Glassdoor Project - Interview Preparation

Glassdoor Data Analysis Project - Detailed Interview Preparation

Objective:

Analyze job postings data from Glassdoor to gain insights into salaries, job roles, company ratings, and required skills. Perform data cleaning, feature engineering, visualization, and machine learning modeling to predict salaries.

Step-by-Step Project Flow with Sample Code:

1. Importing Libraries & Data:

Code:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.ensemble import RandomForestRegressor
import pickle
df = pd.read_csv('glassdoor_data.csv')
```

2. Data Cleaning:

- Remove duplicates: `df.drop_duplicates(inplace=True)`
- Handle missing values: `df['column'].fillna(value, inplace=True)`
- Extract salary ranges:

```python
df['min_salary'] = df['Salary Estimate'].apply(lambda x: int(x.split('-')[0].replace('$','').replace('K','')))
df['max_salary'] = df['Salary Estimate'].apply(lambda x: int(x.split('-')[1].replace('$','').replace('K','')))
df['avg_salary'] = (df['min_salary'] + df['max_salary'])/2
```

3. Feature Engineering:

```python
df['job_state'] = df['Location'].apply(lambda x: x.split(',')[1])
df['company_age'] = df['Founded'].apply(lambda x: 2025 - x if x > 0 else x)
df['python_yn'] = df['Job Description'].apply(lambda x: 1 if 'python' in x.lower() else 0)
```

4. Data Visualization:

```python
sns.histplot(df['avg_salary'])
plt.show()
sns.barplot(x='job_state', y='avg_salary', data=df)
plt.xticks(rotation=90)
plt.show()
```

5. Model Building:

```python
X = df[['Rating', 'company_age', 'python_yn', 'Size', 'Industry']]
y = df['avg_salary']
X = pd.get_dummies(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf = RandomForestRegressor()
rf.fit(X_train, y_train)
predictions = rf.predict(X_test)
```

6. Model Evaluation:

```python
from sklearn.metrics import r2_score, mean_absolute_error
print(r2_score(y_test, predictions))
```

print(mean_absolute_error(y_test, predictions))

```

7. Model Saving:

```python

pickle.dump(rf, open('model.pkl','wb'))

```

Why These Methods Were Used:

- Pandas/NumPy: For fast and easy data manipulation.

- Seaborn/Matplotlib: For clear, insightful visualizations.

- Random Forest: Best accuracy, handles non-linearity.

- Lasso Regression: Helps in feature selection.

- Pickle: To save models for future predictions.

Interview Questions & Answers:

Q1: How did you preprocess the salary data?

A: Extracted min and max salary from text, calculated average salary.

Q2: Why Random Forest?

A: High accuracy, handles categorical + numerical features, less overfitting.

Q3: How did you handle missing data?

A: Filled categorical with mode, numerical with median.

Q4: What performance metrics did you use?

A: R^2 score and Mean Absolute Error.

Q5: What improvements can be made?

A: Try XGBoost, tune hyperparameters, use NLP for job descriptions.

Revision Key Points:
```

Glassdoor Project - Interview Preparation

- Salary extraction logic.
- Feature engineering steps.
- Why Random Forest worked best.
- Top features impacting salary.