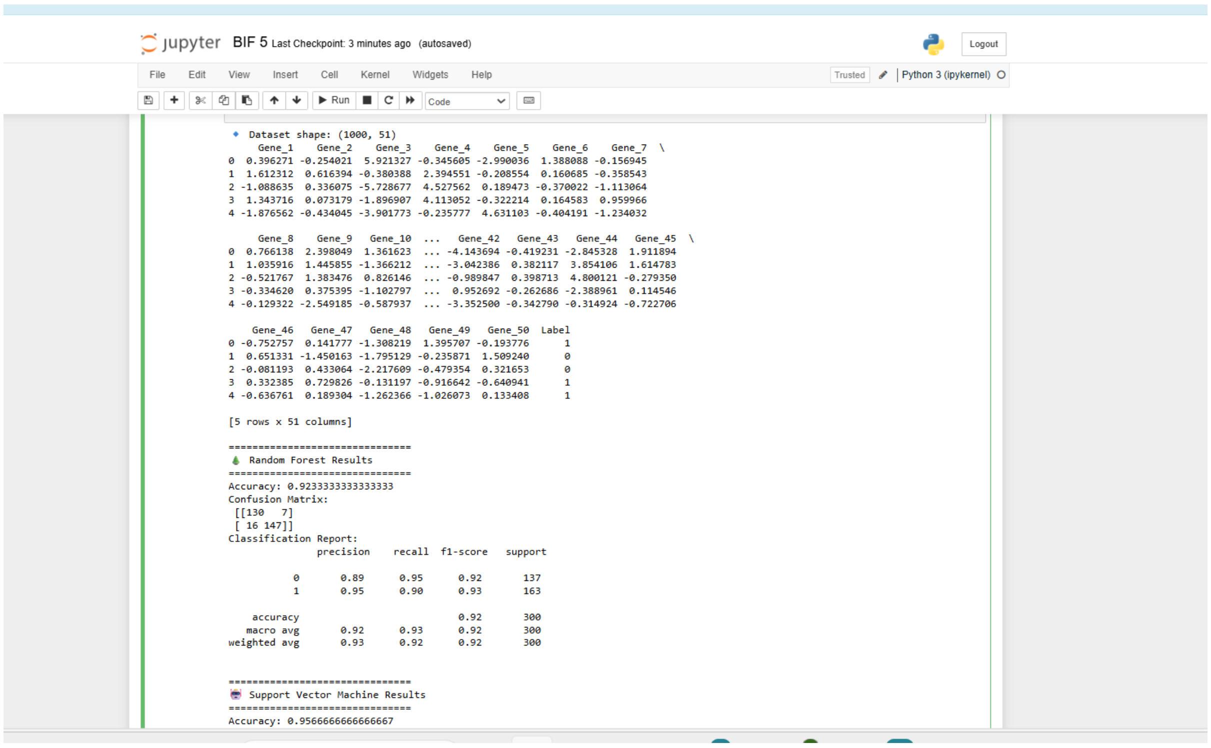
CODE :

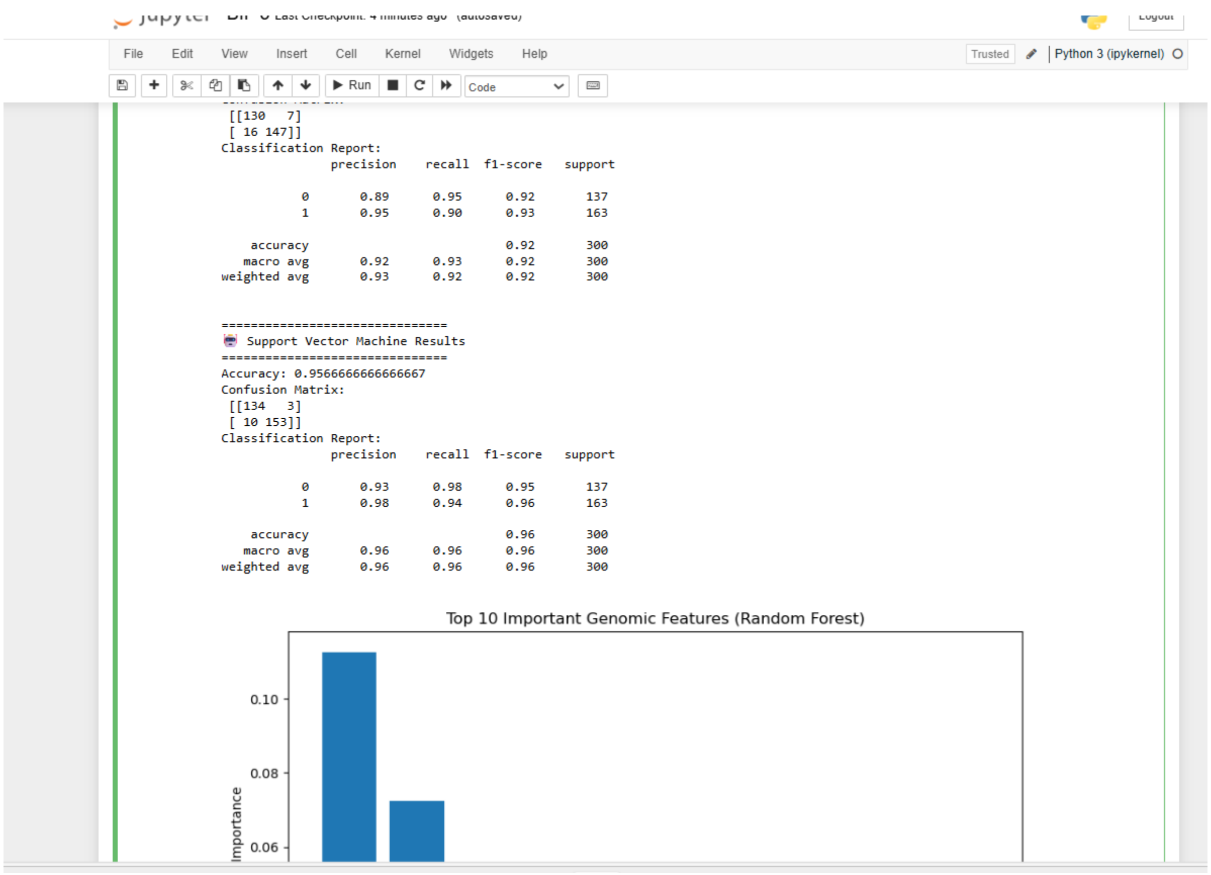
```
import numpy as np

import pandas as pd

from sklearn.datasets import make_classification

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

import matplotlib.pyplot as plt


# Suppose we have 1000 samples and 50 gene markers (features)

X, y = make_classification(

    n_samples=1000,

    n_features=50,

    n_informative=10,

    n_redundant=5,

    n_classes=2,

    random_state=42

)


# Convert to DataFrame for clarity

feature_names = [f"Gene_{i+1}" for i in range(X.shape[1])]

data = pd.DataFrame(X, columns=feature_names)

data['Label'] = y


print("🔹 Dataset shape:", data.shape)

print(data.head())
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# Train Random Forest Classifier
rf = RandomForestClassifier(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)

# Train Support Vector Machine Classifier
svm = SVC(kernel='rbf', probability=True, random_state=42)
svm.fit(X_train, y_train)

# Evaluate both models
rf_pred = rf.predict(X_test)
svm_pred = svm.predict(X_test)

print("\n================================")
print("🌲 Random Forest Results")
print("================================")
print("Accuracy:", accuracy_score(y_test, rf_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, rf_pred))
print("Classification Report:\n", classification_report(y_test, rf_pred))

print("\n================================")
print("🤖 Support Vector Machine Results")
print("================================")
print("Accuracy:", accuracy_score(y_test, svm_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, svm_pred))
print("Classification Report:\n", classification_report(y_test, svm_pred))

# Identify top predictive features (from Random Forest)
importances = rf.feature_importances_
indices = np.argsort(importances)[::-1]

plt.figure(figsize=(10,6))
plt.title("Top 10 Important Genomic Features (Random Forest)")
plt.bar(range(10), importances[indices[:10]], align="center")
plt.xticks(range(10), [feature_names[i] for i in indices[:10]], rotation=45)
plt.ylabel("Feature Importance")
plt.show()

# Summary
print("\n✅ Summary:")
print(f"Random Forest Accuracy: {accuracy_score(y_test, rf_pred):.3f}")
print(f"SVM Accuracy: {accuracy_score(y_test, svm_pred):.3f}")
print("Top 5 important genes:", [feature_names[i] for i in indices[:5]])
```

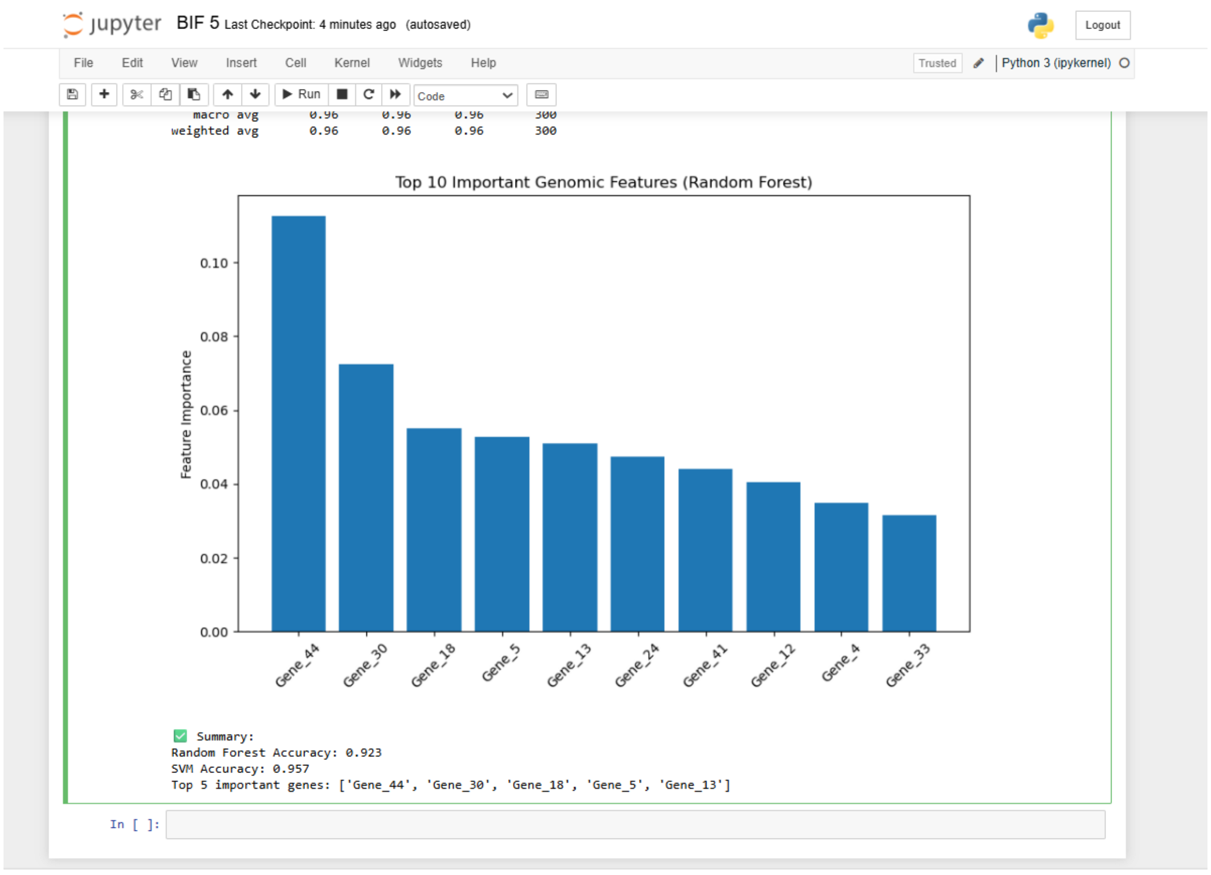
OUTPUT :



```
🔹 Dataset shape: (1000, 51)
     Gene_1    Gene_2    Gene_3    Gene_4    Gene_5    Gene_6    Gene_7  \
0  0.396271 -0.254021  5.921327 -0.345605 -2.990036  1.388088 -0.156945
1  1.612312  0.616394 -0.380388  2.394551 -0.208554  0.160685 -0.358543
2 -1.088635  0.336075 -5.728677  4.527562  0.189473 -0.370022 -1.113064
3  1.343716  0.073179 -1.896907  4.113052 -0.322214  0.164583  0.959966
4 -1.876562 -0.434045 -3.901773 -0.235777  4.631103 -0.404191 -1.234032

     Gene_8    Gene_9   Gene_10  ...   Gene_42   Gene_43   Gene_44   Gene_45  \
0  0.766138  2.398049  1.361623  ... -4.143694 -0.419231 -2.845328  1.911894
1  1.035916  1.445855 -1.366212  ... -3.042386  0.382117  3.854106  1.614783
2 -0.521767  1.383476  0.826146  ... -0.989847  0.398713  4.800121 -0.279350
3 -0.334620  0.375395 -1.102797  ...  0.952692 -0.262686 -2.388961  0.114546
4 -0.129322 -2.549185 -0.587937  ... -3.352500 -0.342790 -0.314924 -0.722706

    Gene_46   Gene_47   Gene_48   Gene_49   Gene_50  Label
0 -0.752757  0.141777 -1.308219  1.395707 -0.193776      1
1  0.651331 -1.450163 -1.795129 -0.235871  1.509240      0
2 -0.081193  0.433064 -2.217609 -0.479354  0.321653      0
3  0.332385  0.729826 -0.131197 -0.916642 -0.640941      1
4 -0.636761  0.189304 -1.262366 -1.026073  0.133408      1

[5 rows x 51 columns]

==============================
🌲 Random Forest Results
==============================
Accuracy: 0.9233333333333333
Confusion Matrix:
 [[130   7]
 [ 16 147]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       137
           1       0.95      0.90      0.93       163

    accuracy                           0.92       300
   macro avg       0.92      0.93      0.92       300
weighted avg       0.93      0.92      0.92       300


==============================
🤖 Support Vector Machine Results
==============================
Accuracy: 0.9566666666666667
```



```
 [[130   7]
 [ 16 147]]
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.95      0.92       137
           1       0.95      0.90      0.93       163

    accuracy                           0.92       300
   macro avg       0.92      0.93      0.92       300
weighted avg       0.93      0.92      0.92       300


==============================
🤖 Support Vector Machine Results
==============================
Accuracy: 0.9566666666666667
Confusion Matrix:
 [[134   3]
 [ 10 153]]
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.98      0.95       137
           1       0.98      0.94      0.96       163

    accuracy                           0.96       300
   macro avg       0.96      0.96      0.96       300
weighted avg       0.96      0.96      0.96       300
```



```
   macro avg       0.96      0.96      0.96       300
weighted avg       0.96      0.96      0.96       300
```

```
✅ Summary:
Random Forest Accuracy: 0.923
SVM Accuracy: 0.957
Top 5 important genes: ['Gene_44', 'Gene_30', 'Gene_18', 'Gene_5', 'Gene_13']
```

In [ ]: