(91) tamil songs - YouTube | Welcome To Colab - Colab | New Tab

colab.research.google.com/#scrollTo=e67c13b4

Welcome To Colab — Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive

RAM Disk

Table of contents

- Welcome to Colab!
- Task
- + Section

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.barplot(x='Model', y='Mean', hue='CV Type', data=accuracy_df, capsize=0.1, errorbar='sd')

plt.title('Model Performance Comparison (Accuracy)', fontsize=14)
plt.xlabel('Model and CV Type', fontsize=12)
plt.ylabel('Mean Accuracy', fontsize=12)
plt.ylim(0.85, 1.0) # Set y-axis limit for better visualization
plt.legend(title='Cross-Validation Type')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
print("Bar plot visualizing mean accuracy with standard deviation as error bars displayed.")
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu

7:19 PM Python 3

(91) tamil songs - YouTube | Welcome To Colab - Colab | New Tab

colab.research.google.com/#scrollTo=307fcd90

Welcome To Colab — Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive

RAM Disk

Table of contents

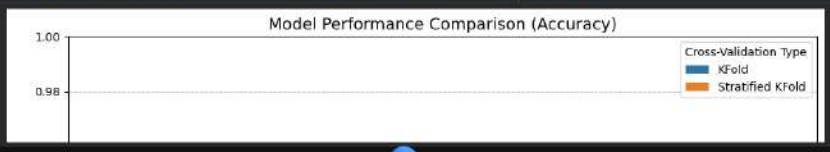
- Welcome to Colab!
- Task
- + Section

```
accuracy_df = comparison_df[comparison_df['Metric'] == 'accuracy']
print("Filtered accuracy DataFrame:")
print(accuracy_df)
```

```
Filtered accuracy DataFrame:
              Model          CV Type    Metric      Mean       Std
0     Random Forest            KFold  accuracy  0.925000  0.055277
4     Random Forest  Stratified KFold  accuracy  0.933333  0.033333
8               SVM            KFold  accuracy  0.958333  0.052705
12              SVM  Stratified KFold  accuracy  0.950000  0.016667
16    Decision Tree            KFold  accuracy  0.916667  0.069722
20    Decision Tree  Stratified KFold  accuracy  0.933333  0.050000
```

Reasoning: I will create a bar plot to visualize the mean accuracy for each model and cross-validation type, including standard deviation as error bars, as instructed in the subtask.

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.barplot(x='Model', y='Mean', hue='CV Type', data=accuracy_df, capsize=0.1, errorbar='sd')

plt.title('Model Performance Comparison (Accuracy)', fontsize=14)
plt.xlabel('Model and CV Type', fontsize=12)
plt.ylabel('Mean Accuracy', fontsize=12)
plt.ylim(0.85, 1.0) # Set y-axis limit for better visualization
plt.legend(title='Cross-Validation Type')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
print("Bar plot visualizing mean accuracy with standard deviation as error bars displayed.")
```



Variables Terminal

7:19 PM Python 3

(91) tamil songs - YouTube | Welcome To Colab - Colab | New Tab

colab.research.google.com/#scrollTo=307fcd90

Welcome To Colab — Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive

RAM Disk

Table of contents

- Welcome to Colab!
- Task
- + Section

```
            'Mean': values.mean(),
            'Std': values.std()
        })

# Create the DataFrame
comparison_df = pd.DataFrame(data_rows)

print("Comparison table of model performance metrics:")
print(comparison_df)
```

```
Comparison table of model performance metrics:
                Model           CV Type     Metric      Mean       Std
0   Random Forest              KFold   accuracy  0.925000  0.055277
1   Random Forest              KFold  precision  0.936243  0.055596
2   Random Forest              KFold     recall  0.925000  0.055277
3   Random Forest              KFold   f1_score  0.926058  0.055151
4   Random Forest  Stratified KFold   accuracy  0.933333  0.033333
5   Random Forest  Stratified KFold  precision  0.946378  0.021933
6   Random Forest  Stratified KFold     recall  0.933333  0.033333
7   Random Forest  Stratified KFold   f1_score  0.932125  0.035037
8             SVM              KFold   accuracy  0.958333  0.052705
9             SVM              KFold  precision  0.963571  0.048874
10            SVM              KFold     recall  0.958333  0.052705
11            SVM              KFold   f1_score  0.958713  0.052557
12            SVM  Stratified KFold   accuracy  0.950000  0.016667
13            SVM  Stratified KFold  precision  0.957153  0.011912
14            SVM  Stratified KFold     recall  0.950000  0.016667
15            SVM  Stratified KFold   f1_score  0.949670  0.017164
16  Decision Tree              KFold   accuracy  0.916667  0.069722
17  Decision Tree              KFold  precision  0.928145  0.071131
18  Decision Tree              KFold     recall  0.916667  0.069722
19  Decision Tree              KFold   f1_score  0.917464  0.070296
20  Decision Tree  Stratified KFold   accuracy  0.933333  0.050000
21  Decision Tree  Stratified KFold  precision  0.942780  0.043369
22  Decision Tree  Stratified KFold     recall  0.933333  0.050000
23  Decision Tree  Stratified KFold   f1_score  0.932324  0.051195
```

```
accuracy_df = comparison_df[comparison_df['Metric'] == 'accuracy']
print("Filtered accuracy DataFrame:")
print(accuracy_df)
```

```
Filtered accuracy DataFrame:
                Model           CV Type    Metric      Mean       Std
0   Random Forest              KFold  accuracy  0.925000  0.055277
```

Variables Terminal

7:19 PM Python 3

(91) tamil songs - YouTube | Welcome To Colab - Colab | New Tab

colab.research.google.com/#scrollTo=307fcd90

Welcome To Colab — Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text | Run all | Copy to Drive

RAM Disk

Table of contents

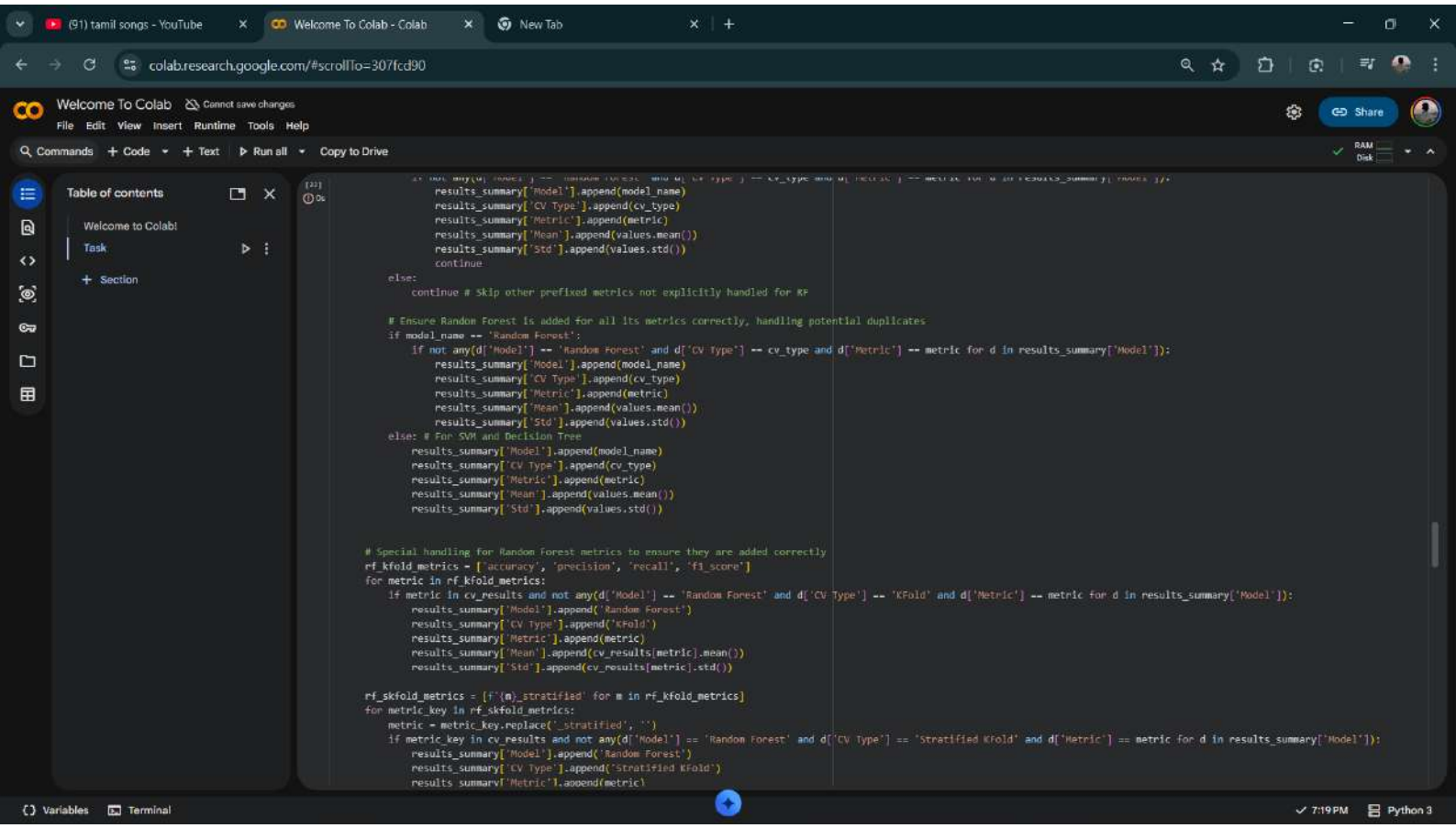
- Welcome to Colab!
- Task
- + Section

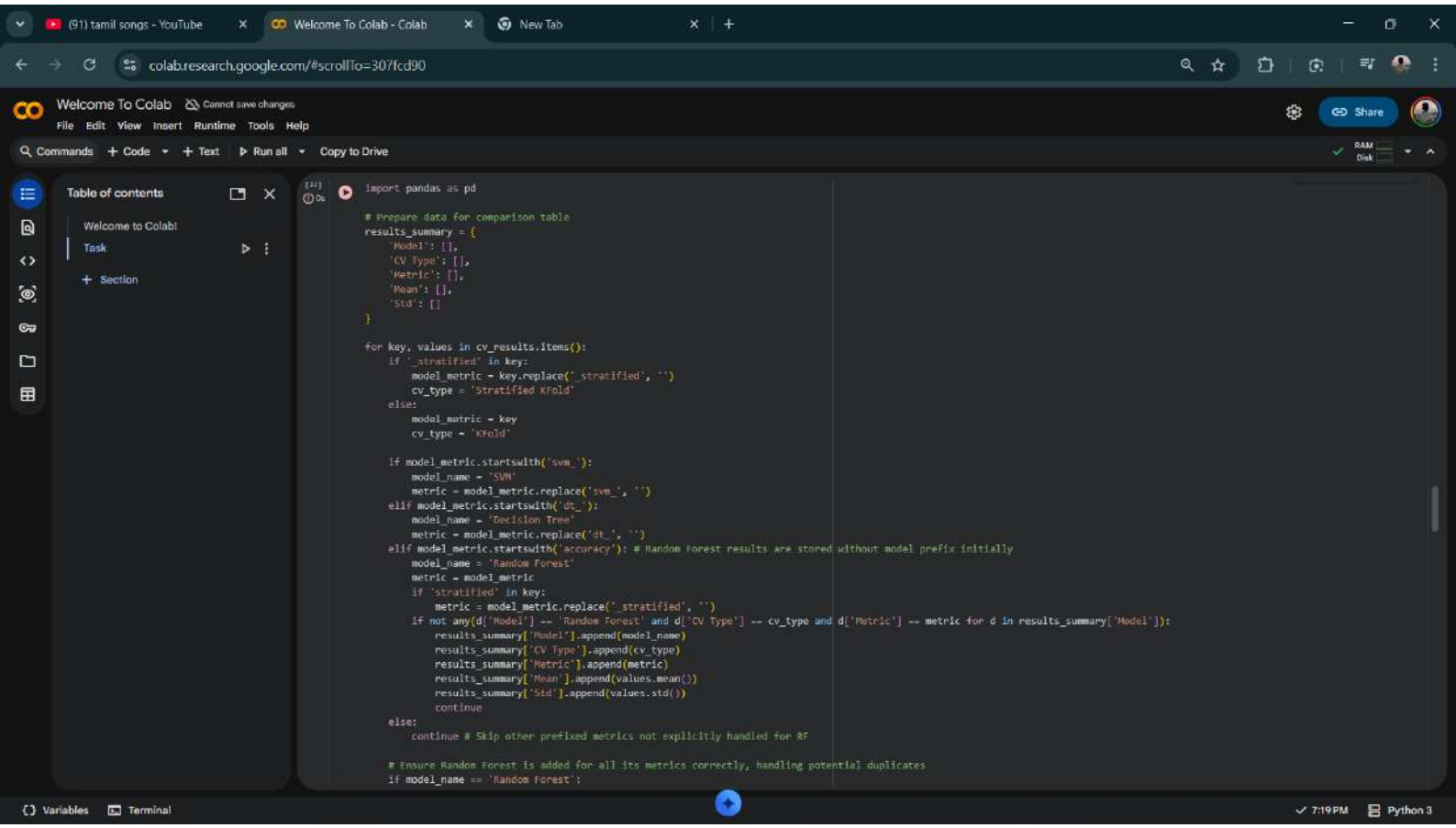
[23] ✓ 0s

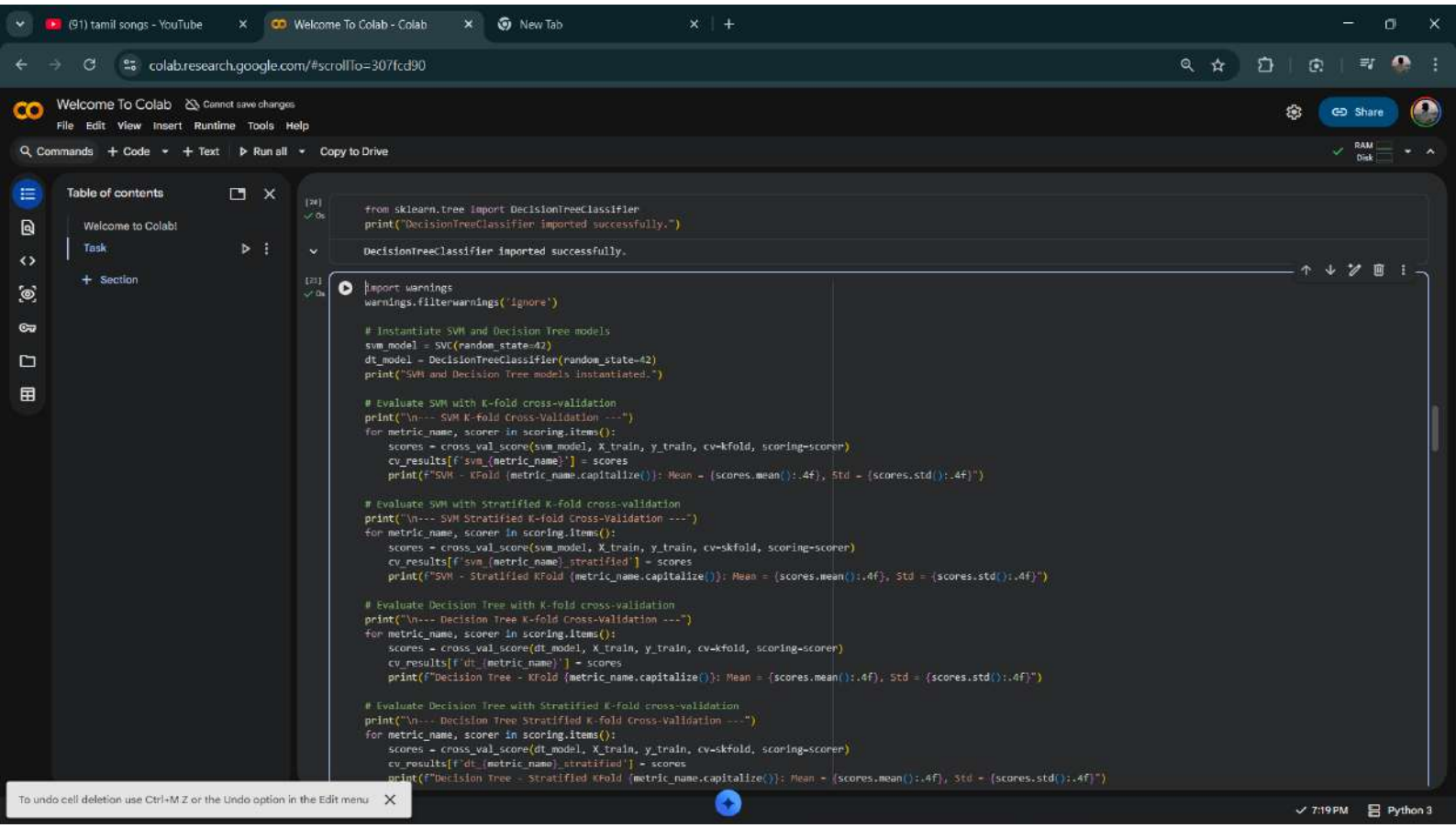
```
import pandas as pd

# Prepare data for comparison table
data_rows = []

for key, values in cv_results.items():
    if '_stratified' in key:
        model_metric = key.replace('_stratified', '')
        cv_type = 'Stratified KFold'
    else:
        model_metric = key
        cv_type = 'KFold'

    model_name = ''
    metric = ''

    if model_metric.startswith('svm_'):
        model_name = 'SVM'
        metric = model_metric.replace('svm_', '')
    elif model_metric.startswith('dt_'):
        model_name = 'Decision Tree'
        metric = model_metric.replace('dt_', '')
    elif any(m in model_metric for m in ['accuracy', 'precision', 'recall', 'f1_score']):
        # For Random Forest, initial keys are just metric names
        model_name = 'Random Forest'
        metric = model_metric

    if model_name:
        data_rows.append({
            'Model': model_name,
            'CV Type': cv_type,
            'Metric': metric,
            'Mean': values.mean(),
            'Std': values.std()
        })

# Create the DataFrame
comparison_df = pd.DataFrame(data_rows)

print("Comparison table of model performance metrics:")
print(comparison_df)
```

Variables | Terminal

7:19 PM | Python 3


(91) tamil songs - YouTube
Welcome To Colab - Colab
New Tab
colab.research.google.com/#scrollTo=307fcd90
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all Copy to Drive
RAM
Disk
Table of contents
Welcome to Colab!
Task
+ Section
[23]
0s
results_summary['Model'].append(model_name)
results_summary['CV Type'].append(cv_type)
results_summary['Metric'].append(metric)
results_summary['Mean'].append(values.mean())
results_summary['Std'].append(values.std())
continue
else:
continue # Skip other prefixed metrics not explicitly handled for RF
# Ensure Random Forest is added for all its metrics correctly, handling potential duplicates
if model_name == 'Random Forest':
if not any(d['Model'] == 'Random Forest' and d['CV Type'] == cv_type and d['Metric'] == metric for d in results_summary['Model']):
results_summary['Model'].append(model_name)
results_summary['CV Type'].append(cv_type)
results_summary['Metric'].append(metric)
results_summary['Mean'].append(values.mean())
results_summary['Std'].append(values.std())
else: # For SVM and Decision Tree
results_summary['Model'].append(model_name)
results_summary['CV Type'].append(cv_type)
results_summary['Metric'].append(metric)
results_summary['Mean'].append(values.mean())
results_summary['Std'].append(values.std())
# Special handling for Random Forest metrics to ensure they are added correctly
rf_kfold_metrics = ['accuracy', 'precision', 'recall', 'f1_score']
for metric in rf_kfold_metrics:
if metric in cv_results and not any(d['Model'] == 'Random Forest' and d['CV Type'] == 'KFold' and d['Metric'] == metric for d in results_summary['Model']):
results_summary['Model'].append('Random Forest')
results_summary['CV Type'].append('KFold')
results_summary['Metric'].append(metric)
results_summary['Mean'].append(cv_results[metric].mean())
results_summary['Std'].append(cv_results[metric].std())
rf_skfold_metrics = [f'{m}_stratified' for m in rf_kfold_metrics]
for metric_key in rf_skfold_metrics:
metric = metric_key.replace('_stratified', '')
if metric_key in cv_results and not any(d['Model'] == 'Random Forest' and d['CV Type'] == 'Stratified KFold' and d['Metric'] == metric for d in results_summary['Model']):
results_summary['Model'].append('Random Forest')
results_summary['CV Type'].append('Stratified KFold')
results_summary['Metric'].append(metric)
Variables
Terminal
7:19 PM
Python 3


(91) tamil songs - YouTube
Welcome To Colab - Colab
New Tab
colab.research.google.com/#scrollTo=307fcd90
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all Copy to Drive
RAM
Disk
Table of contents
Welcome to Colab!
Task
+ Section
[22]
0s
import pandas as pd
# Prepare data for comparison table
results_summary = {
'Model': [],
'CV Type': [],
'Metric': [],
'Mean': [],
'Std': []
}
for key, values in cv_results.items():
if '_stratified' in key:
model_metric = key.replace('_stratified', '')
cv_type = 'Stratified KFold'
else:
model_metric = key
cv_type = 'KFold'
if model_metric.startswith('svm_'):
model_name = 'SVM'
metric = model_metric.replace('svm_', '')
elif model_metric.startswith('dt_'):
model_name = 'Decision Tree'
metric = model_metric.replace('dt_', '')
elif model_metric.startswith('accuracy'): # Random Forest results are stored without model prefix initially
model_name = 'Random Forest'
metric = model_metric
if 'stratified' in key:
metric = model_metric.replace('_stratified', '')
if not any(d['Model'] == 'Random Forest' and d['CV Type'] == cv_type and d['Metric'] == metric for d in results_summary['Model']):
results_summary['Model'].append(model_name)
results_summary['CV Type'].append(cv_type)
results_summary['Metric'].append(metric)
results_summary['Mean'].append(values.mean())
results_summary['Std'].append(values.std())
continue
else:
continue # Skip other prefixed metrics not explicitly handled for RF
# Ensure Random Forest is added for all its metrics correctly, handling potential duplicates
if model_name == 'Random Forest':
Variables
Terminal
7:19 PM
Python 3


(91) tamil songs - YouTube
Welcome To Colab - Colab
New Tab
colab.research.google.com/#scrollTo=307fcd90
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all Copy to Drive
RAM
Disk
Table of contents
Welcome to Colab!
Task
+ Section
from sklearn.tree import DecisionTreeClassifier
print("DecisionTreeClassifier imported successfully.")
DecisionTreeClassifier imported successfully.
import warnings
warnings.filterwarnings('ignore')
# Instantiate SVM and Decision Tree models
svm_model = SVC(random_state=42)
dt_model = DecisionTreeClassifier(random_state=42)
print("SVM and Decision Tree models instantiated.")
# Evaluate SVM with K-fold cross-validation
print("\n--- SVM K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
scores = cross_val_score(svm_model, X_train, y_train, cv=kfold, scoring=scorer)
cv_results[f'svm_{metric_name}'] = scores
print(f"SVM - KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")
# Evaluate SVM with Stratified K-fold cross-validation
print("\n--- SVM Stratified K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
scores = cross_val_score(svm_model, X_train, y_train, cv=skfold, scoring=scorer)
cv_results[f'svm_{metric_name}_stratified'] = scores
print(f"SVM - Stratified KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")
# Evaluate Decision Tree with K-fold cross-validation
print("\n--- Decision Tree K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
scores = cross_val_score(dt_model, X_train, y_train, cv=kfold, scoring=scorer)
cv_results[f'dt_{metric_name}'] = scores
print(f"Decision Tree - KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")
# Evaluate Decision Tree with Stratified K-fold cross-validation
print("\n--- Decision Tree Stratified K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
scores = cross_val_score(dt_model, X_train, y_train, cv=skfold, scoring=scorer)
cv_results[f'dt_{metric_name}_stratified'] = scores
print(f"Decision Tree - Stratified KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")
To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu
7:19 PM
Python 3

(91) tamil songs - YouTube | Welcome To Colab - Colab | New Tab

colab.research.google.com/#scrollTo=db70e37a

Welcome To Colab — Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive

RAM Disk

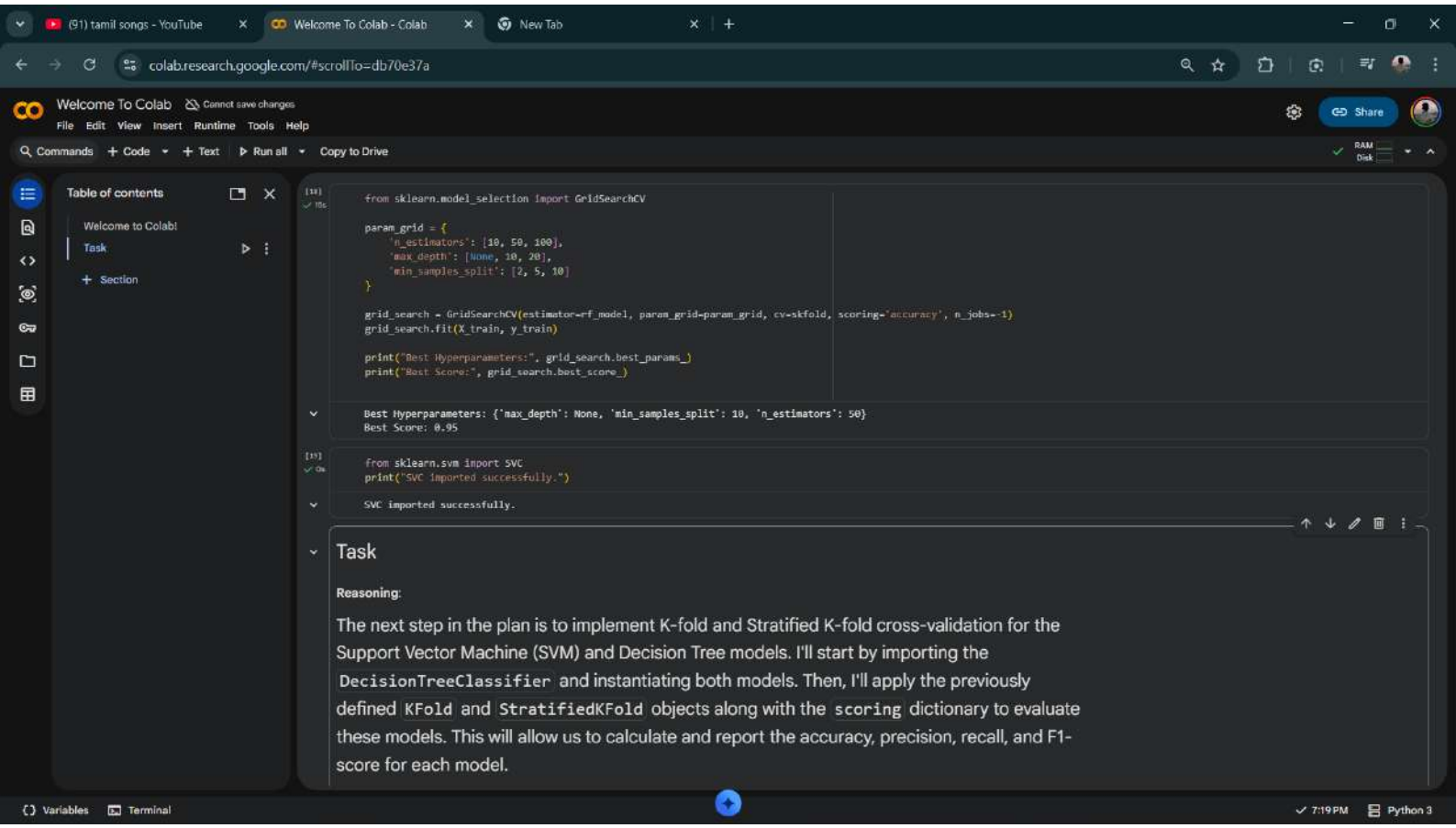
Table of contents

- Welcome to Colab!
- Task
- + Section

```
from sklearn.tree import DecisionTreeClassifier
print("DecisionTreeClassifier imported successfully.")

# Instantiate SVM and Decision Tree models
svm_model = SVC(random_state=42)
dt_model = DecisionTreeClassifier(random_state=42)
print("SVM and Decision Tree models instantiated.")

# Evaluate SVM with K-fold cross-validation
print("\n--- SVM K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
    scores = cross_val_score(svm_model, X_train, y_train, cv=kfold, scoring=scorer)
    cv_results[f'svm_{metric_name}'] = scores
    print(f"SVM - KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")

# Evaluate SVM with Stratified K-fold cross-validation
print("\n--- SVM Stratified K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
    scores = cross_val_score(svm_model, X_train, y_train, cv=skfold, scoring=scorer)
    cv_results[f'svm_{metric_name}_stratified'] = scores
    print(f"SVM - Stratified KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")

# Evaluate Decision Tree with K-fold cross-validation
print("\n--- Decision Tree K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
    scores = cross_val_score(dt_model, X_train, y_train, cv=kfold, scoring=scorer)
    cv_results[f'dt_{metric_name}'] = scores
    print(f"Decision Tree - KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")

# Evaluate Decision Tree with Stratified K-fold cross-validation
print("\n--- Decision Tree Stratified K-fold Cross-Validation ---")
for metric_name, scorer in scoring.items():
    scores = cross_val_score(dt_model, X_train, y_train, cv=skfold, scoring=scorer)
    cv_results[f'dt_{metric_name}_stratified'] = scores
    print(f"Decision Tree - Stratified KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")
```

Variables Terminal

7:19 PM Python 3


(91) tamil songs - YouTube
Welcome To Colab - Colab
New Tab
colab.research.google.com/#scrollTo=db70e37a
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all Copy to Drive
RAM
Disk
Table of contents
Welcome to Colab!
Task
+ Section
from sklearn.model_selection import GridSearchCV
param_grid = {
'n_estimators': [10, 50, 100],
'max_depth': [None, 10, 20],
'min_samples_split': [2, 5, 10]
}
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=skfold, scoring='accuracy', n_jobs=-1)
grid_search.fit(X_train, y_train)
print("Best Hyperparameters:", grid_search.best_params_)
print("Best Score:", grid_search.best_score_)
Best Hyperparameters: {'max_depth': None, 'min_samples_split': 10, 'n_estimators': 50}
Best Score: 0.95
from sklearn.svm import SVC
print("SVC imported successfully.")
SVC imported successfully.
Task
Reasoning:
The next step in the plan is to implement K-fold and Stratified K-fold cross-validation for the Support Vector Machine (SVM) and Decision Tree models. I'll start by importing the DecisionTreeClassifier and instantiating both models. Then, I'll apply the previously defined KFold and StratifiedKFold objects along with the scoring dictionary to evaluate these models. This will allow us to calculate and report the accuracy, precision, recall, and F1-score for each model.
Variables
Terminal
7:19 PM
Python 3

(91) tamil songs - YouTube | Welcome To Colab - Colab | New Tab

colab.research.google.com/#scrollTo=b3dc3d10

Welcome To Colab — Cannot save changes

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all Copy to Drive

RAM Disk

Table of contents

- Welcome to Colab!
- Task
- + Section

```
[14] ✓ 5s
cv_results = {}
for metric_name, scorer in scoring.items():
    scores = cross_val_score(rf_model, X_train, y_train, cv=kfold, scoring=scorer)
    cv_results[metric_name] = scores
    print(f"Random Forest - KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")

print("K-fold cross-validation completed for Random Forest Classifier.")
```

```
Random Forest - KFold Accuracy: Mean = 0.9250, Std = 0.0553
Random Forest - KFold Precision: Mean = 0.9362, Std = 0.0556
Random Forest - KFold Recall: Mean = 0.9250, Std = 0.0553
Random Forest - KFold F1_score: Mean = 0.9261, Std = 0.0552
K-fold cross-validation completed for Random Forest Classifier.
```

```
[11] ✓ 0s
from sklearn.model_selection import StratifiedKFold
print("StratifiedKFold imported successfully.")
```

```
StratifiedKFold imported successfully.
```

```
[16] ✓ 0s
skfold = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
print("StratifiedKFold object instantiated successfully.")
```

```
StratifiedKFold object instantiated successfully.
```

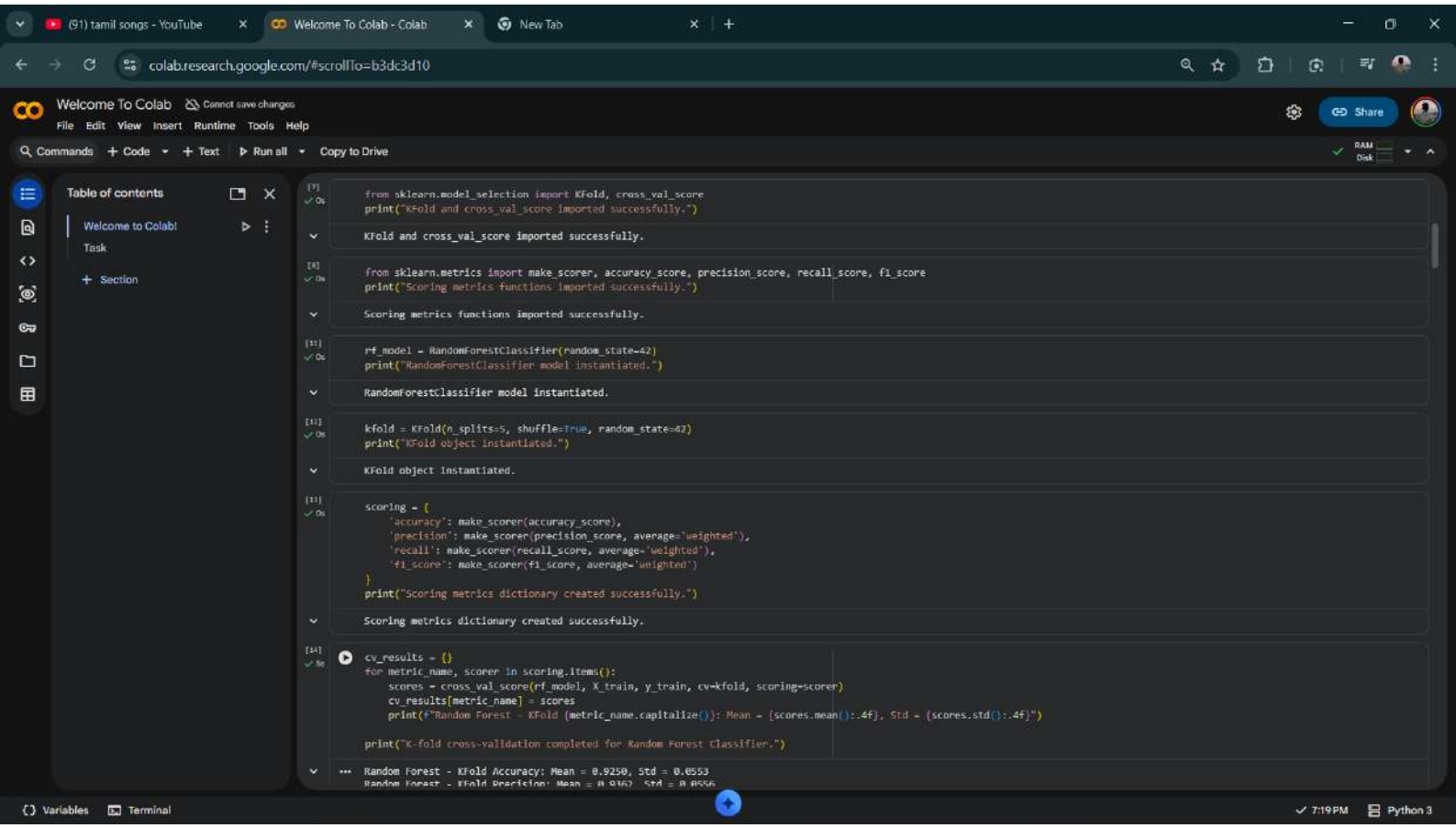
```
[17] ✓ 3s
for metric_name, scorer in scoring.items():
    scores = cross_val_score(rf_model, X_train, y_train, cv=skfold, scoring=scorer)
    cv_results[f'{metric_name}_stratified'] = scores
    print(f"Random Forest - Stratified KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")

print("Stratified K-fold cross-validation completed for Random Forest Classifier.")
```

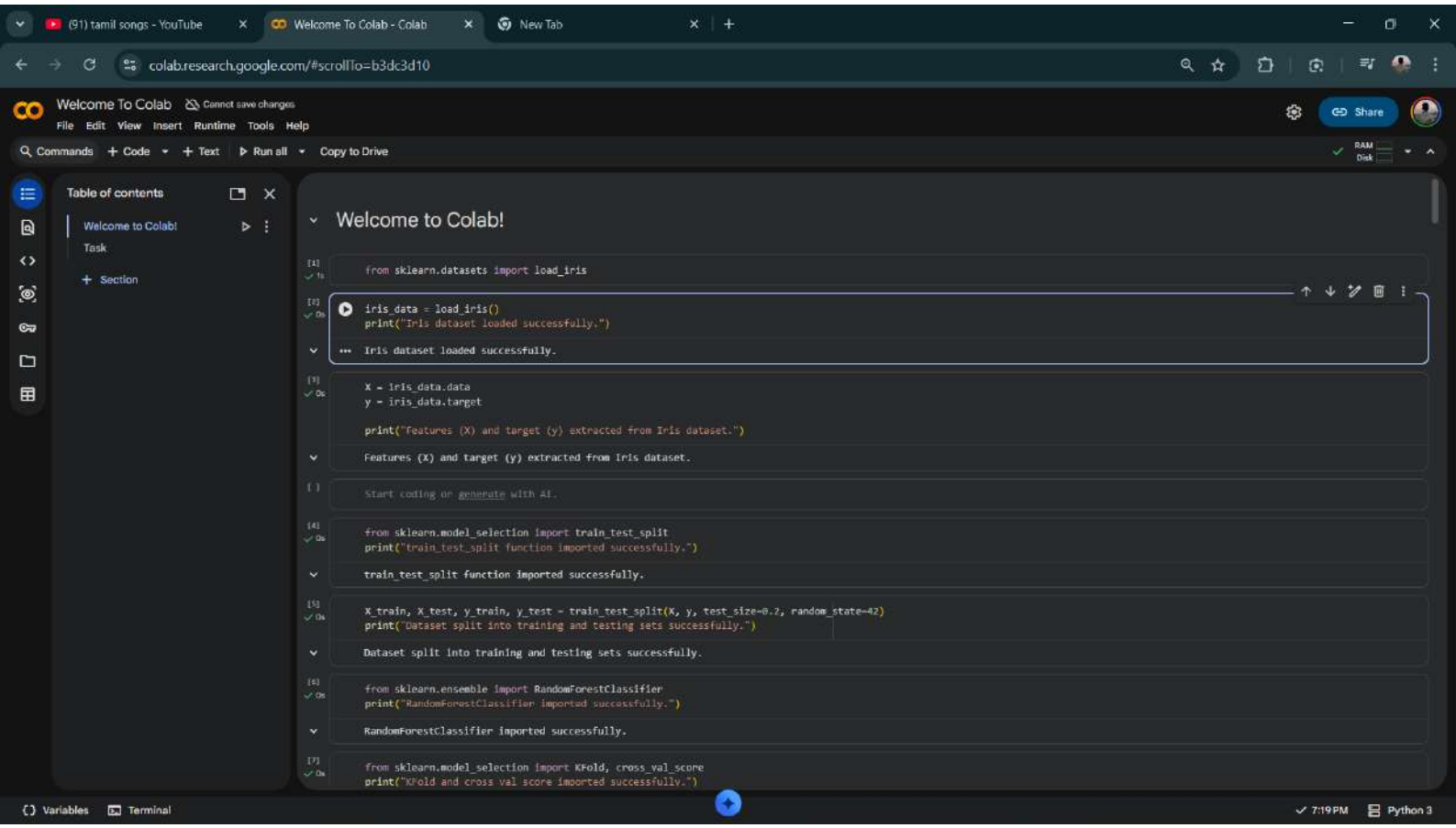
```
Random Forest - Stratified KFold Accuracy: Mean = 0.9333, Std = 0.0333
Random Forest - Stratified KFold Precision: Mean = 0.9464, Std = 0.0219
Random Forest - Stratified KFold Recall: Mean = 0.9333, Std = 0.0333
Random Forest - Stratified KFold F1_score: Mean = 0.9321, Std = 0.0350
Stratified K-fold cross-validation completed for Random Forest Classifier.
```

```
[18] ✓ 10s
from sklearn.model_selection import GridSearchCV

param_grid = {
    'n_estimators': [10, 50, 100],
```

Variables Terminal

7:19 PM Python 3


(91) tamil songs - YouTube
Welcome To Colab - Colab
New Tab
colab.research.google.com/#scrollTo=b3dc3d10
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all Copy to Drive
RAM
Disk
Table of contents
Welcome to Colab!
Task
+ Section
from sklearn.model_selection import KFold, cross_val_score
print("KFold and cross_val_score imported successfully.")
KFold and cross_val_score imported successfully.
from sklearn.metrics import make_scorer, accuracy_score, precision_score, recall_score, f1_score
print("Scoring metrics functions imported successfully.")
Scoring metrics functions imported successfully.
rf_model = RandomForestClassifier(random_state=42)
print("RandomForestClassifier model instantiated.")
RandomForestClassifier model instantiated.
kfold = KFold(n_splits=5, shuffle=True, random_state=42)
print("KFold object instantiated.")
KFold object instantiated.
scoring = {
'accuracy': make_scorer(accuracy_score),
'precision': make_scorer(precision_score, average='weighted'),
'recall': make_scorer(recall_score, average='weighted'),
'f1_score': make_scorer(f1_score, average='weighted')
}
print("Scoring metrics dictionary created successfully.")
Scoring metrics dictionary created successfully.
cv_results = {}
for metric_name, scorer in scoring.items():
scores = cross_val_score(rf_model, X_train, y_train, cv=kfold, scoring=scorer)
cv_results[metric_name] = scores
print(f"Random Forest - KFold {metric_name.capitalize()}: Mean = {scores.mean():.4f}, Std = {scores.std():.4f}")
print("K-fold cross-validation completed for Random Forest Classifier.")
Random Forest - KFold Accuracy: Mean = 0.9250, Std = 0.0553
Variables
Terminal
7:19 PM
Python 3


(91) tamil songs - YouTube
Welcome To Colab - Colab
New Tab
colab.research.google.com/#scrollTo=b3dc3d10
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all Copy to Drive
RAM
Disk
Table of contents
Welcome to Colab!
Task
+ Section
Welcome to Colab!
from sklearn.datasets import load_iris
iris_data = load_iris()
print("Iris dataset loaded successfully.")
Iris dataset loaded successfully.
X = iris_data.data
y = iris_data.target
print("Features (X) and target (y) extracted from Iris dataset.")
Features (X) and target (y) extracted from Iris dataset.
Start coding or generate with AI.
from sklearn.model_selection import train_test_split
print("train_test_split function imported successfully.")
train_test_split function imported successfully.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
print("Dataset split into training and testing sets successfully.")
Dataset split into training and testing sets successfully.
from sklearn.ensemble import RandomForestClassifier
print("RandomForestClassifier imported successfully.")
RandomForestClassifier imported successfully.
from sklearn.model_selection import KFold, cross_val_score
print("KFold and cross val score imported successfully.")
Variables
Terminal
7:19 PM
Python 3