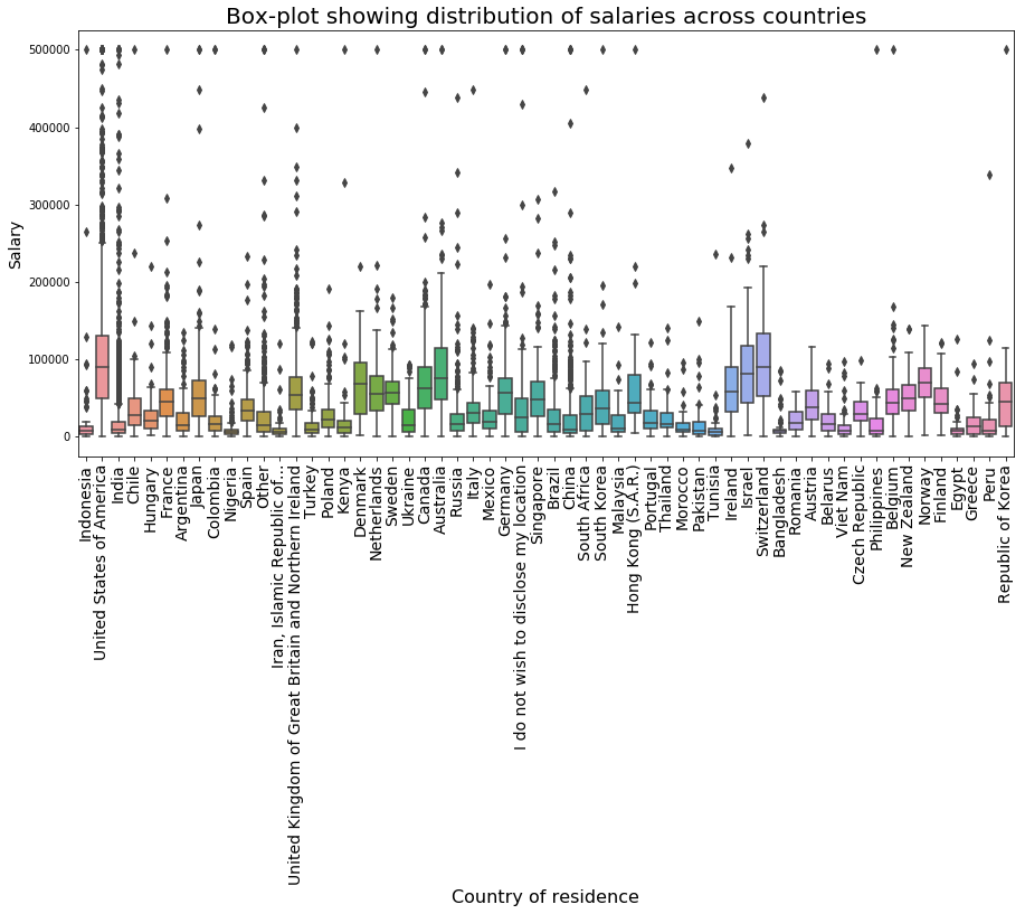
Assignment - 2

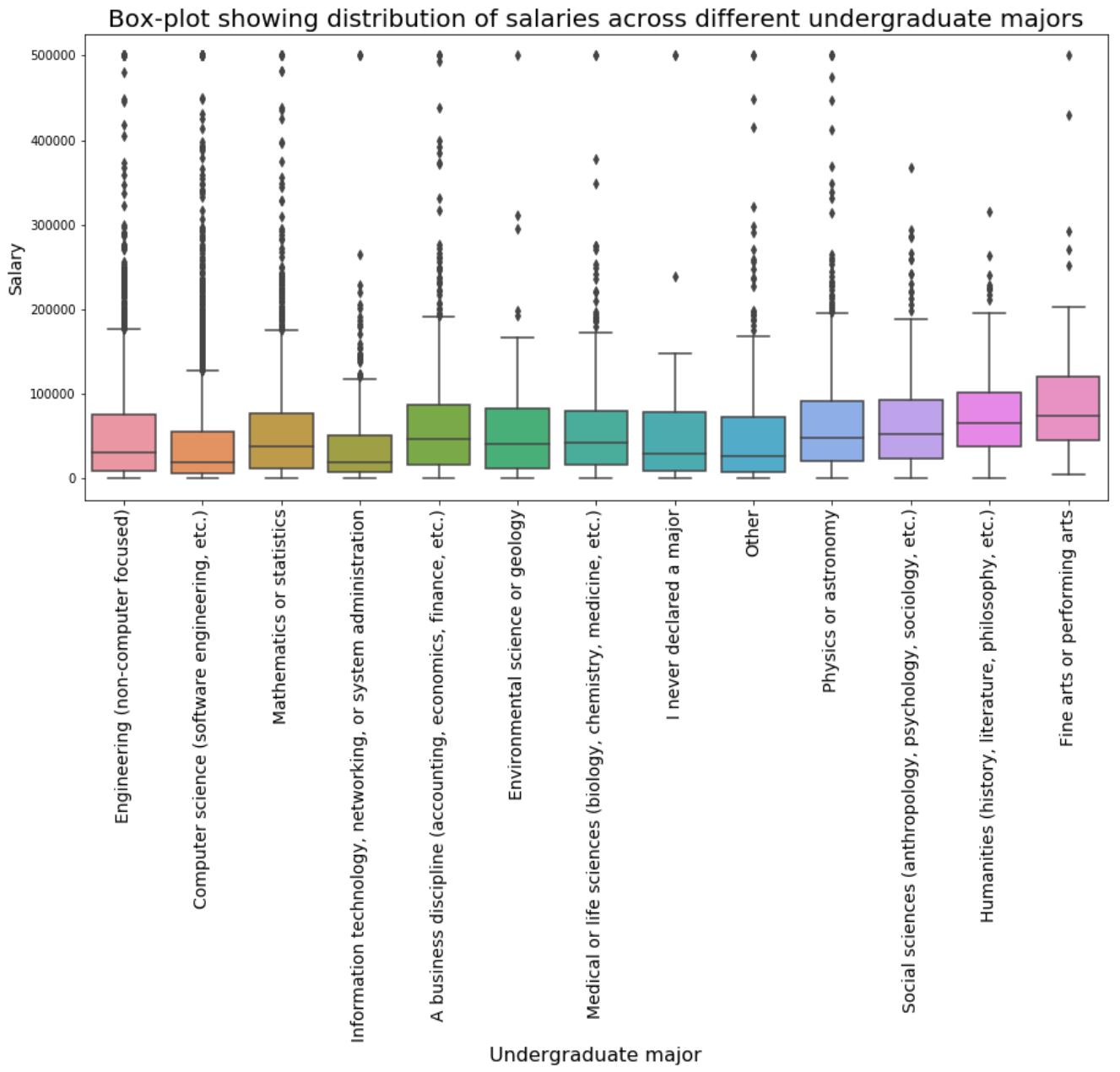
Abhineet Sain

1004529165

Exploratory Data Analysis

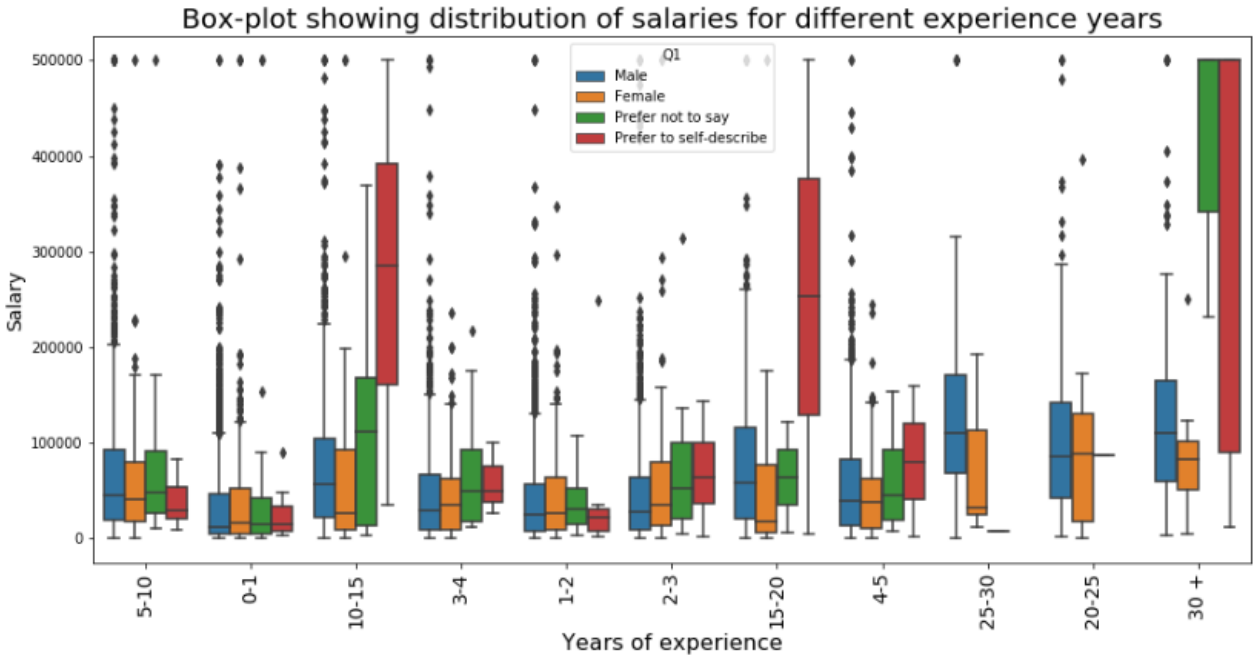


- Highest median salaries are for United States and Switzerland.
- Hints on the importance of the 'place of residence' feature for salary prediction.

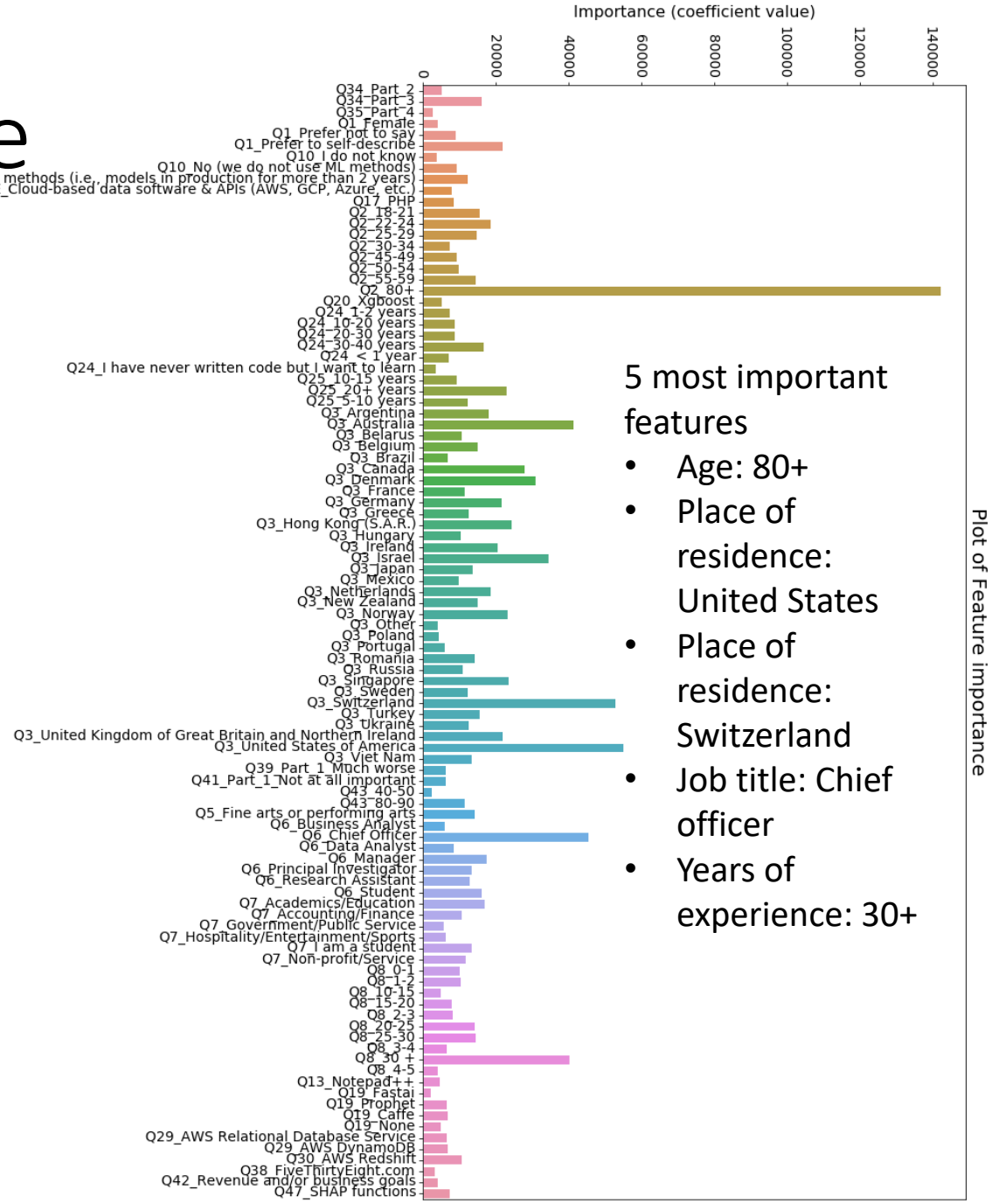


- Overall, location of box-plots is more or less uniform indicating a lower importance of undergrad than the country of residence.

EDA + Feature importance



- Median salary for experience upto 5-10 years, more or less at similar levels, go up beyond 10 years.
- As experience increases, outlier salary clusters above median shift upwards.
- Gender disparity in higher experience groups.



5 most important features

- Age: 80+
- Place of residence: United States
- Place of residence: Switzerland
- Job title: Chief officer
- Years of experience: 30+

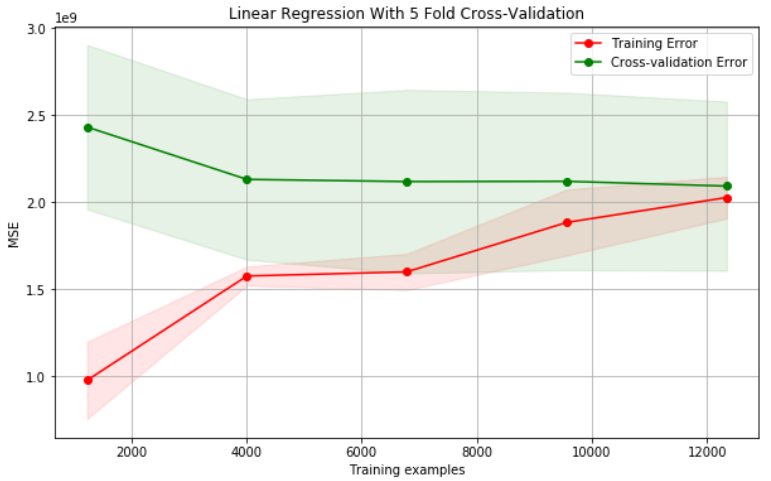
Results (10 fold CV)

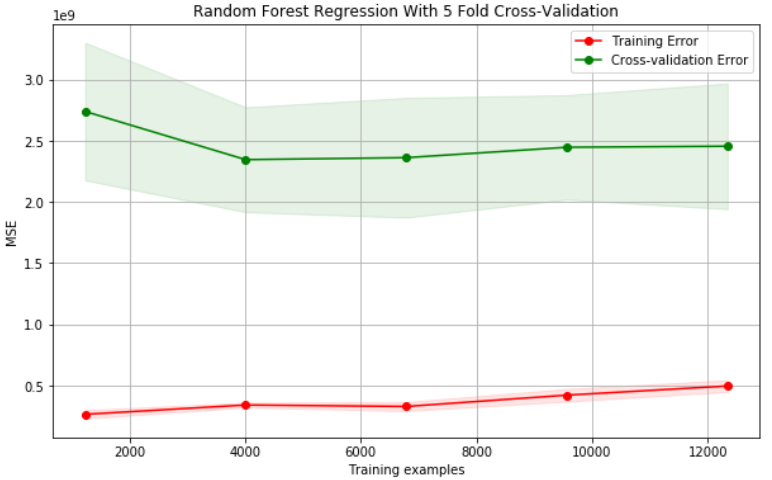
```
Fitting Linear Regression model
    Fold 1: Test Accuracy: 41.086 Train Accuracy: 47.296%
    Fold 2: Test Accuracy: 53.699 Train Accuracy: 46.345%
    Fold 3: Test Accuracy: 58.469 Train Accuracy: 46.143%
    Fold 4: Test Accuracy: 46.415 Train Accuracy: 47.546%
    Fold 5: Test Accuracy: 44.433 Train Accuracy: 47.854%
    Fold 6: Test Accuracy: 43.834 Train Accuracy: 47.839%
    Fold 7: Test Accuracy: 46.618 Train Accuracy: 47.59%
    Fold 8: Test Accuracy: 41.174 Train Accuracy: 48.19%
    Fold 9: Test Accuracy: 41.016 Train Accuracy: 48.146%
    Fold 10: Test Accuracy: 19.72 Train Accuracy: 49.448%
  Average test Score: 43.647%(9.644%)
  Average training Score: 47.64%(0.892%)
```

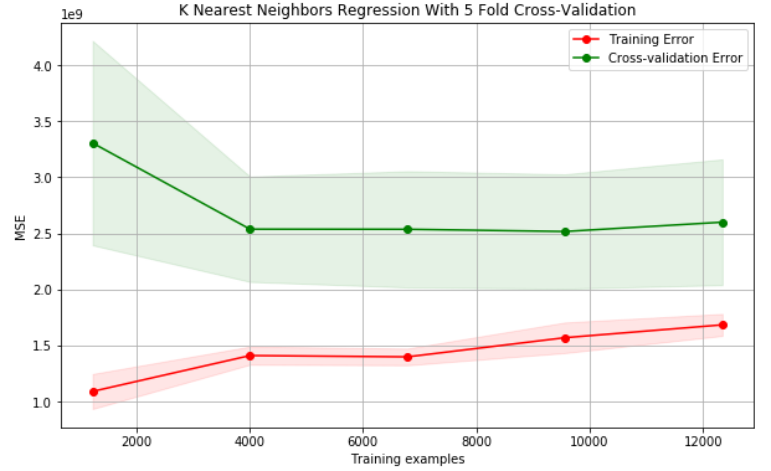
```
Fitting Random Forest Regression model
/home/jupyterlab/conda/lib/python3.6/site-packages/sklearn
mators will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
    Fold 1: Test Accuracy: 27.549 Train Accuracy: 87.135%
    Fold 2: Test Accuracy: 42.933 Train Accuracy: 86.892%
    Fold 3: Test Accuracy: 46.899 Train Accuracy: 86.338%
    Fold 4: Test Accuracy: 39.912 Train Accuracy: 87.089%
    Fold 5: Test Accuracy: 34.458 Train Accuracy: 87.611%
    Fold 6: Test Accuracy: 35.656 Train Accuracy: 87.369%
    Fold 7: Test Accuracy: 41.201 Train Accuracy: 87.064%
    Fold 8: Test Accuracy: 32.111 Train Accuracy: 88.448%
    Fold 9: Test Accuracy: 26.566 Train Accuracy: 88.156%
    Fold 10: Test Accuracy: 7.405 Train Accuracy: 87.952%
  Average test Score: 33.469%(10.681%)
  Average training Score: 87.405%(0.607%)
```

```
Fitting K Nearest Neighbors Regression model
    Fold 1: Test Accuracy: 20.5 Train Accuracy: 56.999%
    Fold 2: Test Accuracy: 38.404 Train Accuracy: 55.34%
    Fold 3: Test Accuracy: 43.951 Train Accuracy: 55.4%
    Fold 4: Test Accuracy: 38.249 Train Accuracy: 56.466%
    Fold 5: Test Accuracy: 30.353 Train Accuracy: 57.524%
    Fold 6: Test Accuracy: 34.012 Train Accuracy: 56.277%
    Fold 7: Test Accuracy: 39.036 Train Accuracy: 56.638%
    Fold 8: Test Accuracy: 25.863 Train Accuracy: 56.853%
    Fold 9: Test Accuracy: 30.55 Train Accuracy: 56.988%
    Fold 10: Test Accuracy: 4.368 Train Accuracy: 58.741%
  Average test Score: 30.529%(10.923%)
  Average training Score: 56.723%(0.939%)
```

```
Fitting Gradient Boosting Regressor model
    Fold 1: Test Accuracy: 41.302 Train Accuracy: 52.598%
    Fold 2: Test Accuracy: 53.544 Train Accuracy: 51.451%
    Fold 3: Test Accuracy: 58.483 Train Accuracy: 52.102%
    Fold 4: Test Accuracy: 45.645 Train Accuracy: 53.199%
    Fold 5: Test Accuracy: 42.528 Train Accuracy: 53.533%
    Fold 6: Test Accuracy: 40.795 Train Accuracy: 53.795%
    Fold 7: Test Accuracy: 46.917 Train Accuracy: 52.98%
    Fold 8: Test Accuracy: 42.212 Train Accuracy: 53.135%
    Fold 9: Test Accuracy: 42.895 Train Accuracy: 53.568%
    Fold 10: Test Accuracy: 22.678 Train Accuracy: 54.8%
  Average test Score: 43.7%(8.888%)
  Average training Score: 53.116%(0.881%)
```







Overall, Gradient Boosting regression gives the best accuracy on the test dataset with the lowest std dev.

Results (Hyperparameter tuning)

```
Random Forest Regression
Fitting 10 folds for each of 30 candidates, totalling 300 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 300 out of 300 | elapsed:  3.4min finished
Best parameters: {'n_estimators': 80, 'min_samples_split': 15, 'min_samples_leaf': 4, 'max_features': 'auto', 'max_depth': No
ne, 'bootstrap': False}
Best cross-validation score: 34.52%
K Nearest Neighbors Regression
Fitting 10 folds for each of 10 candidates, totalling 100 fits
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed: 28.0min finished
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
Best parameters: {'weights': 'uniform', 'n_neighbors': 14, 'algorithm': 'kd_tree'}
Best cross-validation score: 40.92%
Gradient Boosting Regressor
Fitting 10 folds for each of 30 candidates, totalling 300 fits
```

- For Random Forests, the accuracy increased from ~31.7% to 48.12%
- For K nearest neighbors, the accuracy increased from ~30% to 40.93%
- For Gradient boosting regression, the accuracy increased from 43.7% to 50.52%