Team 1:
Neha Annamalai
Anna Hubbard
Aaria Sethi

University Recruitment and Placement Status

Background

1. Does the final report describe the setting and reference to related research?

A student's prior education and job history play a crucial role in university recruitment and campus acceptance. Myriad students, including ourselves, have undergone the ever-so difficult college application process. Be it standardized test scores, grades, or extracurricular achievements - every aspect of student involvement is utilized for university admissions. Origins and further background information regarding this data set remain unaddressed. The dataset description mentions how Dr. Dhimant Ganatara, a professor at Jain University in India, is responsible for the collection and creation of this dataset.

2. Does the final report describe the data that you are working with and how it was derived?

Our dataset will be used to help determine the placement status of a student in the workforce and focuses on fourteen features — serial number, gender, secondary education percentage (10th grade), secondary board of education, higher secondary education percentage (12th grade), higher secondary board of education, specialization in higher secondary education, degree percentage, undergrad degree type, work experience, employability test percentage, post-grad (MBA) specialization, MBA percentage, and salary offered by corporates (to the placed students). The target variable, in this case, is the placement status. This categorical variable has two classes — placed or not placed.

The dataset was derived from students in an "XYZ campus" and is focused on the placement of these students in the workforce (Roshan).

Design

1. Does the final report have a concrete well defined experimental design describing the learning task?

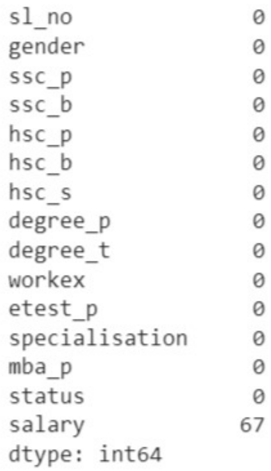
Our learning task was to use our given dataset to help predict and further analyze the placement of students into a secondary education program.

2. Does the final report describe which features and models are used and why they were chosen?

As a result, we used the Naive Bayes classifier, Decision Tree, and Random Forest algorithm to help implement our experimental design. The Naive Bayes classifier helps compute a probability for a class in regards to the probability distribution entailed in the training data. The Naive Bayes assumption states that all features are independent. It is commonly used for classification tasks and detailed computation that is heavily dependent on probability. A Decision

Tree poses questions and categorizes each instance in regard to the answer. It can be thought of as a compilation of if-then statements. The end result of a decision tree is a classification for a given input of x. In our example, each student attribute may be used as a means for classification. The final leaves will determine placement or not. Since decision trees are extremely visualizable, this was a beneficial algorithm to help learn about feature importance. However, often Decision Trees are overfitted and have high variance. To combat this issue, the Random Forest Algorithm utilizes bootstrap aggregation and creates multiple models to further combine into one result. This eliminates large impacts based on varying discrepancies. This is a classification task as well.

Since not all students have prior work experience, they do not all have a salary offered by corporations. Thus, the dataset contains some missing data. To ensure our assumption was correct, we checked the data for columns containing null values and received the following results:

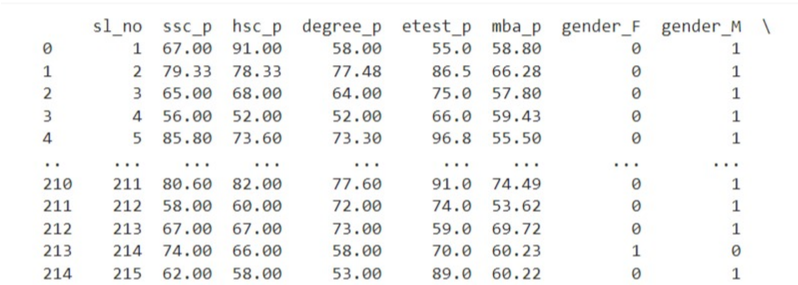


```
sl_no              0
gender             0
ssc_p              0
ssc_b              0
hsc_p              0
hsc_b              0
hsc_s              0
degree_p           0
degree_t           0
workex             0
etest_p            0
specialisation     0
mba_p              0
status             0
salary            67
dtype: int64
```

Figure 1

This chart (*Figure 1*) shows that the salary column contains 67 null observation values. To accommodate this, we incorporated the most suitable feature imputation method into our coding project. In this case, we removed the salary feature because it contains missing data which implies no previous job experience. We decided to remove the salary feature because it did not play an important role in the main classification task of whether or not a student was placed in the workforce. If we were to perform a more specific experiment on those places and look at the difference in salaries based on other variables then we would have left the salary feature included and simply removed the rows containing the missing data.

Since we have categorical feature columns, we performed one-hot encoding to transform the categorical features into numerical features. This process allows us to next perform filter feature selection on the dataset. Below, in *Figure 2*, is a snippet of the one-hot encoded data.

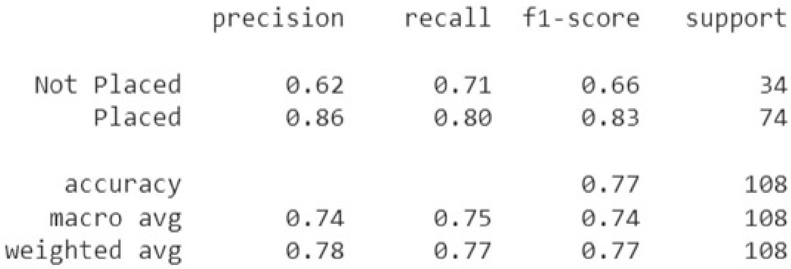


```
     sl_no  ssc_p  hsc_p  degree_p  etest_p  mba_p  gender_F  gender_M  \
0        1  67.00  91.00     58.00     55.0  58.80         0         1
1        2  79.33  78.33     77.48     86.5  66.28         0         1
2        3  65.00  68.00     64.00     75.0  57.80         0         1
3        4  56.00  52.00     52.00     66.0  59.43         0         1
4        5  85.80  73.60     73.30     96.8  55.50         0         1
..     ...    ...    ...       ...      ...    ...       ...       ...
210    211  80.60  82.00     77.60     91.0  74.49         0         1
211    212  58.00  60.00     72.00     74.0  53.62         0         1
212    213  67.00  67.00     73.00     59.0  69.72         0         1
213    214  74.00  66.00     58.00     70.0  60.23         1         0
214    215  62.00  58.00     53.00     89.0  60.22         0         1
```

Figure 2

After altering the data to consist of numerical inputs and a categorical output, we used filter feature selection and chi-squared methods to determine which features play a role in the status placement classification. The results show that we were able to eliminate two columns (one feature) that do not play an important role in the classification.

Next, we split our data into test and train and used Naive Bayes to predict the probability of a student being placed or not placed on the observed feature values. After running Naive Bayes on the test and train data, the results showed that out of a total of 108 points, 25 of them were mislabeled. To follow this up, we built a performance matrix on y_test and y_pred as shown below. The report in *Figure 3* indicates an accuracy of 77%. This score is ideal and represents a high ratio of correctly predicted observations to total observations.



```
              precision    recall  f1-score   support

  Not Placed       0.62      0.71      0.66        34
      Placed       0.86      0.80      0.83        74

    accuracy                           0.77       108
   macro avg       0.74      0.75      0.74       108
weighted avg       0.78      0.77      0.77       108
```

Figure 3

However, given that the Naive Bayes classifier assumes all features are independent, we decided that we may need an additional method of classification. Hence, next, we created a decision tree. This allows us to have a visual model of the actual decision process that determines the placement of students. Random forests often provide us with a more generalized output. We also implemented decision trees for visualization purposes.

Implementation

1. Did you implement the required appropriate techniques?

The libraries that we used to implement these functions were sklearn, pandas, numpy, and matplotlib.

2. Does the final report describe what was implemented and what tools were used?

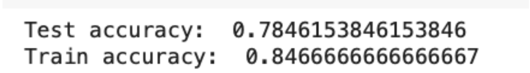
Sklearn has Naive Bayes, RandomForestClassifier, DecisionTreeClassier functions. More specifically, we were able to further use selectkbest, chi2, LabelEncoder, train_test_split and GaussianNB to help us access the data. Visualizations were also printed as a result of the process.

Results

1. Does the final report have quantitative results from learning or experimenting with your data?

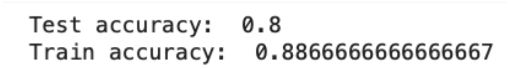
Quantitative Results:

Max depth 2 - decision tree



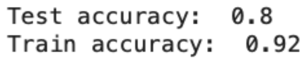
```
Test accuracy:  0.7846153846153846
Train accuracy:  0.8466666666666667
```

Max depth 3 - decision tree



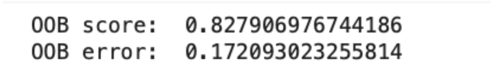
```
Test accuracy:  0.8
Train accuracy:  0.8866666666666667
```

Max depth 4 - decision tree



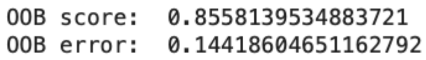
```
Test accuracy:  0.8
Train accuracy:  0.92
```

OOB n_estimators = 15



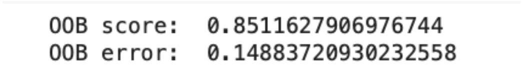
```
OOB score:  0.827906976744186
OOB error:  0.172093023255814
```

OOB n_estimators = 17



```
OOB score:  0.8558139534883721
OOB error:  0.14418604651162792
```

OOB n_estimators = 20

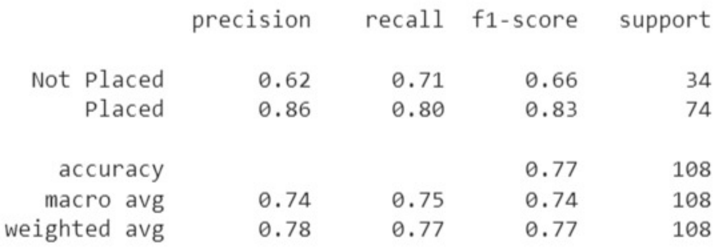


```
OOB score:  0.8511627906976744
OOB error:  0.14883720930232558
```

3. Does the final report have results evaluating the learning of your model?

Evaluation Results:

Gaussian NB Evaluation: metrics with precision, recall, f1 score and support for the placement class (place vs not placed)



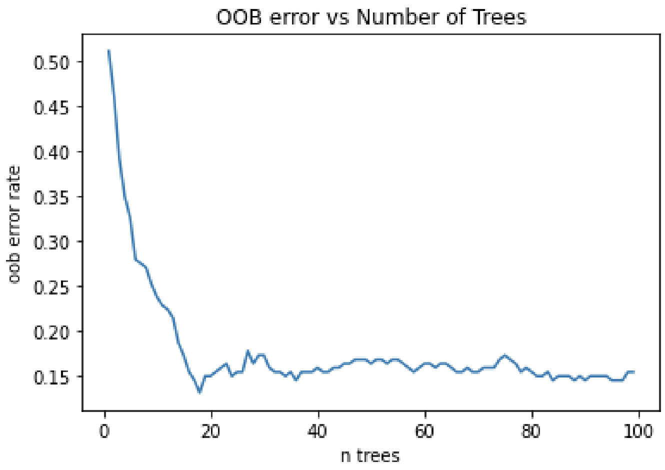
```
              precision    recall  f1-score   support

  Not Placed       0.62      0.71      0.66        34
      Placed       0.86      0.80      0.83        74

    accuracy                           0.77       108
   macro avg       0.74      0.75      0.74       108
weighted avg       0.78      0.77      0.77       108
```

4. Does the final report make effective use of graphs that are appropriately labeled and properly described in the document?

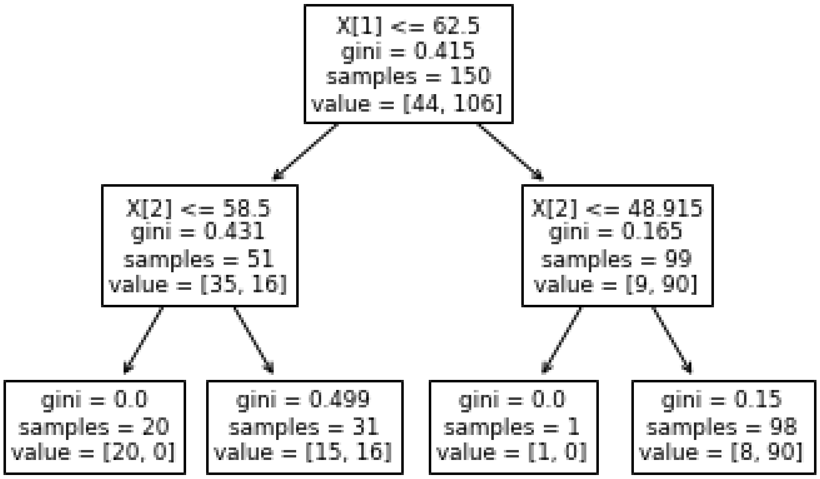
Number of trees vs OOB error graph for Random Forest



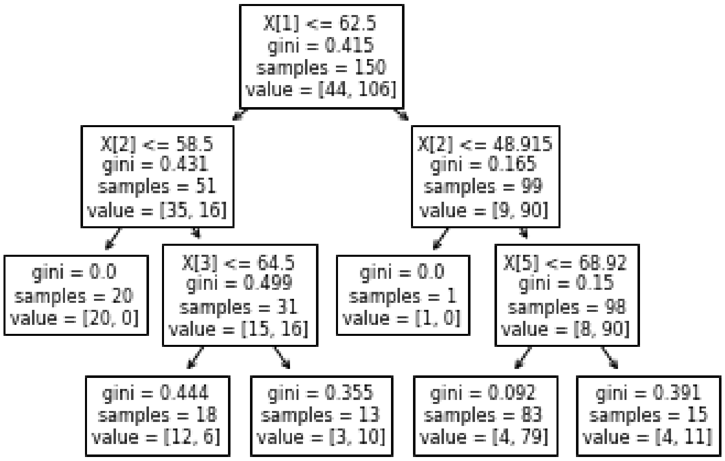
2. Does the final report have data analysis using visualization tools? Does the report have findings for the same?

Qualitative/ Visual Results:

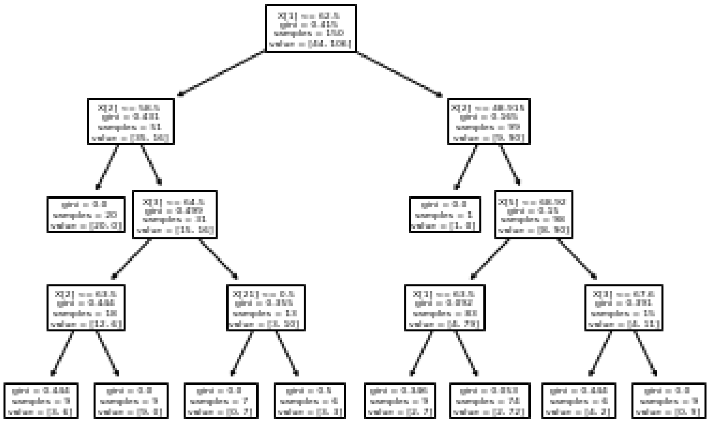
Decision Tree for max depth 2



Decision Tree for max depth 3



Decision Tree for max depth 4



Interpretation

1. Does the final report attempt to interpret the results? Does the interpretation correctly use concepts to justify the results?

According to our final results, we were able to gain insight on numerous aspects of our data. For our decision tree classifier, we tried to find the point where the training and testing accuracy were similar - before divergence. This happened to be the case when the max depth was 2 and 3. Once we changed max depth to four, the accuracy divergence was significant. As we can see, many of the leaf nodes have more than 1 sample in it for decision trees of max depth 2 and 3. This is because shortening the max depth reduces the chance of overfitting. For our random forest classifier, we needed to find the optimal number of trees. To do this, we plotted the OOB error versus the number of trees. From this graph, it was rather simple to analyze that the lowest error was present at 17 trees.

References

"FACTORS IN ADMISSION DECISIONS." *National Association for College Admission Counseling*, https://www.nacacnet.org/globalassets/documents/publications/research/soca_chapter3.pdf. Accessed 4 June 2022.

Higher ED Admissions. "Students Want Meaningful Relationships: 4 Ways to Make This Part of Recruiting — 5° Branding." *5 Degrees Branding*, 19 February 2020, https://www.5degreesbranding.com/blog-full/2020/2/19/students-want-meaningful-relationships-4-ways-to-make-this-part-of-recruiting. Accessed 4 June 2022.

Roshan, Ben. "Campus Recruitment." *Kaggle*, https://www.kaggle.com/datasets/benroshan/factors-affecting-campus-placement. Accessed 4 June 2022.

Times Higher Education. "How universities can develop and sustain relationships throughout the recruitment and admissions journey." *Times Higher Education*, https://www.timeshighereducation.com/hub/salesforceorg/p/how-universities-can-develop-and-sustain-relationships-throughout-recruitment. Accessed 4 June 2022.