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')

• • • •

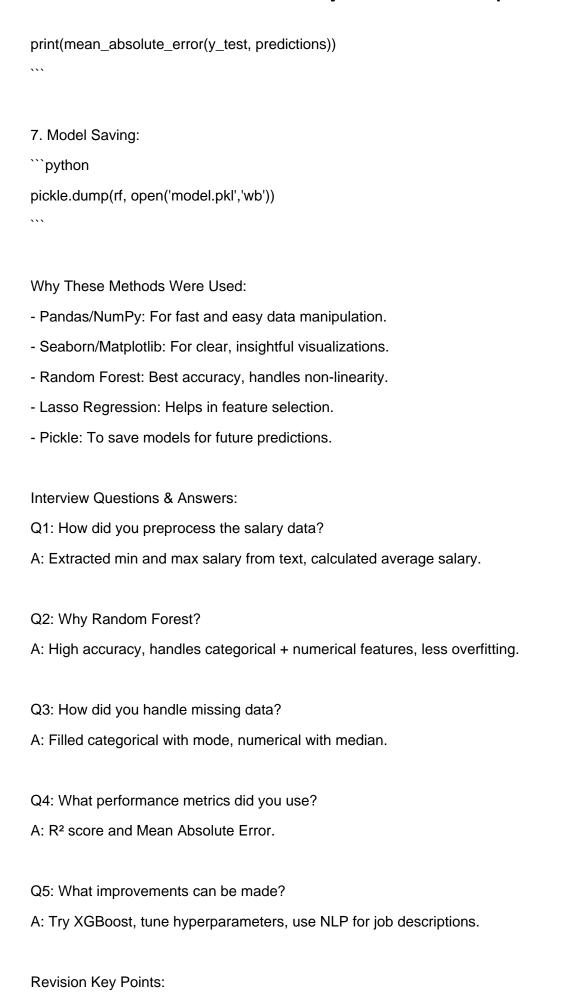
#### 2. Data Cleaning:

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

```
```python
```

```
\begin{split} &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 \end{split}
```

```
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))
```



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