Multicollinearity Assignment

Correlation Matrix with seaborn Heatmap

The code below shows the correlation matrix for the dataset:

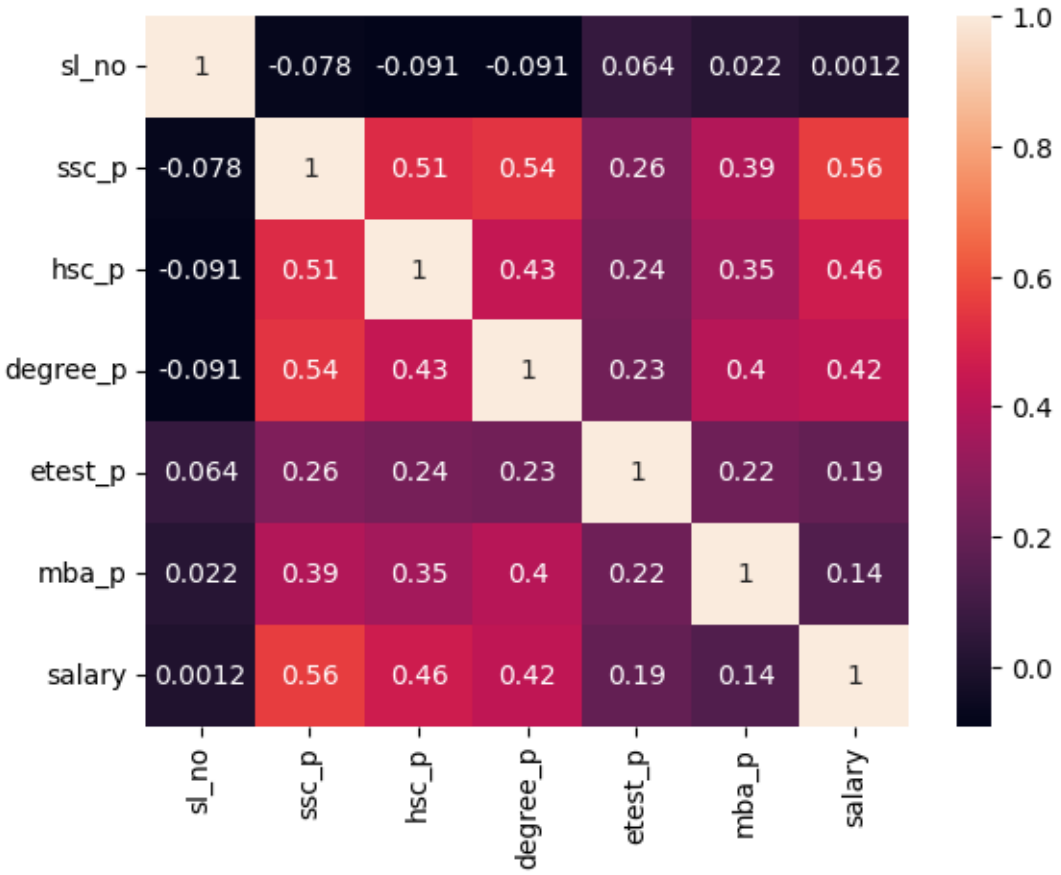
```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix for numeric data
corr_matrix = numeric_data.corr()

# Plot heatmap of the correlation matrix
sns.heatmap(corr_matrix, annot=True)
plt.show()
```

What Happens to the Dataset?

1. **Correlation Matrix Calculation**: numeric_data.corr() computes pairwise correlation coefficients for each feature in numeric_data, showing how strongly each feature is related to the others.
2. **Heatmap Visualization**: The heatmap uses colors and values to display correlation coefficients (ranging from -1 to 1):
    - Values near 1 or -1 indicate high positive or negative correlations, suggesting multicollinearity.
    - Values near 0 suggest little or no correlation between features.
3. **Next Steps**: After observing the heatmap:
    - Identify pairs with high absolute correlation values (|correlation| > 0.8).
    - Consider removing one feature from each highly correlated pair to reduce redundancy.
    - For instance, if ssc_p and degree_p are highly correlated, you might keep only one.

Final Dataset Outcome

Removing highly correlated features will leave you with a dataset that's less prone to multicollinearity, which often improves model performance, especially in linear regression models.



Using Variance Inflation Factor (VIF)

Alternatively, the **VIF** method calculates the Variance Inflation Factor for each feature:

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
import pandas as pd


def calc_vif(X):
    # Calculating VIF
    vif = pd.DataFrame()
    vif["variables"] = X.columns
    vif["VIF"] = [variance_inflation_factor(X.values, i) for i in range(X.shape[1])]
    return vif
```

What Happens to the Dataset?

1. **VIF Calculation**: This code computes the VIF for each feature in X, where a high VIF (generally >10) indicates a strong correlation with other features.

2. **Decision Making**:
    - Identify features with high VIF values and consider removing them from X.
    - Removing these features helps reduce multicollinearity.
3. **Final Dataset Outcome**: Like the correlation matrix method, removing features with high VIF will reduce multicollinearity. This method can be more precise than the correlation matrix since it accounts for how a feature correlates with all other features collectively.

Summary of Differences

- **Correlation Matrix**: Helps visually identify pairwise correlations. Best for initial exploration.
- **VIF**: Quantifies the effect of multicollinearity in regression models. Useful for more detailed analysis.

In the end, both methods yield a cleaner dataset with fewer redundant features, improving model performance, interpretability, and stability.