Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=5BsmDThzyAcE&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

[ ]

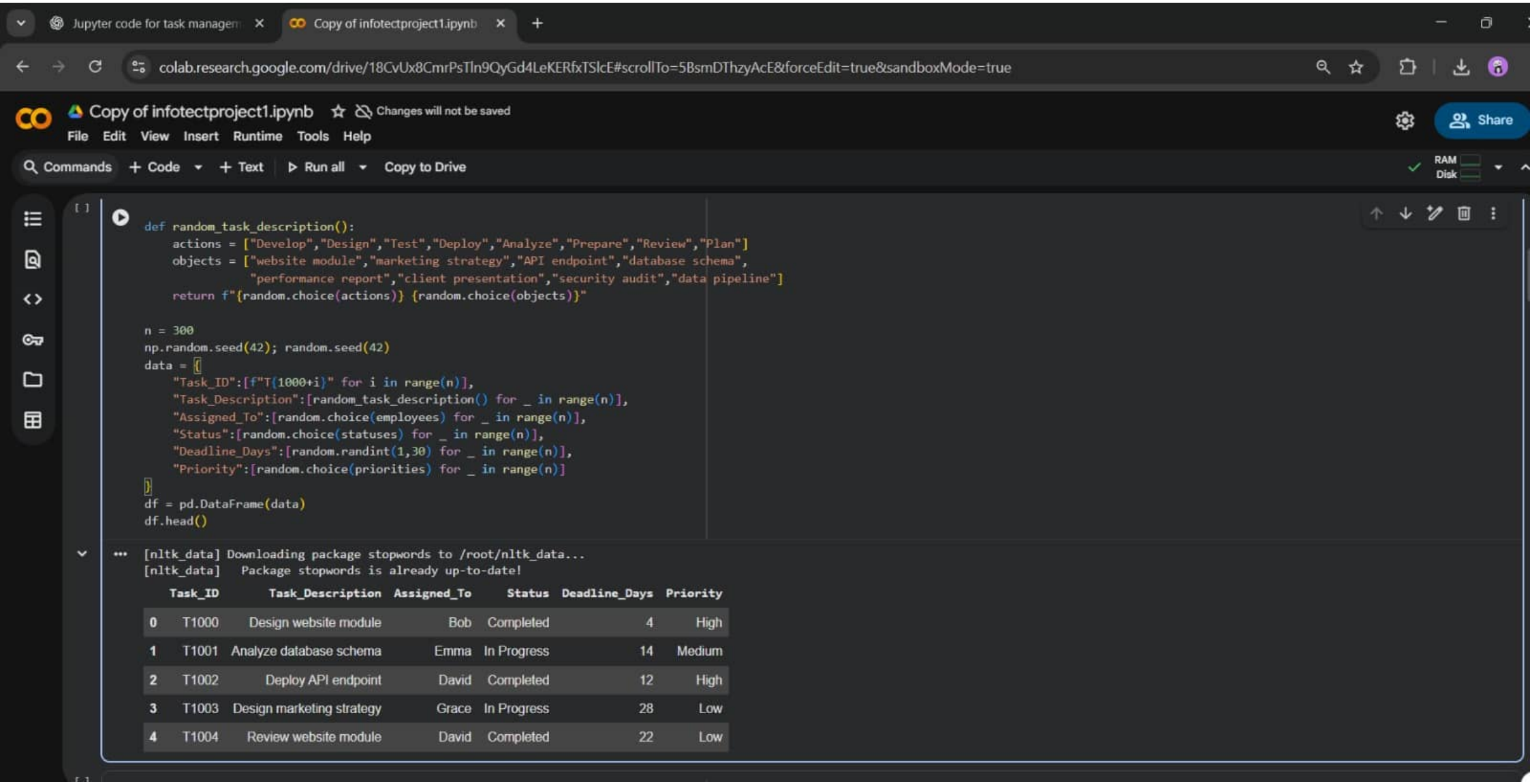
```
# ---- Week 1: Setup and Data Generation ----
!pip install wordcloud xgboost --quiet

import pandas as pd
import numpy as np
import random, string
import matplotlib.pyplot as plt
import seaborn as sns
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from wordcloud import WordCloud
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import accuracy_score, classification_report

nltk.download('stopwords')

# ----- Generate Synthetic Professional Dataset -----
employees = ["Alice","Bob","Charlie","David","Emma","Frank","Grace","Hannah"]
statuses = ["Pending","In Progress","Completed"]
priorities = ["Low","Medium","High"]

def random_task_description():
    actions = ["Develop","Design","Test","Deploy","Analyze","Prepare","Review","Plan"]
    objects = ["website module","marketing strategy","API endpoint","database schema",
               "performance report","client presentation","security audit","data pipeline"]
    return f"{random.choice(actions)} {random.choice(objects)}"
```



Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=5BsmDThzyAcE&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

```
# ---- Week 1: EDA ----
print("Dataset shape:", df.shape)
print(df.info())

sns.countplot(data=df, x='Priority', palette='viridis')
plt.title("Priority Distribution"); plt.show()

sns.countplot(data=df, x='Status', palette='magma')
plt.title("Task Status Distribution"); plt.show()

text = " ".join(df['Task_Description'])
wordcloud = WordCloud(width=800,height=400,background_color='white').generate(text)
plt.imshow(wordcloud, interpolation='bilinear'); plt.axis("off")
plt.title("WordCloud of Task Descriptions"); plt.show()
# ---- Week 1: Text Preprocessing ----
stop_words = set(stopwords.words('english'))
stemmer = PorterStemmer()

def preprocess_text(text):
    text = text.lower()
    text = ''.join([ch for ch in text if ch not in string.punctuation])
    tokens = text.split()
    tokens = [stemmer.stem(w) for w in tokens if w not in stop_words]
    return ' '.join(tokens)

df['Clean_Description'] = df['Task_Description'].apply(preprocess_text)
df[['Task_Description','Clean_Description']].head()
```

```
Dataset shape: (300, 6)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 300 entries, 0 to 299
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype
```

Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=5BsmDThzyAcE&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb — Changes will not be saved

File Edit View Insert Runtime Tools Help

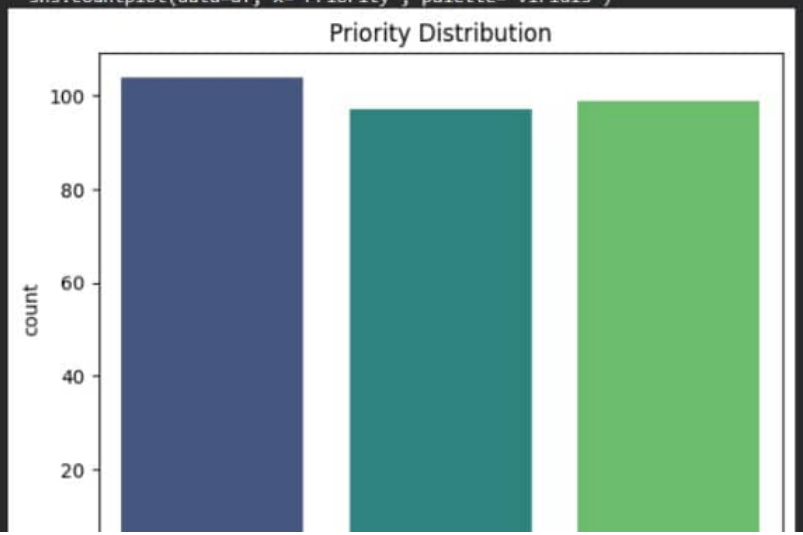
Commands + Code + Text Run all Copy to Drive

RAM Disk

```
 0   Task_ID          300 non-null    object
 1   Task_Description  300 non-null    object
 2   Assigned_To       300 non-null    object
 3   Status            300 non-null    object
 4   Deadline_Days     300 non-null    int64
 5   Priority          300 non-null    object
dtypes: int64(1), object(5)
memory usage: 14.2+ KB
None
/tmp/ipython-input-23361538.py:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='Priority', palette='viridis')
```



Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=5BsmDThzyAcE&forceEdit=true&sandboxMode=true

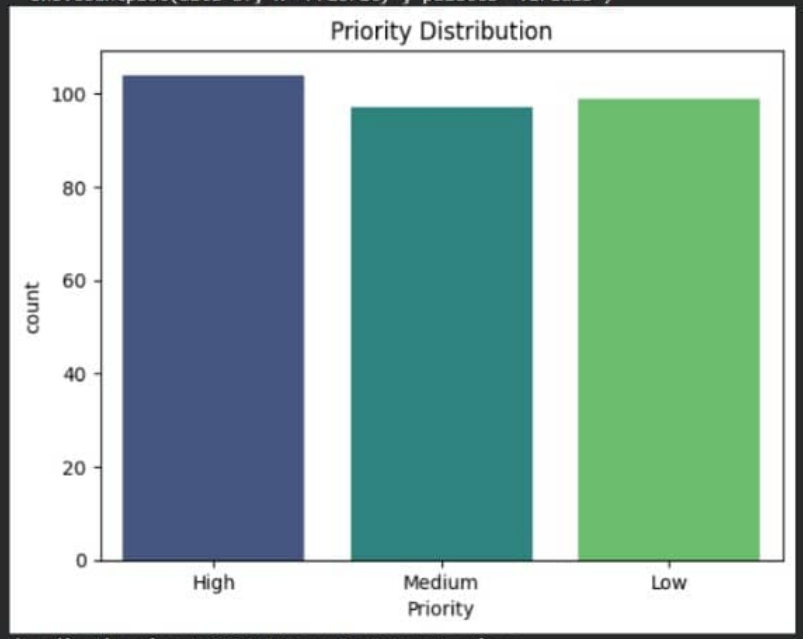
Copy of infotectproject1.ipynb | Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

RAM Disk

```
sns.countplot(data=df, x='Priority', palette='viridis')
```




```
/tmp/ipython-input-23361538.py:8: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='Status', palette='magma')
```

Jupyter code for task managem | Copy of infotectproject1.ipynb

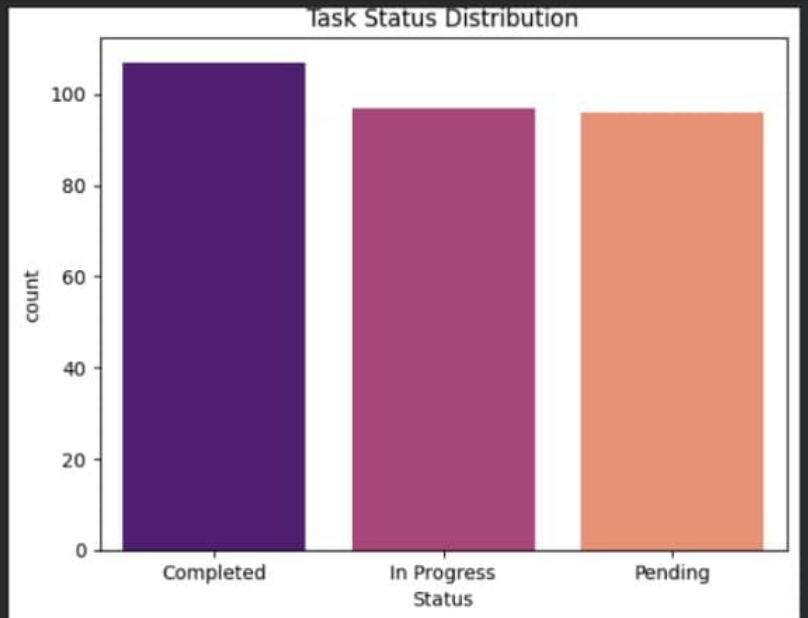
colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=UX6WMajy0EX1&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

RAM Disk



Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=UX6WMajy0EX1&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

	Task_Description	Clean_Description
0	Design website module	design websit modul
1	Analyze database schema	analyz databas schema
2	Deploy API endpoint	deploy api endpoint
3	Design marketing strategy	design market strategi
4	Review website module	review websit modul

Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=UX6WMajy0EX1&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

```
# ---- Week 2: Task Classification (Text) ----
priority_map = {'Low':0,'Medium':1,'High':2}
df['Priority_Level'] = df['Priority'].map(priority_map)

X = df['Clean_Description']
y = df['Priority_Level']

tfidf = TfidfVectorizer(max_features=500)
X_tfidf = tfidf.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, y, test_size=0.2, random_state=42)

# Naive Bayes
nb = MultinomialNB().fit(X_train, y_train)
y_pred_nb = nb.predict(X_test)
print("Naive Bayes Accuracy:", accuracy_score(y_test, y_pred_nb))
print(classification_report(y_test, y_pred_nb))

# SVM
svm = SVC(kernel='linear', C=1).fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)
print("SVM Accuracy:", accuracy_score(y_test, y_pred_svm))
print(classification_report(y_test, y_pred_svm))
```

```
Naive Bayes Accuracy: 0.2833333333333333
              precision    recall  f1-score   support

           0       0.33      0.17      0.23        23
           1       0.32      0.29      0.30        21
           2       0.24      0.44      0.31        16

    accuracy                           0.28        60
   macro avg       0.30      0.30      0.28        60
weighted avg       0.30      0.28      0.28        60
```

Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=UX6WMajy0EX1&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

```
SVM Accuracy: 0.36666666666666664
              precision    recall  f1-score   support

           0       0.47      0.30      0.37        23
           1       0.55      0.29      0.38        21
           2       0.26      0.56      0.36        16

    accuracy                           0.37        60
   macro avg       0.43      0.38      0.37        60
weighted avg       0.44      0.37      0.37        60
```

[ ]

```
# ---- Week 3: Priority Prediction (Structured Data) ----
df['Status_Code'] = df['Status'].map({'Pending':0,'In Progress':1,'Completed':2})
X_struct = df[['Deadline_Days','Status_Code']]
y_struct = df['Priority_Level']

X_train_s, X_test_s, y_train_s, y_test_s = train_test_split(X_struct, y_struct, test_size=0.2, random_state=42)

# Random Forest
rf = RandomForestClassifier(n_estimators=100, random_state=42).fit(X_train_s, y_train_s)
y_pred_rf = rf.predict(X_test_s)
print("Random Forest Accuracy:", accuracy_score(y_test_s, y_pred_rf))
print(classification_report(y_test_s, y_pred_rf))

# XGBoost
xgb = XGBClassifier(use_label_encoder=False, eval_metric='mlogloss', random_state=42)
xgb.fit(X_train_s, y_train_s)
y_pred_xgb = xgb.predict(X_test_s)
print("XGBoost Accuracy:", accuracy_score(y_test_s, y_pred_xgb))
print(classification_report(y_test_s, y_pred_xgb))
```

```
Random Forest Accuracy: 0.36666666666666664
```

Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=UX6WMajy0EX1&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

```
                precision    recall  f1-score   support

           0       0.28      0.22      0.24        23
           1       0.33      0.29      0.31        21
           2       0.46      0.69      0.55        16

    accuracy                           0.37        60
   macro avg       0.36      0.40      0.37        60
weighted avg       0.35      0.37      0.35        60

XGBoost Accuracy: 0.36666666666666664
              precision    recall  f1-score   support

           0       0.28      0.22      0.24        23
           1       0.30      0.29      0.29        21
           2       0.50      0.69      0.58        16

    accuracy                           0.37        60
   macro avg       0.36      0.40      0.37        60
weighted avg       0.34      0.37      0.35        60

/usr/local/lib/python3.12/dist-packages/xgboost/training.py:199: UserWarning: [16:10:38] WARNING: /workspace/src/learner.cc:790:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)
```

[ ]

```
# ---- Week 4: Workload Balancing & Dashboard ----
workload = df.groupby('Assigned_To')['Task_ID'].count().sort_values(ascending=False)
plt.figure(figsize=(8,5))
sns.barplot(x=workload.index, y=workload.values, palette='cool')
plt.title("Employee Workload Distribution"); plt.ylabel("Number of Tasks"); plt.show()

avg_load = workload.mean()
over = workload[workload > avg_load + 2]
under = workload[workload < avg_load - 2]
```

Jupyter code for task managen | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=UX6WMajy0EX1&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

```
[ ]
    avg_load = workload.mean()
    over = workload[workload > avg_load + 2]
    under = workload[workload < avg_load - 2]

    print("\nOverloaded:", list(over.index))
    print("Underloaded:", list(under.index))
    if len(over)>0 and len(under)>0:
        print("\nSuggested Transfers:")
        for i in range(min(len(over), len(under))):
            print(f"Transfer 1 task from {over.index[i]} → {under.index[i]}")
    # ---- Final Dashboard Summary ----
    models = ['Naive Bayes','SVM','Random Forest','XGBoost']
    scores = [
        accuracy_score(y_test, y_pred_nb),
        accuracy_score(y_test, y_pred_svm),
        accuracy_score(y_test_s, y_pred_rf),
        accuracy_score(y_test_s, y_pred_xgb)
    ]

    print("\n===== 📊 PROJECT SUMMARY =====")
    for m,s in zip(models, scores):
        print(f"{m}: {s:.2f}")

    plt.figure(figsize=(7,4))
    sns.barplot(x=models, y=scores, palette='plasma')
    plt.title("Model Comparison - Accuracy"); plt.ylim(0,1); plt.show()
```

```
/tmp/ipython-input-3251324710.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=workload.index, y=workload.values, palette='cool')
```

Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=lj94TgCUznf3&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

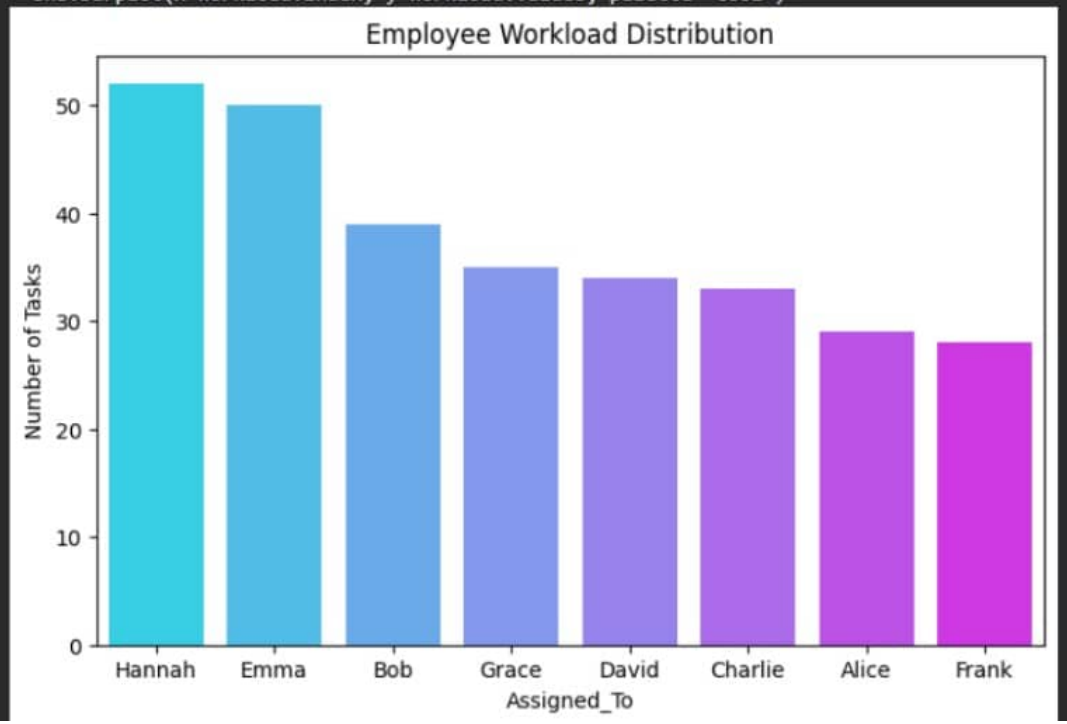
Commands + Code + Text ▷ Run all Copy to Drive

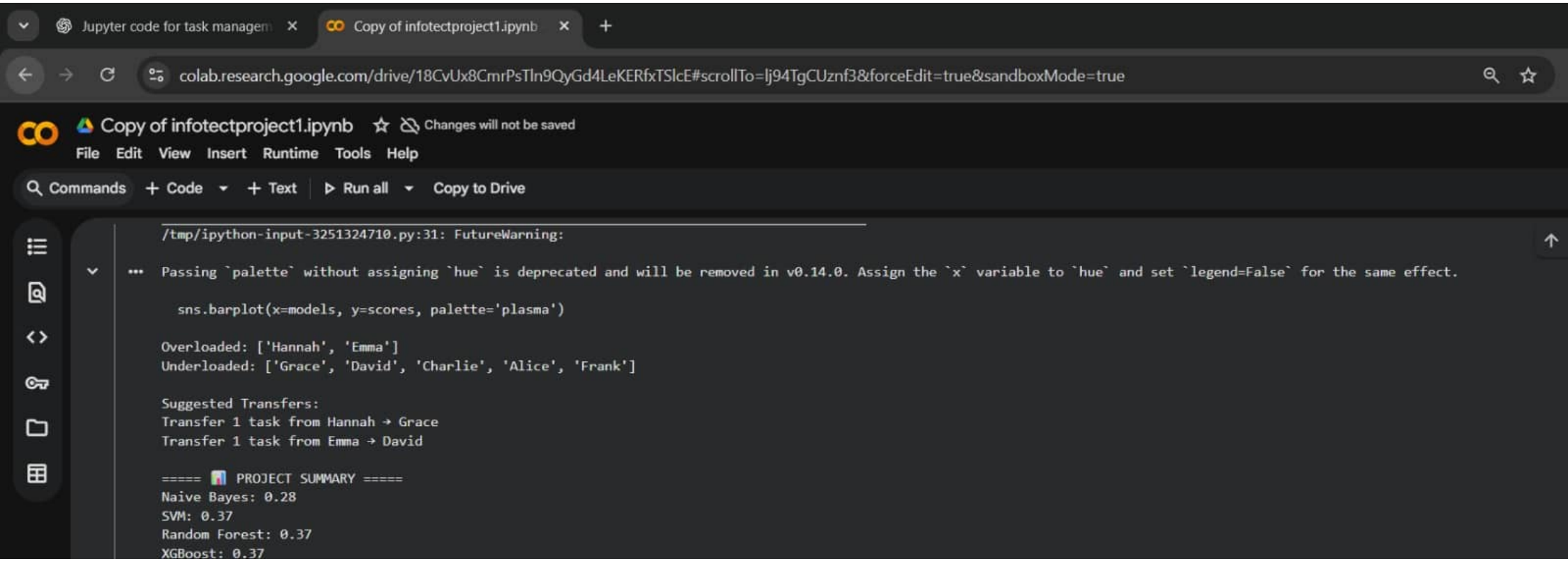
```
sns.barplot(x=models, y=scores, palette='plasma')
plt.title("Model Comparison - Accuracy"); plt.ylim(0,1); plt.show()
```

```
/tmp/ipython-input-3251324710.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=workload.index, y=workload.values, palette='cool')
```





Jupyter code for task managem | Copy of infotectproject1.ipynb

colab.research.google.com/drive/18CvUx8CmrPsTln9QyGd4LeKERfxTSlcE#scrollTo=lj94TgCUznf3&forceEdit=true&sandboxMode=true

Copy of infotectproject1.ipynb ☆ Changes will not be saved

File Edit View Insert Runtime Tools Help

Commands + Code + Text ▷ Run all Copy to Drive

```
    sns.barplot(x=models, y=scores, palette='plasma')
```

```
Overloaded: ['Hannah', 'Emma']
Underloaded: ['Grace', 'David', 'Charlie', 'Alice', 'Frank']

Suggested Transfers:
Transfer 1 task from Hannah → Grace
Transfer 1 task from Emma → David

===== 📊 PROJECT SUMMARY =====
Naive Bayes: 0.28
SVM: 0.37
Random Forest: 0.37
XGBoost: 0.37
```