and 20) showed fluctuations—certain models performed well, while others dropped.
    - At 30 hub genes, performance was more stable, meaning it is a more reliable choice across different models.
3. Avoiding Overfitting with 40 Genes
    - While 40 hub genes worked well for some models, others showed a decrease in accuracy.
    - This suggests adding too many genes might introduce noise rather than improving classification.

Final Conclusion

30 hub genes provide the best balance between high accuracy, consistency, and reliability for colorectal cancer classification. This makes it the optimal choice for biomarker selection in this study.