Data Science Salaries Classification (2023)

By: Muhammad Yusuf Royhan

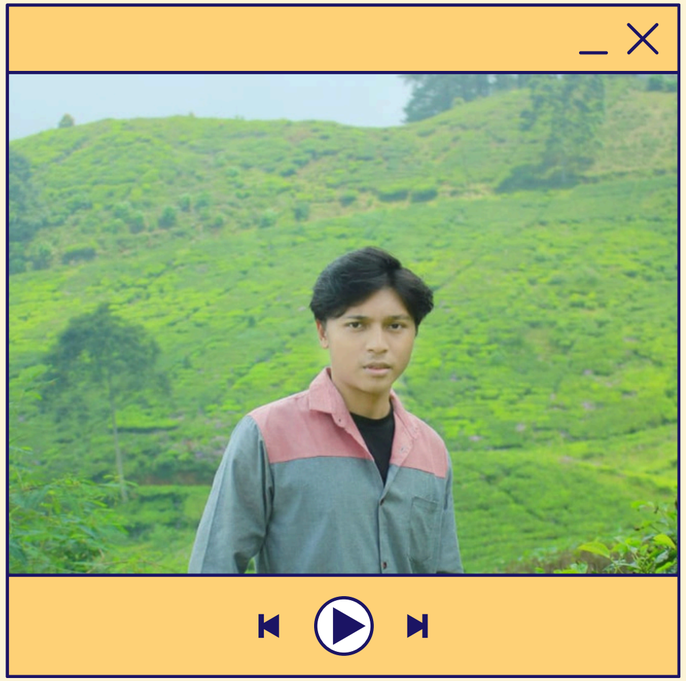
Portfolio - Project Digital Skill Fair 35.0 Data Science

MUHAMMAD YUSUF ROYHAN

I am an Electrical Engineering graduate from Cendekia Abditama University, with strong analytical skills in data processing. Currently, I am ambitious to develop my potential and switch my career to the field of Data Science, specifically as a Data Analyst, Data Scientist, or Data Engineer.

I have skills that support this transition, including mastery of Microsoft Office, Python, Power BI, Tableau, Google Colab, and SQL. These skills enable me to effectively analyse and visualise data, as well as provide reliable insights for data-driven decision-making.

With a solid technical background and a deep interest in data science, I am ready to contribute significantly to a dynamic team. I believe that the combination of my education, analytical skills, and desire for continuous learning will make me a valuable asset to the organisation I join. I am highly motivated to collaborate on challenging projects and provide innovative solutions to challenges in the data world.



INTRODUCTION!

DATASET FROM KAGGLE = Data-Science-Salaries-Classification-2023

The ‘Data Science Salaries Classification 2023’ dataset available on Kaggle contains information about the salaries received by data science and analytics professionals in 2023. The main purpose of this dataset is to provide insight into the salaries offered to workers across different positions, experience levels, locations, and company types.

This dataset is presented in CSV format and consists of several columns, including years of employment, experience level (such as Senior, Mid-level, Entry-level, and Executive), job type (Full-time, Contract, Freelance), specific job title (such as Data Scientist, Machine Learning Engineer, and Data Analyst), as well as information on salary in local currency and US dollars.

📊 Initial Dataset Exploration Results

- Data count: 3.755 rows, 11 columns
- Data type:
  - 4 numeric columns (work_year, salary, salary_in_usd, remote_ratio)
  - 7 category columns (experience_level, employment_type, job_title, salary_currency, employee_residence, company_location, company_size)
- No missing values 🎉
- Salary in USD (salary_in_usd) has a large range, from $5,132 to $450,000 with Salary Classification (Low, Medium, High).

DATA DESCRIPTION!

DATASET FROM KAGGLE = Data-Science-Salaries-Classification-2023

No	Column Name	Meaning
1	work_year	Representing the specific year of salary data collection.
2	Experience_level	The level of work experience of the employees, categorized as EN (Entry-Level), EX (Experienced), MI (Mid-Level), SE (Senior).
3	Employment_type	The type of employment, labelled as FT (Full-Time), CT (Contractor), FL (Freelancer), PT (Part-Time).
4	Job_title	The job titles of the employees, such as "Applied Scientist", "Data Quality Analyst", etc.
5	Salary	The salary figures in their respective currency formats.
6	Salary_currency	The currency code representing the salary.
7	Salary_in_usd	The converted salary figures in USD for uniform comparison.
8	Company_location	The location of the companies, specified as country codes (e.g., "US" for the United States and "NG" for Nigeria).
9	Company_size	The size of the companies, classified as "L" (Large), "M" (Medium), and "S" (Small).

TOOLS USED

Google Colaboratory

pandas

kaggle

scikits learn machine learning

seaborn

NumPy

python™

matplotlib

FLOWCHART !

Start

Import Library

Load Dataset

Data Preprocessing

Exploratory Data Analysis

Split Dataset

Modelling Random Forest

Evaluation

Finish

UPLOAD & EKSTRAK FILE ZIP TO GOOGLE COLAB

+ Code + Text

30s

```python
# Step 1: Upload ZIP file ke Google Colab
from google.colab import files
import zipfile
import pandas as pd

# Remove the path argument, files.upload() expects no arguments to open the upload dialog.
uploaded = files.upload()  # Ini akan membuka dialog untuk mengunggah file ZIP

# Step 2: Ekstrak file ZIP
zip_filename = list(uploaded.keys())[0]  # Ambil nama file ZIP yang diunggah
extract_folder = "extracted_data"

with zipfile.ZipFile(zip_filename, 'r') as zip_ref:
    zip_ref.extractall(extract_folder)  # Ekstrak ke folder
```

```
Choose Files Data Scien...s (2023).zip
• Data Science Salaries (2023).zip(application/zip) - 26034 bytes, last modified: 1/29/2025 - 100% done
Saving Data Science Salaries (2023).zip to Data Science Salaries (2023).zip
```

IMPORTING LIBRARIES!

```
[2]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     import seaborn as sns
     from sklearn.datasets import make_classification
     from sklearn.linear_model import LogisticRegression
     from sklearn.model_selection import train_test_split
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score
     from sklearn.metrics import confusion_matrix
```

Alongside these are some of the Python libraries that we used to build our salary classification model for data science.

LOAD DATASET! - EXPLORATORY DATA ANALYSIS

```
+ Code + Text

import pandas as pd

# Ensure the path is correct, it's assumed your CSV file is named 'ds_salaries.csv':
csv_path = "extracted_data/ds_salaries.csv"
df = pd.read_csv(csv_path)

# Now you can use 'df' as your DataFrame for the rest of the code.

# Get the feature names from the DataFrame's columns
feature_names = df.columns[:-1]  # Exclude the target column

X = df[feature_names].values   # inputan untuk machine learning
y = df["salary_in_usd"].values  # output yang dinginkan dari machine learning

# Mengonversi data fitur dan target menjadi DataFrame
df_X = pd.DataFrame(X, columns=feature_names)
df_y = pd.Series(y, name='target')

# Gabungkan fitur dan target dalam satu DataFrame
df = pd.concat([df_X, df_y], axis=1)

df.head(10)
```

In this Data Science Salaries there are several Input Variables, I took only 10 rows from the feature and target data that has been merged into the DataFrame

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	target
0	2023	Senior	Full-Time	Principal Data Scientist	80000	EUR	85847	ES	100	ES	85847
1	2023	Mid-Level	Contractor	ML Engineer	30000	USD	30000	US	100	US	30000
2	2023	Mid-Level	Contractor	ML Engineer	25500	USD	25500	US	100	US	25500
3	2023	Senior	Full-Time	Data Scientist	175000	USD	175000	CA	100	CA	175000
4	2023	Senior	Full-Time	Data Scientist	120000	USD	120000	CA	100	CA	120000
5	2023	Senior	Full-Time	Applied Scientist	222200	USD	222200	US	0	US	222200
6	2023	Senior	Full-Time	Applied Scientist	136000	USD	136000	US	0	US	136000
7	2023	Senior	Full-Time	Data Scientist	219000	USD	219000	CA	0	CA	219000
8	2023	Senior	Full-Time	Data Scientist	141000	USD	141000	CA	0	CA	141000

EXPLORATORY DATA ANALYSIS (EDA) ✨

SOME INFORMATION

```
# Step 5: Tampilkan informasi dataset
print("Informasi Dataset:")
print(df.info())
```

```
Informasi Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3755 entries, 0 to 3754
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   work_year           3755 non-null   int64
 1   experience_level    3755 non-null   object
 2   employment_type     3755 non-null   object
 3   job_title           3755 non-null   object
 4   salary              3755 non-null   int64
 5   salary_currency     3755 non-null   object
 6   salary_in_usd       3755 non-null   int64
 7   employee_residence  3755 non-null   object
 8   remote_ratio        3755 non-null   int64
 9   company_location    3755 non-null   object
 10  company_size        3755 non-null   object
dtypes: int64(4), object(7)
memory usage: 322.8+ KB
None
```

CHECK MISSING VALUES

```
df.isnull().sum()
```

	0
work_year	0
experience_level	0
employment_type	0
job_title	0
salary	0
salary_currency	0
salary_in_usd	0
employee_residence	0
remote_ratio	0
company_location	0
target	0

dtype: int64

EXPLORATORY DATA ANALYSIS (EDA) ✨

STATISTIC

```
df.describe()
```

	target
count	3755.000000
mean	137570.389880
std	63055.625278
min	5132.000000
25%	95000.000000
50%	135000.000000
75%	175000.000000
max	450000.000000

SALARY

```
# Step 6: Buat kategori gaji (Low, Medium, High)
def categorize_salary(salary):
    if salary < 100000:
        return "Low"
    elif salary <= 175000:
        return "Medium"
    else:
        return "High"

df["salary_category"] = df["salary_in_usd"].apply(categorize_salary)

# Step 7: Tampilkan distribusi kategori gaji
print("\nDistribusi Kategori Gaji:")
print(df["salary_category"].value_counts())
```

```
Distribusi Kategori Gaji:
salary_category
Medium    1832
Low        991
High       932
Name: count, dtype: int64
```

DATA PREPROCESSING ✨

+ Code + Text

```
# ✅ Check if all selected features are present in the DataFrame columns
missing_features = [feature for feature in selected_features if feature not in df.columns]
if missing_features:
    # ✅ If features are missing, print the available columns and raise the KeyError
    print(f"Available columns in the DataFrame: {df.columns.tolist()}")
    raise KeyError(f"The following features are not in the DataFrame: {missing_features}")

X = df[selected_features]  # Fitur (independent variables)
y = df["salary_category"]  # Target (dependent variable)

# One-Hot Encoding untuk "job_title" (karena banyak kategorinya)
X = pd.get_dummies(X, columns=["job_title"])

# 3️⃣ Membagi dataset menjadi training & testing (80-20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

# 4️⃣ Normalisasi (Opsional - hanya jika pakai KNN atau SVM)
scaler = StandardScaler()
X_train[["salary_in_usd", "remote_ratio"]] = scaler.fit_transform(X_train[["salary_in_usd", "remote_ratio"]])
X_test[["salary_in_usd", "remote_ratio"]] = scaler.transform(X_test[["salary_in_usd", "remote_ratio"]])

# Cek hasil akhir
print(f"Shape X_train: {X_train.shape}, Shape X_test: {X_test.shape}")
print("Data preprocessing selesai! 🚀")
```

```
Shape X_train: (3004, 98), Shape X_test: (751, 98)
Data preprocessing selesai! 🚀
```

SPLIT DATA & TRAIN THE MODEL ✨

```python
from sklearn.model_selection import train_test_split

# Membagi data menjadi train dan test
X_train, X_test, y_train, y_test = train_test_split(df_X, df_y, test_size=0.2, random_state=69)
```

MODEL LOGISTIC REGRESSION

LogisticRegression

```
LogisticRegression(max_iter=1000, random_state=69)
```

MODEL RANDOM FOREST

RandomForestClassifier

```
RandomForestClassifier(random_state=69)
```

MODEL SVC

SVC

```
SVC(kernel='linear', random_state=69)
```

MODEL KNN

KNeighborsClassifier

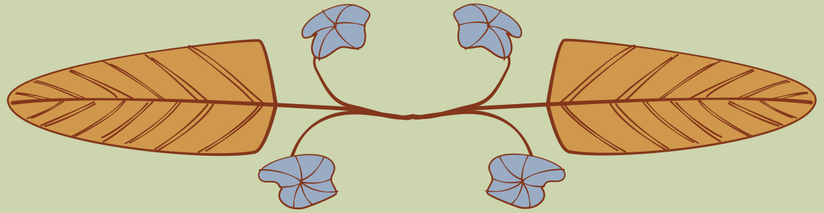
```
KNeighborsClassifier(n_neighbors=69)
```

PREDICT & EVALUATION ✨

MODEL ACCURACY 😊

```python
from sklearn.metrics import accuracy_score

# 1. Memprediksi dan mengevaluasi
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print("Laporan Klasifikasi:")
print(f"🎯 Akurasi Model: {accuracy * 100:.2f}%")
```

```
Laporan Klasifikasi:
🎯 Akurasi Model: 98.67%
```



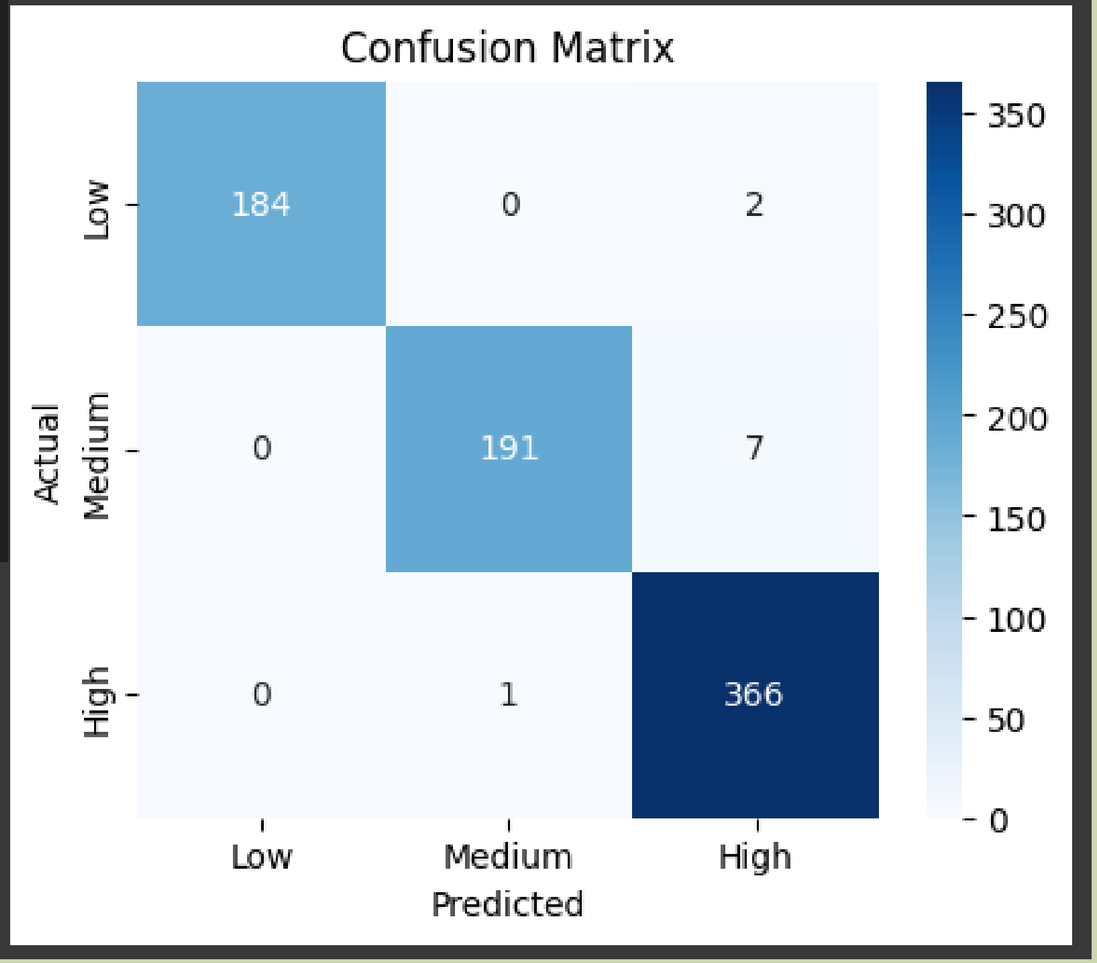
CLASSIFICATION REPORT 😊

```python
# Tampilkan Classification Report
print("\n📊 Classification Report:")
print(classification_report(y_test, y_pred))
```

```
📊 Classification Report:
              precision    recall  f1-score   support

        High       1.00      0.99      0.99       186
         Low       0.99      0.96      0.98       198
      Medium       0.98      1.00      0.99       367

    accuracy                           0.99       751
   macro avg       0.99      0.98      0.99       751
weighted avg       0.99      0.99      0.99       751
```

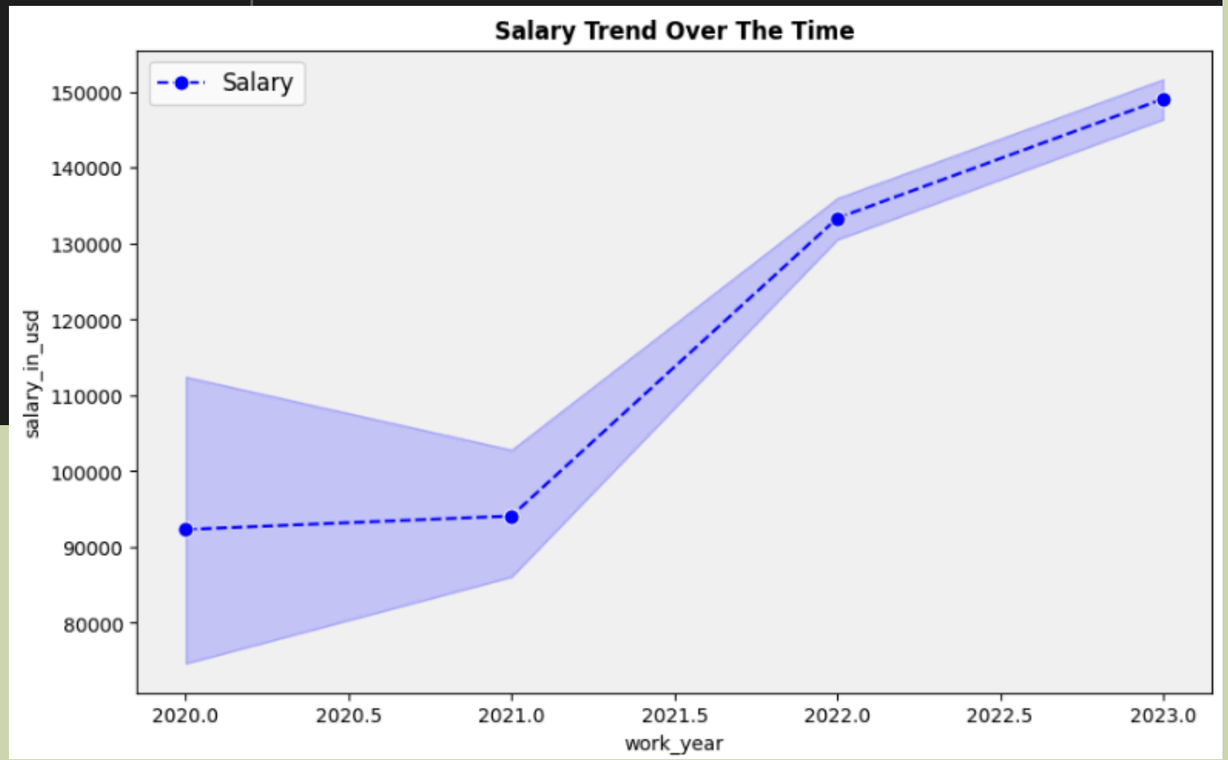
CONFUSION MATRIX ✨

```python
# Tampilkan Confusion Matrix
plt.figure(figsize=(5,4))
sns.heatmap(confusion_matrix(y_test, y_pred),
            annot=True, fmt='d',
            cmap='Blues',
            xticklabels=["Low", "Medium", "High"],
            yticklabels=["Low", "Medium", "High"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```



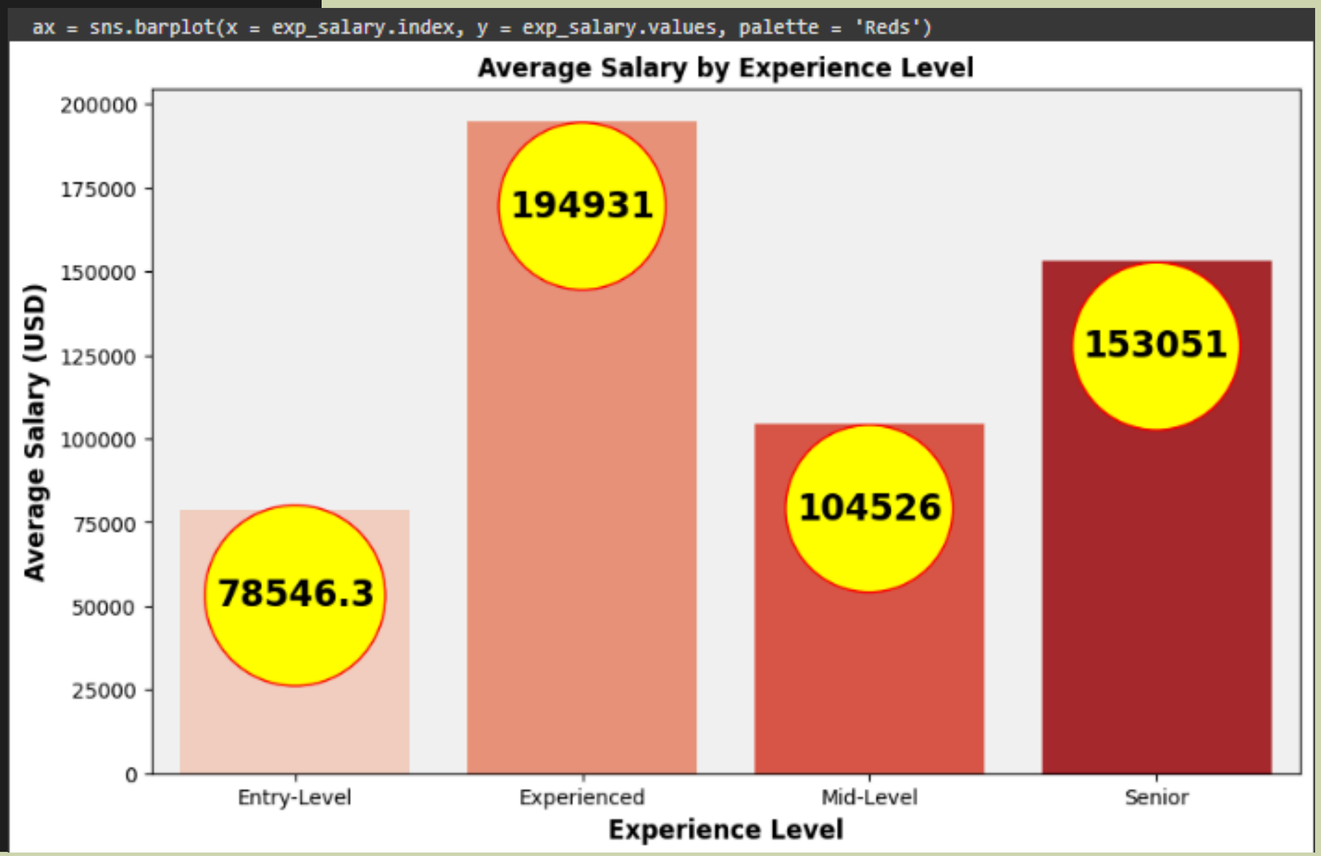
DATA SCIENCE SALARY TRENDS

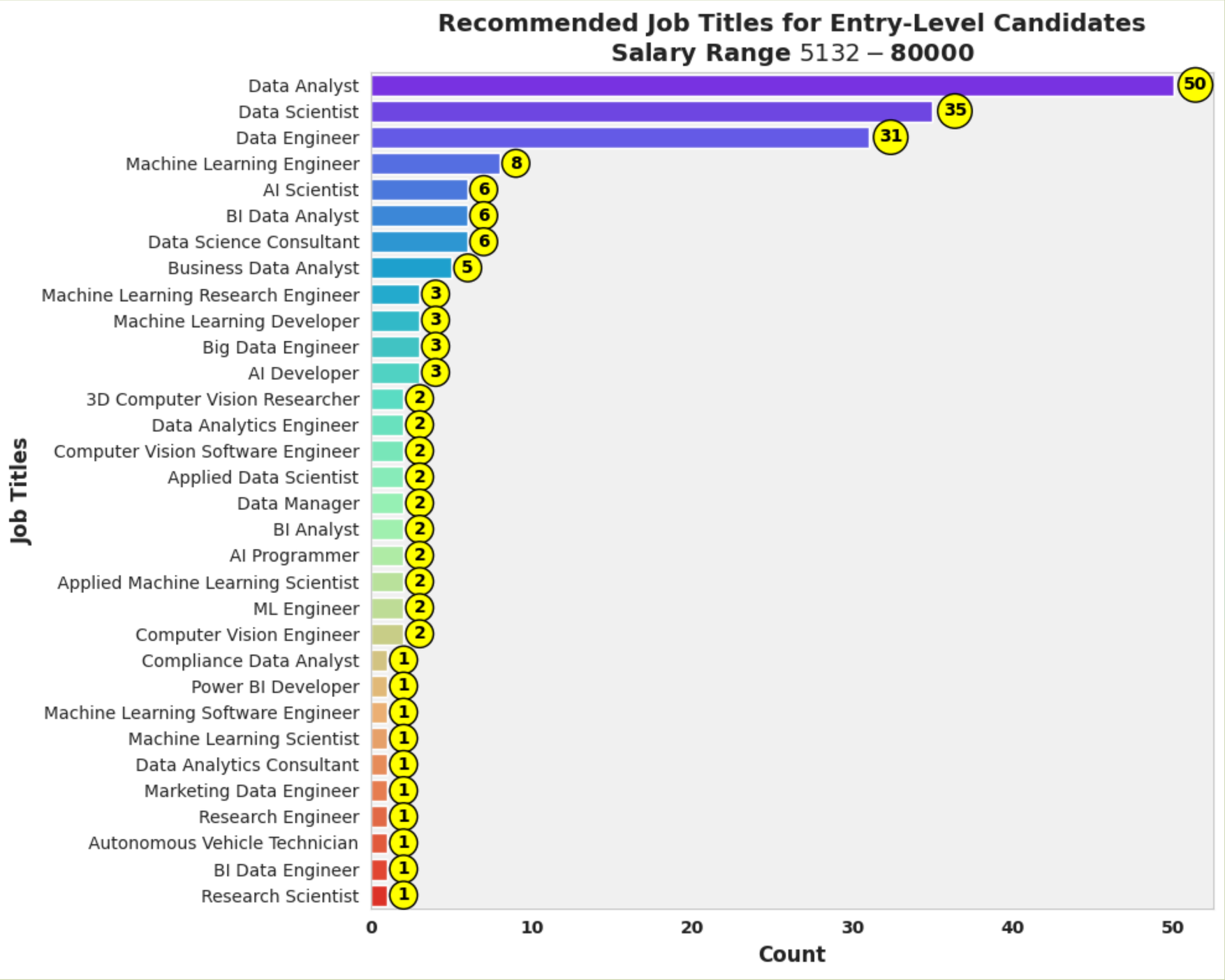
```python
# Add this line at the beginning of your code cell
import matplotlib.pyplot as plt
import seaborn as sns # Import the seaborn library

plt.figure(figsize = (10,6))
salary_trend = df[['salary_in_usd', 'work_year']].sort_values(by = 'work_year')
p = sns.lineplot(data =salary_trend ,x = 'work_year', y = 'salary_in_usd', marker = 'o',linestyle='--', color='Blue', markersize=8 )
plt.title('Salary Trend Over The Time', fontsize=12, fontweight='bold')

# Customize the background color
p.set_facecolor("#f4f4f4")
plt.legend(['Salary'], loc='best', fontsize=12)

# Remove the grid lines
p.grid(False)

plt.show()
```



AVERAGE SALARY BY EXPERIENCE LEVEL

```
exp_salary = df.groupby('experience_level')['salary_in_usd'].mean()
plt.figure(figsize = (10,6))
ax = sns.barplot(x = exp_salary.index, y = exp_salary.values, palette = 'Reds')
plt.title('Average Salary by Experience Level', fontsize=12, fontweight='bold')
plt.xlabel('Experience Level', fontsize=12, fontweight='bold')
plt.ylabel('Average Salary (USD)', fontsize=12, fontweight='bold')

for container in ax.containers:
    ax.bar_label(container,
                 padding = -50,
                 fontsize = 17,
                 bbox = {'boxstyle': 'circle',
                         'edgecolor': 'red',
                         'facecolor': 'yellow'},
                 label_type="edge",
                 fontweight = 'bold'
                 )

# Customize the background color
ax.set_facecolor("#f4f4f4")

# Remove the grid lines
ax.grid(False)

plt.show()
```




Recommended Job Titles for Entry-Level Candidates
Salary Range 5132 – 80000
Data Analyst 50
Data Scientist 35
Data Engineer 31
Machine Learning Engineer 8
AI Scientist 6
BI Data Analyst 6
Data Science Consultant 6
Business Data Analyst 5
Machine Learning Research Engineer 3
Machine Learning Developer 3
Big Data Engineer 3
AI Developer 3
3D Computer Vision Researcher 2
Data Analytics Engineer 2
Computer Vision Software Engineer 2
Applied Data Scientist 2
Data Manager 2
BI Analyst 2
AI Programmer 2
Applied Machine Learning Scientist 2
ML Engineer 2
Computer Vision Engineer 2
Compliance Data Analyst 1
Power BI Developer 1
Machine Learning Software Engineer 1
Machine Learning Scientist 1
Data Analytics Consultant 1
Marketing Data Engineer 1
Research Engineer 1
Autonomous Vehicle Technician 1
BI Data Engineer 1
Research Scientist 1
Job Titles
0
10
20
30
40
50
Count


JOB TITLE
RECOMMENDATION


I can do it!
GREAT
NICE
FANTASTIC
GOOD
Wonderful
PERFECT
GOOD JOB
Excellent!

CONCLUSION!

LINK KODINGAN!

The ‘Data Science Salaries Classification 2023’ project demonstrates that analysing salaries in the field of data science requires a systematic and data-driven approach. Through the exploration of a dataset covering 3,755 rows and 11 columns, we were able to identify significant patterns in salaries based on experience level, job title, location, and company type. Using modelling techniques such as Logistic Regression, Random Forest, SVC, and KNN, we were able to classify salaries into low, medium, and high categories with satisfactory accuracy.

The results of this analysis provide valuable insights for professionals looking to pursue a career in data science, as well as for companies looking to establish a competitive salary structure. The combination of solid technical expertise and a deep understanding of salary dynamics will facilitate better decision-making in the future. This project proves that data is not just a number, but also a source of information that can drive innovation and development in the data science industry.

THANK YOU!

See you later