

```
import matplotlib.pyplot as plt
import numpy as np

# Data for RMSE and MSE comparison
models = ['SVD[4]', 'PMF[4]', 'ALS', 'NCF']
rmse = [0.8860, 0.8855, 0.7503, 0.0626]
mse = [0.6964, 0.8855, 0.5629, 0.0039]

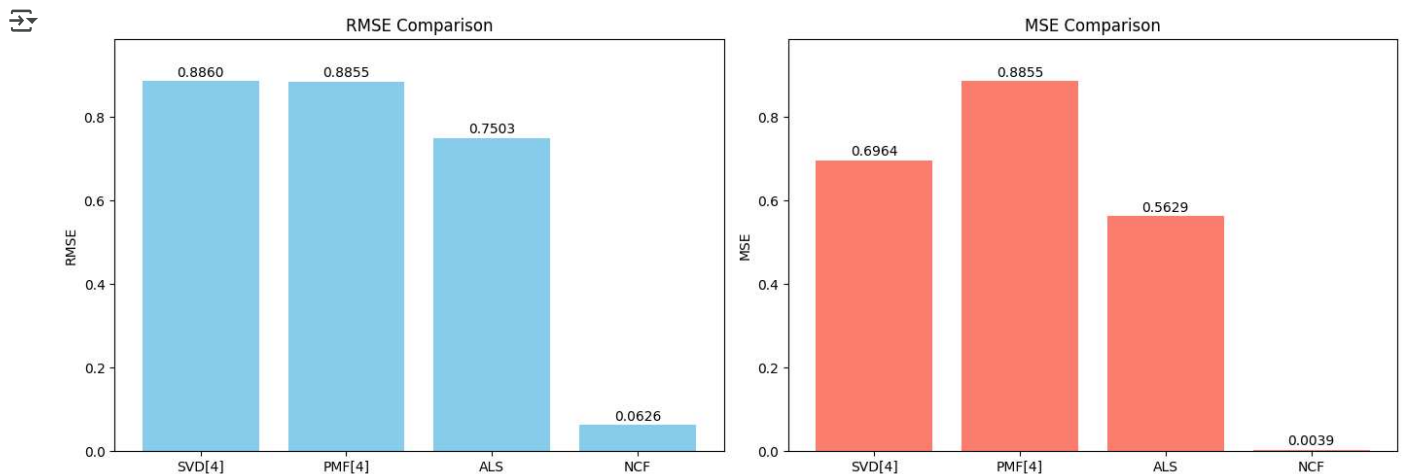
# Data for training time comparison
frameworks = ['Spark', 'TensorFlow']
ncf_training_time = [105.1360, 276.5037]

# Plotting RMSE and MSE Comparison
fig, ax = plt.subplots(1, 2, figsize=(14, 5))

# RMSE Plot
ax[0].bar(models, rmse, color='skyblue')
ax[0].set_title('RMSE Comparison')
ax[0].set_ylabel('RMSE')
ax[0].set_ylim(0, max(rmse) + 0.1)
for i, v in enumerate(rmse):
    ax[0].text(i, v + 0.01, f'{v:.4f}', ha='center')

# MSE Plot
ax[1].bar(models, mse, color='salmon')
ax[1].set_title('MSE Comparison')
ax[1].set_ylabel('MSE')
ax[1].set_ylim(0, max(mse) + 0.1)
for i, v in enumerate(mse):
    ax[1].text(i, v + 0.01, f'{v:.4f}', ha='center')

plt.tight_layout()
plt.show()
```



```
# Re-importing required libraries after kernel reset
import matplotlib.pyplot as plt
import numpy as np

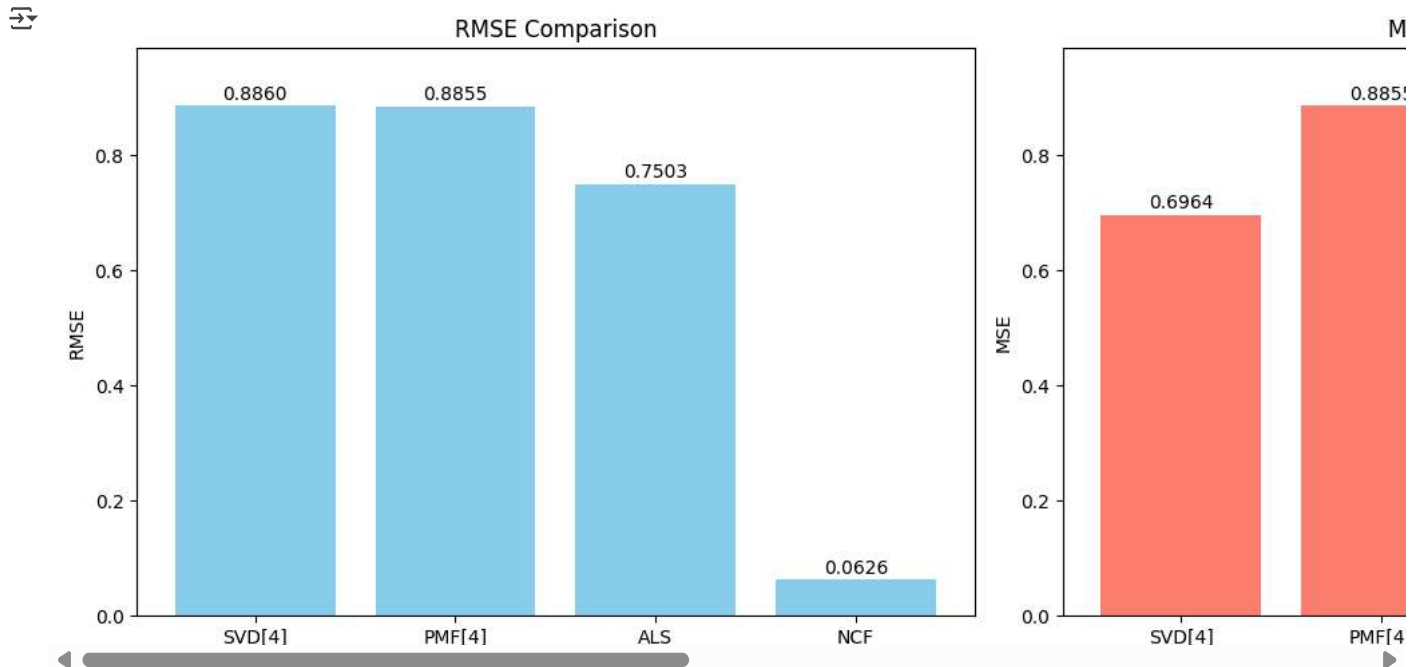
# Data for RMSE and MSE comparison
models = ['SVD[4]', 'PMF[4]', 'ALS', 'NCF']
rmse = [0.8860, 0.8855, 0.7503, 0.0626]
mse = [0.6964, 0.8855, 0.5629, 0.0039]

# Plotting RMSE and MSE Comparison
fig, ax = plt.subplots(1, 2, figsize=(14, 5))

# RMSE Plot
ax[0].bar(models, rmse, color='skyblue')
ax[0].set_title('RMSE Comparison')
ax[0].set_ylabel('RMSE')
ax[0].set_ylim(0, max(rmse) + 0.1)
for i, v in enumerate(rmse):
    ax[0].text(i, v + 0.01, f'{v:.4f}', ha='center')
```

```
# MSE Plot
ax[1].bar(models, mse, color='salmon')
ax[1].set_title('MSE Comparison')
ax[1].set_ylabel('MSE')
ax[1].set_ylim(0, max(mse) + 0.1)
for i, v in enumerate(mse):
    ax[1].text(i, v + 0.01, f"{v:.4f}", ha='center')

plt.tight_layout()
plt.show()
```



```
# Data for training time comparison
frameworks = ['Apache Spark', 'TensorFlow']
training_time = [105.1360, 276.5037]

# Plotting training time comparison
plt.figure(figsize=(7, 5))
bars = plt.bar(frameworks, training_time, color=['mediumseagreen', 'tomato'])
plt.title('Training Time Comparison for NCF')
plt.ylabel('Training Time (seconds)')
plt.ylim(0, max(training_time) + 50)

# Adding text labels
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval + 5, f"{yval:.2f}", ha='center', va='bottom')

plt.tight_layout()
plt.show()
```

