AI Impact on Management Practices

By : Irfan Ullah Khan

Everything is being changed by artificial intelligence (AI) at breakneck speed in almost every business, management being no exception. This paper evaluates AI's effects on management in relation to how it complements or otherwise impacts decision-making, organizational structure, and efficient functioning. It reviews existing literature, cites in-practice examples, and discusses challenges and future trends in integrating AI into management.

📘 AI Impact on Management Practices - A to Z Analysis

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import os
import joblib
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, ConfusionMatrixDisplay
%matplotlib inline
```

In [ ]:
```
df = pd.read_csv("ai_management_data.csv")
df.head()
```

Out[ ]:

	ID	Industry	Department	AI_Impact_Score	Impact_Type	Comment
0	1	Finance	Sales	2.26	Neutral	Improved data-driven decisions
1	2	Healthcare	R&D	3.80	Positive	Improved efficiency
2	3	Education	Sales	6.66	Negative	Automated tasks
3	4	Finance	Operations	3.78	Positive	Resistance from staff
4	5	Manufacturing	IT	8.00	Neutral	Improved efficiency

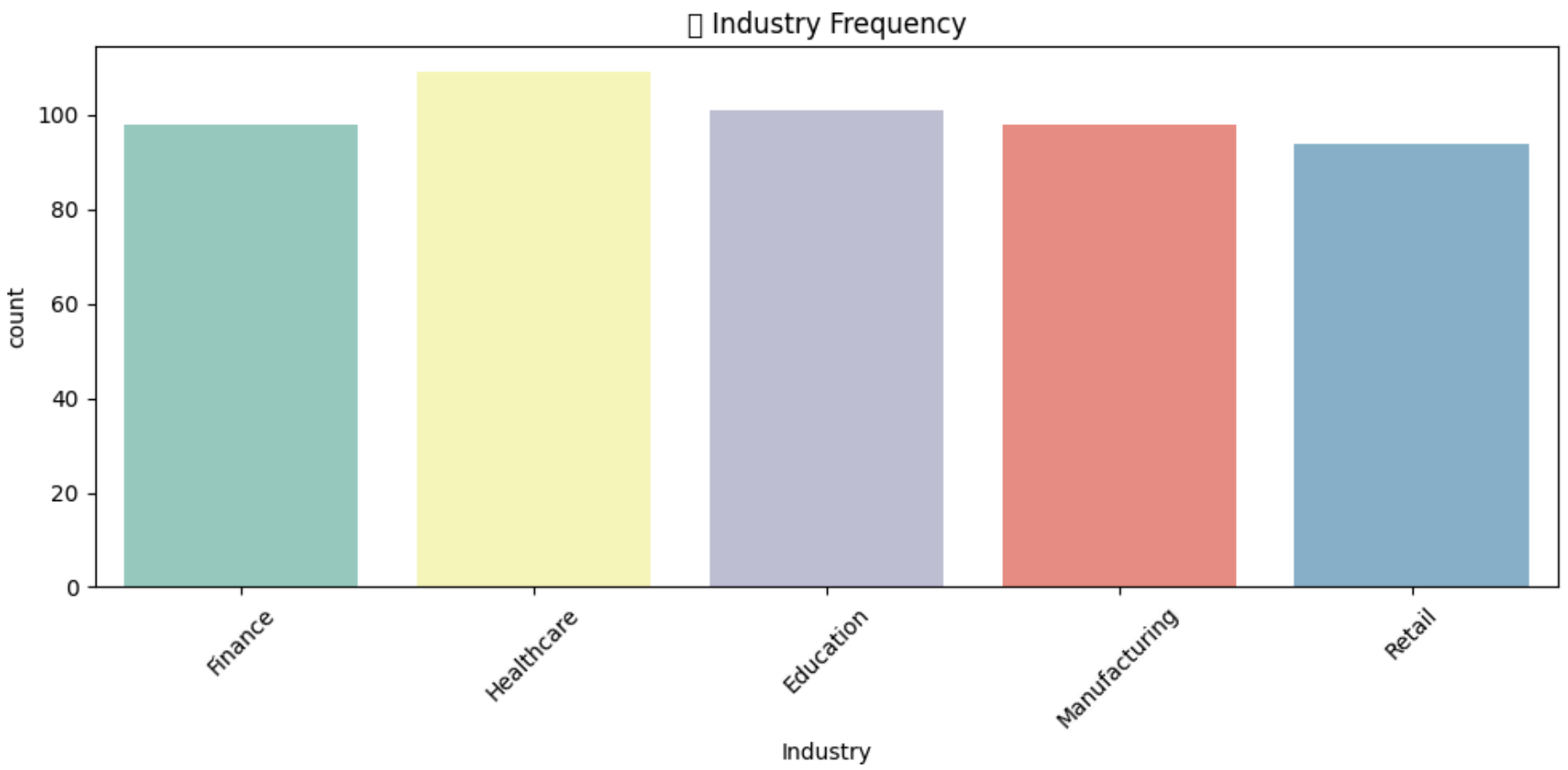
In [ ]:
```
print("Shape:", df.shape)
print("Missing values:\n", df.isnull().sum())
print("Data types:\n", df.dtypes)
print("Unique values:\n", df.nunique())
```

```
Shape: (500, 6)
Missing values:
 ID                 0
Industry           0
Department         0
AI_Impact_Score    0
Impact_Type        0
Comment            0
dtype: int64
Data types:
 ID                   int64
Industry            object
Department          object
AI_Impact_Score    float64
Impact_Type         object
Comment             object
dtype: object
Unique values:
 ID                 500
Industry             5
Department           6
AI_Impact_Score    388
Impact_Type          3
Comment              6
dtype: int64
```

📊 Industry Frequency

In [ ]:
```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Industry', palette='Set3')
plt.title("📊 Industry Frequency")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-4-3460862283.py:2: FutureWarning:

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

  sns.countplot(data=df, x='Industry', palette='Set3')
/tmp/ipython-input-4-3460862283.py:5: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from
font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```

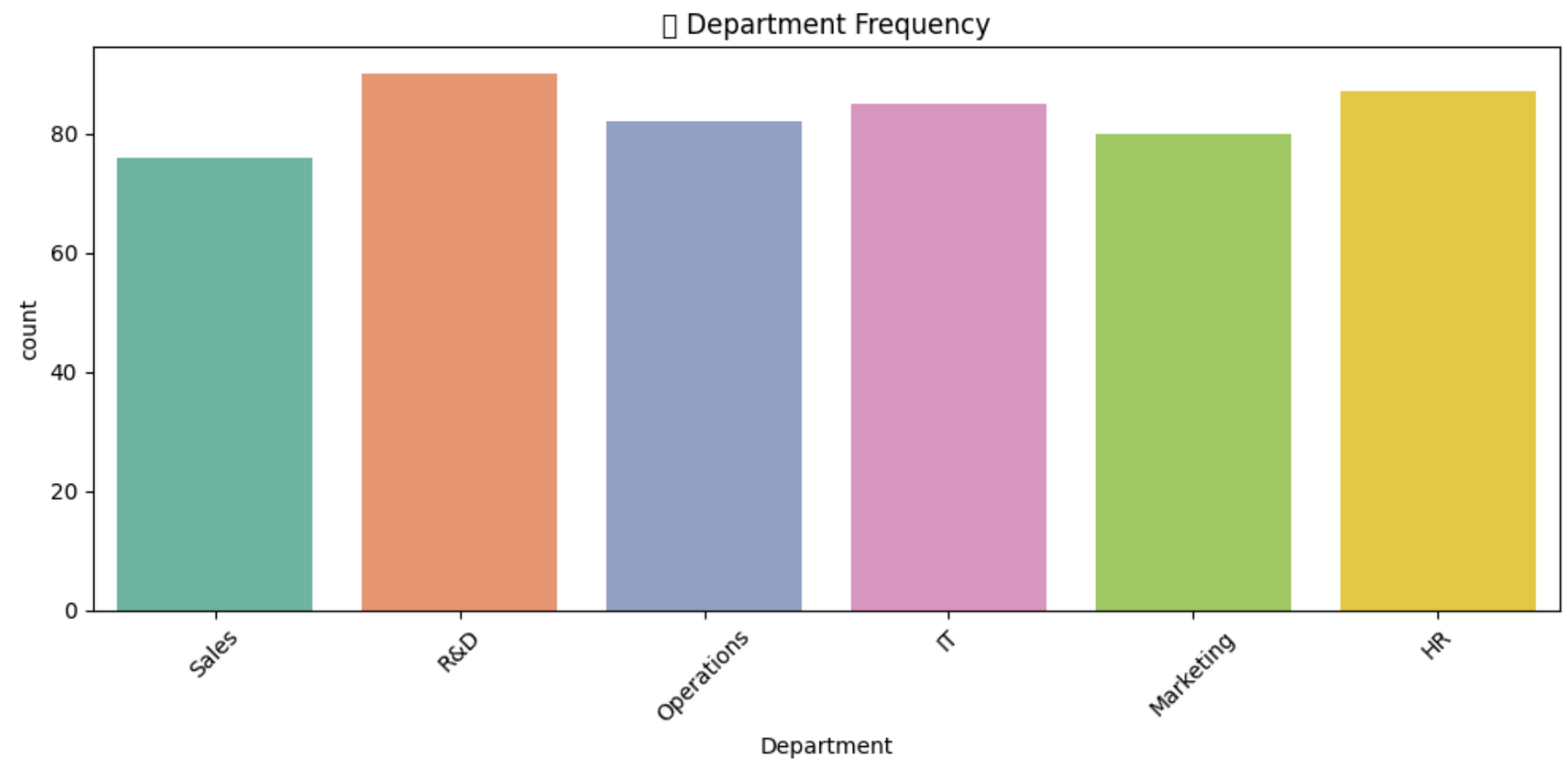


📊 Department Frequency

In [ ]:
```
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Department', palette='Set2')
plt.title("📊 Department Frequency")
plt.xticks(rotation=45)
```

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

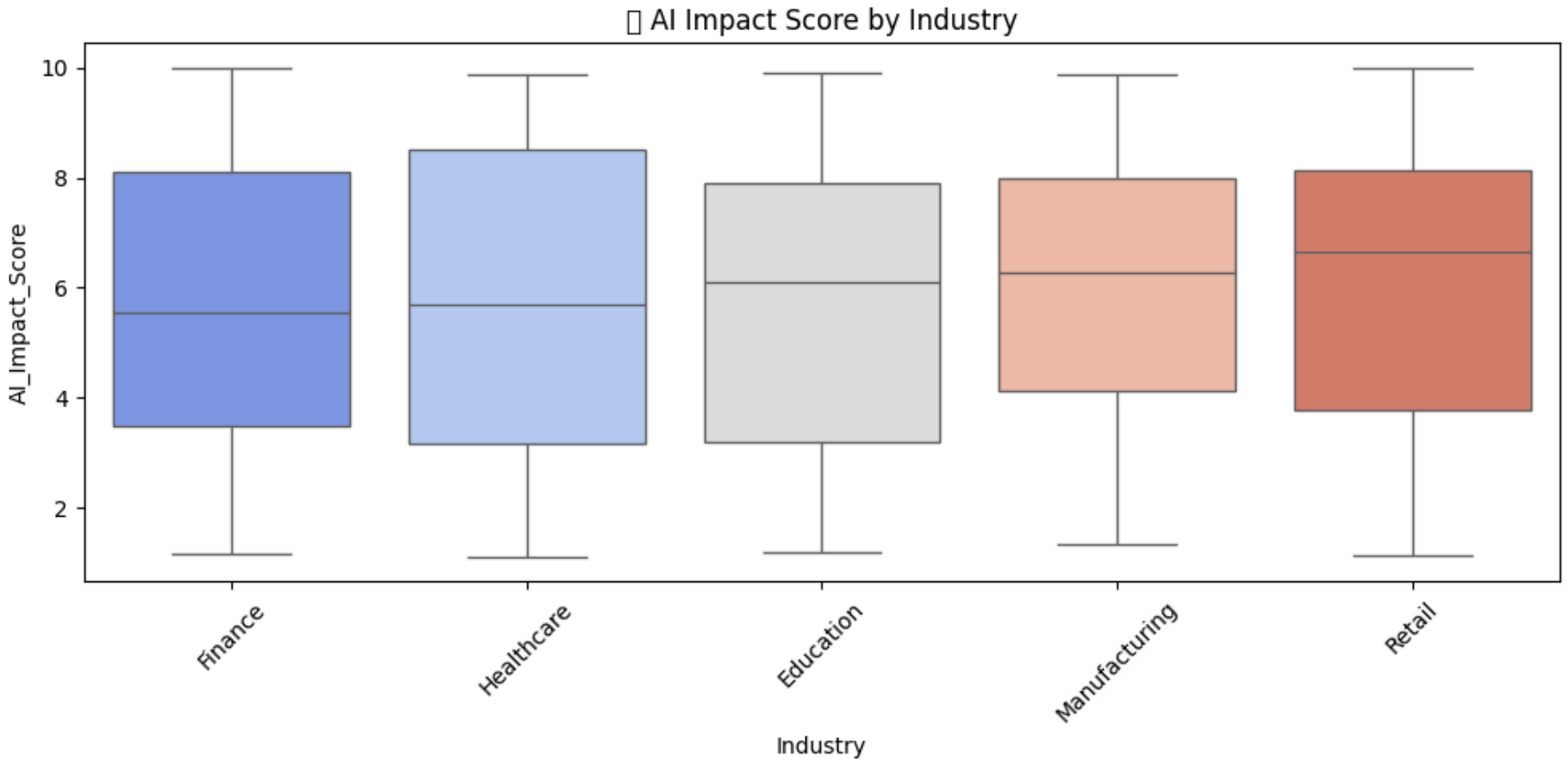
```
/tmp/ipython-input-5-4237692145.py:2: FutureWarning:

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

  sns.countplot(data=df, x='Department', palette='Set2')
/tmp/ipython-input-5-4237692145.py:5: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from
font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```



📊 AI Impact Score by Industry

In [ ]:
```
plt.figure(figsize=(10, 5))
sns.boxplot(data=df, x='Industry', y='AI_Impact_Score', palette='coolwarm')
plt.title("📊 AI Impact Score by Industry")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

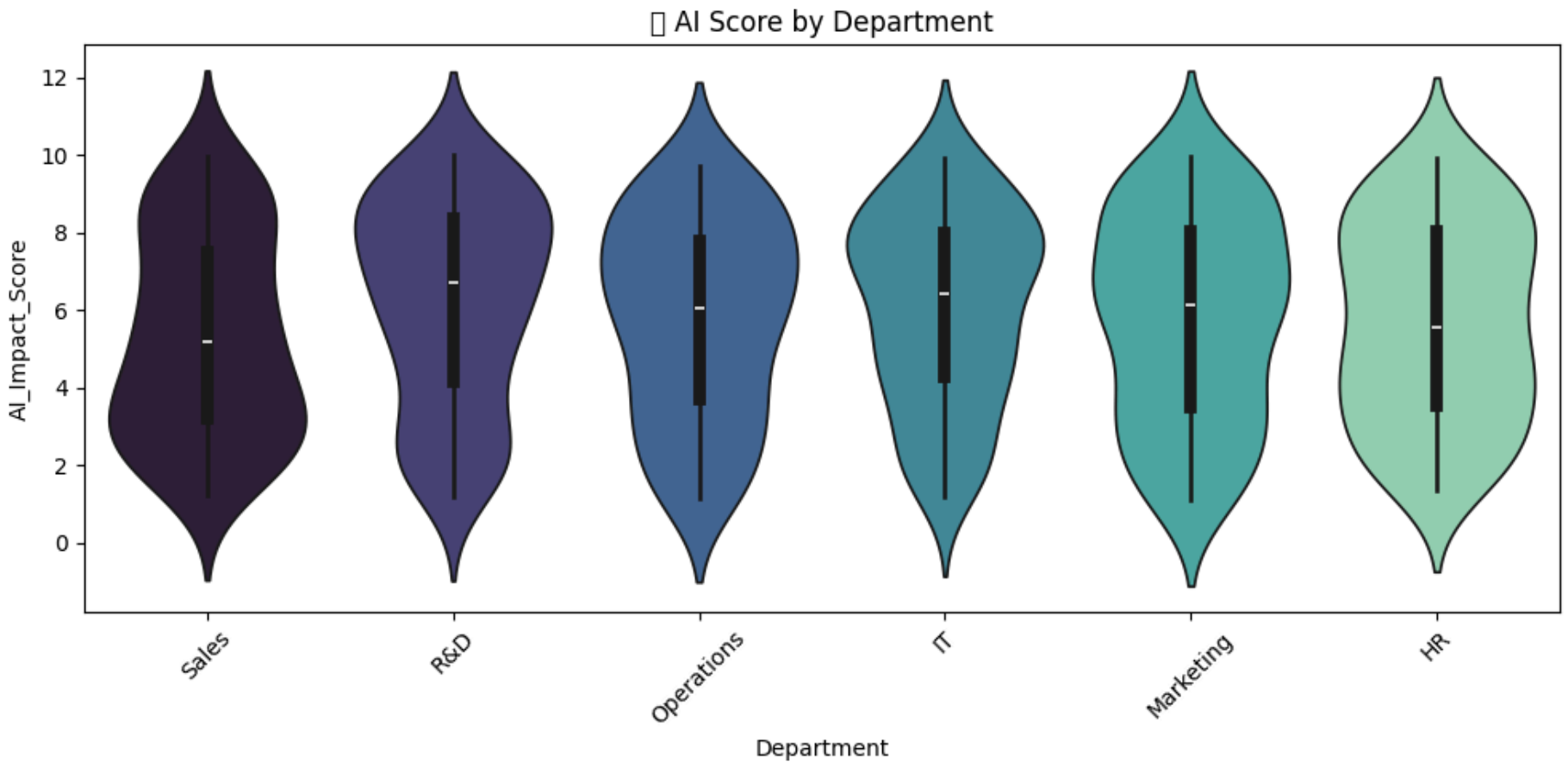
```
/tmp/ipython-input-6-841417123.py:2: FutureWarning:

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

  sns.boxplot(data=df, x='Industry', y='AI_Impact_Score', palette='coolwarm')
/tmp/ipython-input-6-841417123.py:5: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from
font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```



AI Score by Department

In [ ]:
```
plt.figure(figsize=(10, 5))
sns.violinplot(data=df, x='Department', y='AI_Impact_Score', palette='mako')
plt.title("AI Score by Department")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

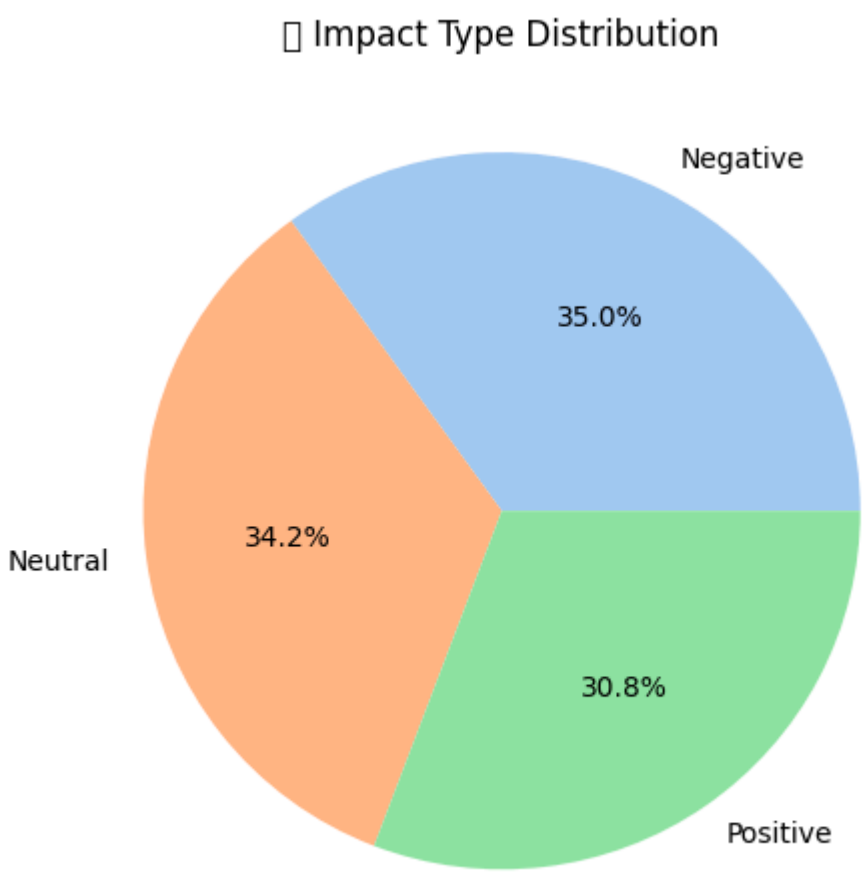
```
/tmp/ipython-input-7-1730640150.py:2: FutureWarning:

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

  sns.violinplot(data=df, x='Department', y='AI_Impact_Score', palette='mako')
/tmp/ipython-input-7-1730640150.py:5: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from
font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```



Impact Type Distribution

In [ ]:
```
plt.figure(figsize=(10, 5))
df['Impact_Type'].value_counts().plot.pie(autopct='%1.1f%%', colors=sns.color_palette('pastel')); plt.ylabel('')
plt.title("Impact Type Distribution")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

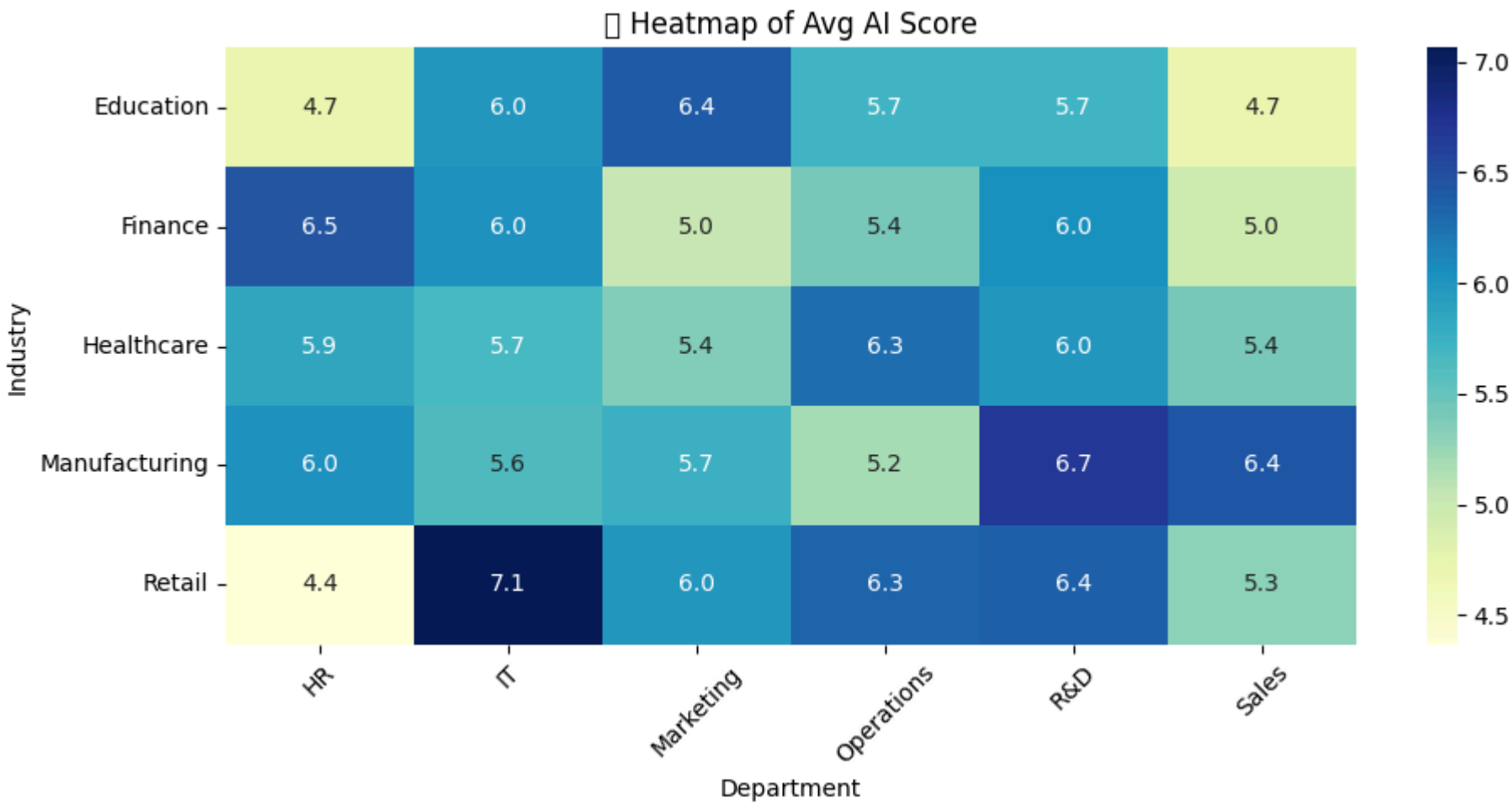
```
/tmp/ipython-input-8-1681166839.py:5: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```



📊 Heatmap of Avg AI Score

In [ ]:
```
plt.figure(figsize=(10, 5))

pivot = df.pivot_table(index="Industry", columns="Department", values="AI_Impact_Score", aggfunc="mean")
sns.heatmap(pivot, annot=True, cmap="YlGnBu", fmt=".1f")

plt.title("📊 Heatmap of Avg AI Score")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-9-36013258.py:8: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  plt.tight_layout()
/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 128202 (\N{BAR CHART}) missing from font(s) DejaVu Sans.
  fig.canvas.print_figure(bytes_io, **kw)
```



🌞 Sunburst Chart

In [ ]:
```
fig = px.sunburst(df, path=['Industry', 'Department', 'Impact_Type'],
                  values='AI_Impact_Score', color='AI_Impact_Score',
                  color_continuous_scale='Blues',
                  title="Sunburst Chart: AI Score Hierarchy")
fig.show()
```

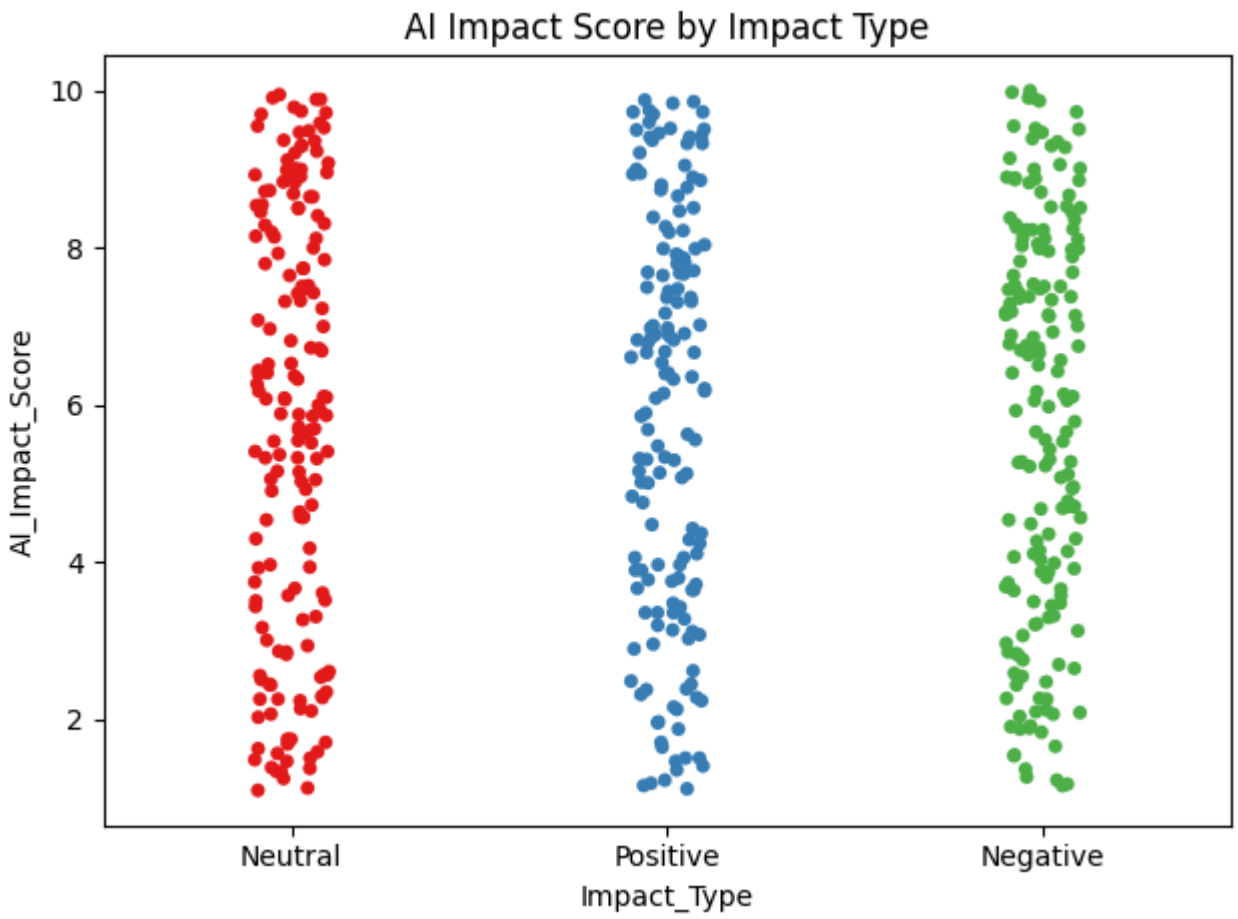
🧱 Treemap Chart

In [ ]:
```
fig = px.treemap(df, path=['Industry', 'Department', 'Impact_Type'],
                 values='AI_Impact_Score', color='AI_Impact_Score',
                 color_continuous_scale='RdBu',
                 title="Treemap: AI Impact Score")
fig.show()
```

🎯 Strip Plot

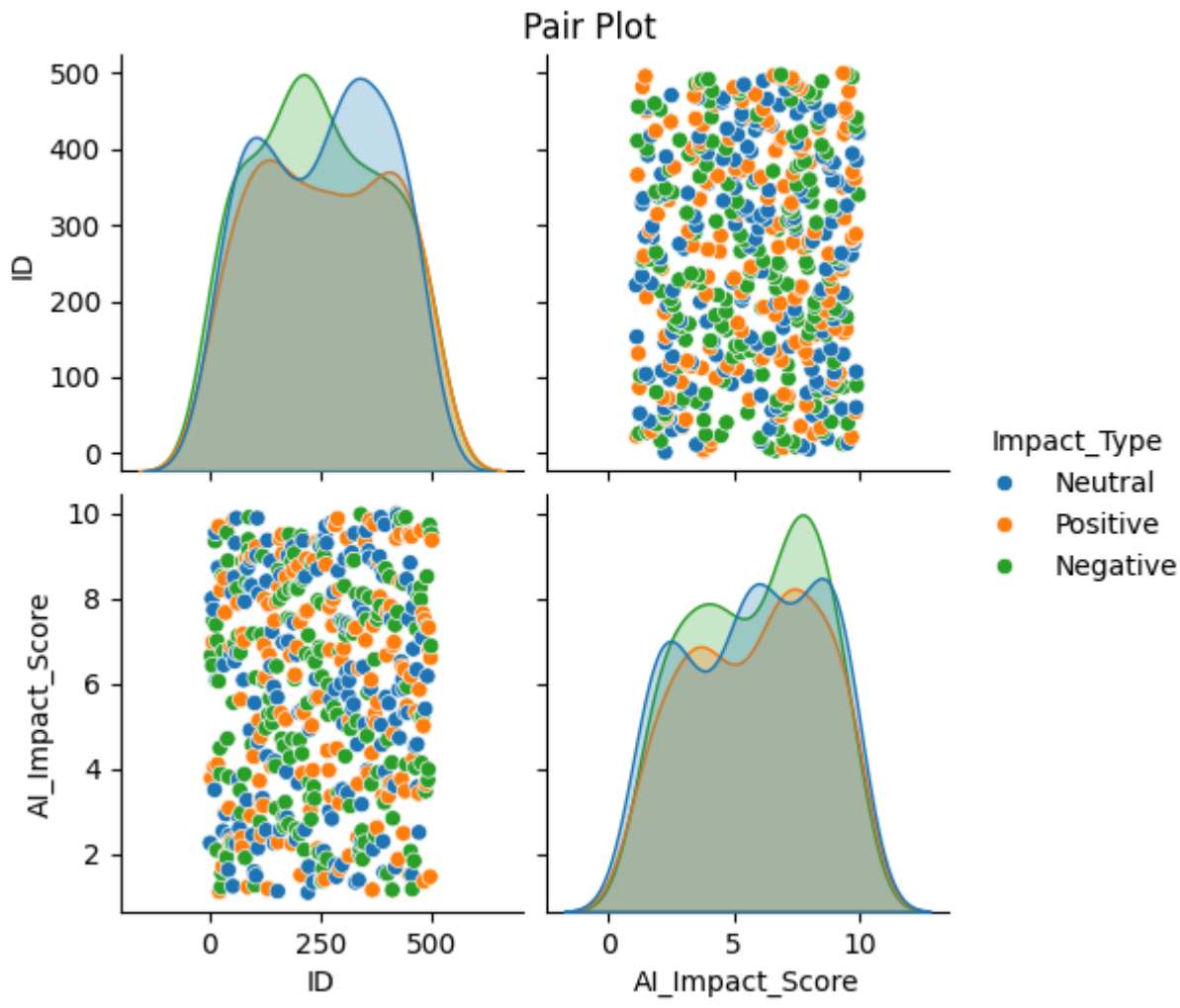
In [ ]:
```
sns.stripplot(data=df, x="Impact_Type", y="AI_Impact_Score", jitter=True, palette="Set1")
plt.title("AI Impact Score by Impact Type")
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-12-2210075936.py:1: FutureWarning:


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



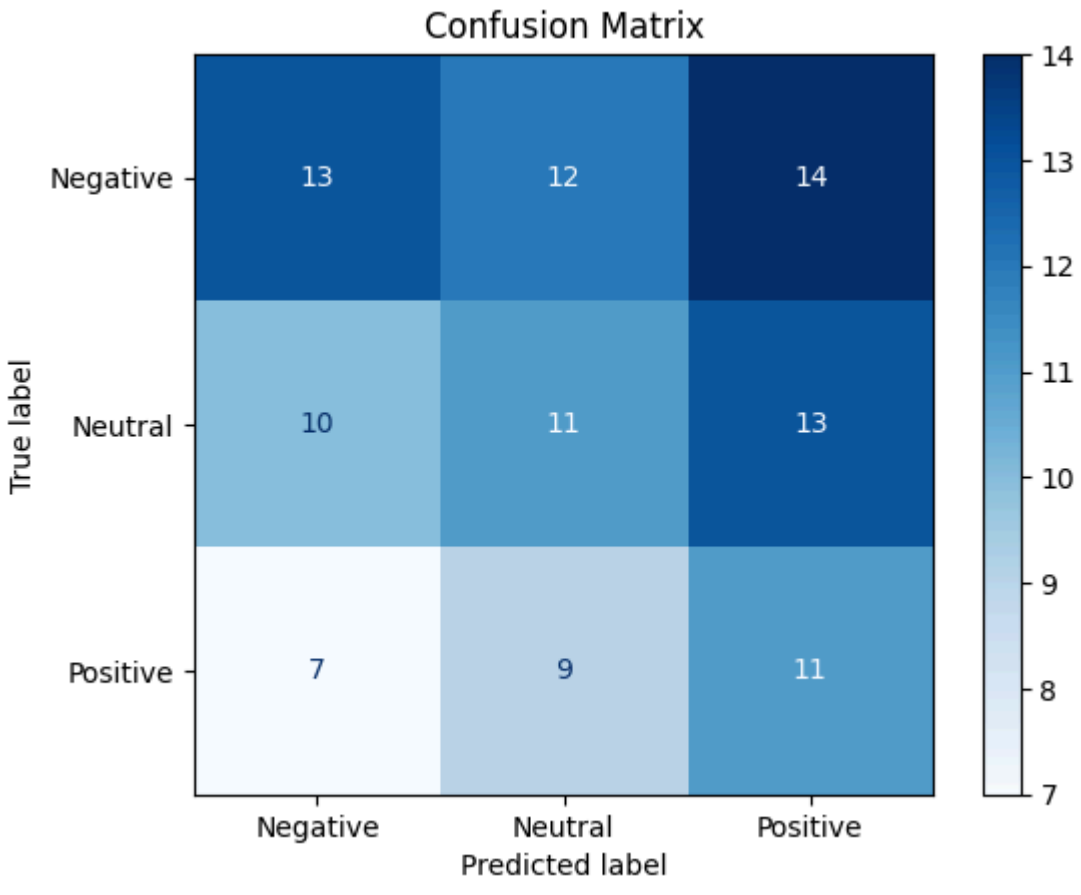
Pair Plot

In [ ]:
```
sns.pairplot(df, hue="Impact_Type")
plt.suptitle("Pair Plot", y=1.02)
plt.show()
```



Confusion Matrix

In [ ]:
```
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=le_target.classes_)
disp.plot(cmap="Blues")
plt.title("Confusion Matrix")
plt.show()
```



💾 Save Model & Encoders

In [ ]:
```
joblib.dump(model, "ai_impact_model.pkl")
joblib.dump(le_target, "label_encoder_target.pkl")
joblib.dump(le_industry, "label_encoder_industry.pkl")
joblib.dump(le_department, "label_encoder_department.pkl")
joblib.dump(le_comment, "label_encoder_comment.pkl")
```

Out[ ]:
```
['label_encoder_comment.pkl']
```

👨‍💻 By: Irfan Ullah Khan

GITHUB PROFILE KAGGLE PROFILE LINKEDIN PROFILE

YOUTUBE PROFILE EMAIL CONTACT ME WEBSITE CONTACT ME