American University research

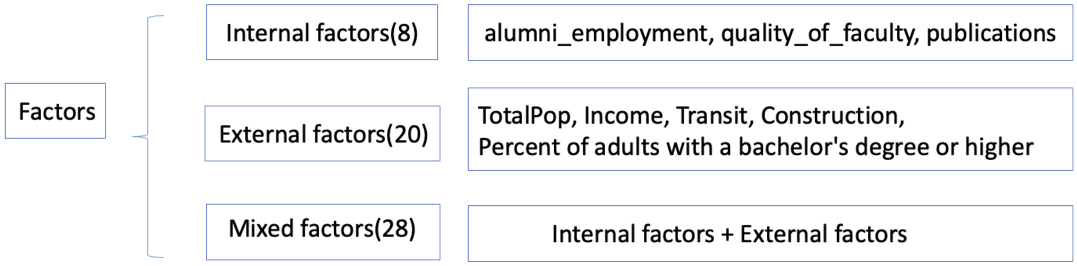
Project ID: B8
Teamer: Chaolong Shi, Kebei Yu, Yulun Wu

Background:
There are hundreds of different national and international university ranking systems like US News and QS. Ranking of universities are affected by a large number of factors. And it is a difficult and controversial practice ranking universities. For both universities and future students, the world rank of top 200 is an important benchmark.

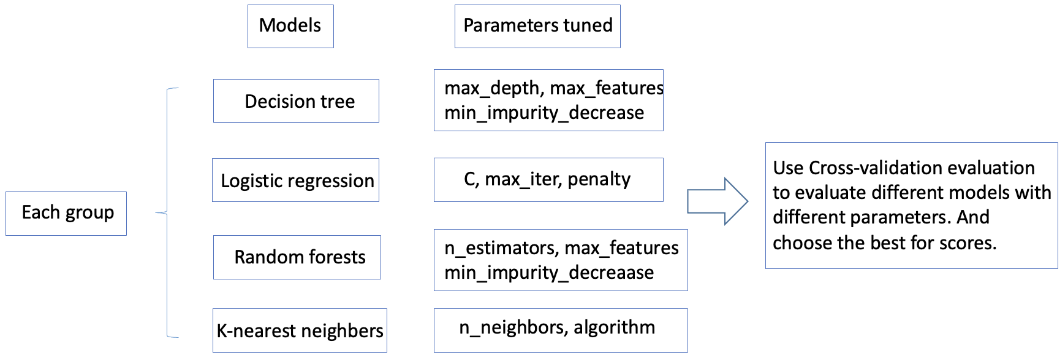
Objective:
The problems we hope to solve are which factors and models are better predictors to predict whether a university is top 200 or not. We break down the factors into three groups: internal factors, external factors and mixed factors to find the best group of factors and models.

Methodology:

1. Use data integration techniques to combine data from different sources.
2. Classify features into three groups. (graph1)
3. Use machine learning such as decision tree, linear regression to perform analysis on the data and output results for discussions. (graph2)
4. Use cross-validation methods to test and evaluate the models, and choose the best models. (graph2)


Factors
Internal factors(8)
alumni_employment, quality_of_faculty, publications
External factors(20)
TotalPop, Income, Transit, Construction,
Percent of adults with a bachelor's degree or higher
Mixed factors(28)
Internal factors + External factors

(graph1)



(graph2)

Part I: Data Preparation

University Ranking Data

University Ranking data were published by The Center for World University Rankings (CWUR) in 2019.

Sources: The Center for World University Rankings (CWUR)

https://www.kaggle.com/mylesoneill/world-university-rankings/download

University Information Data

University Location Data is a dataset that contains locations and university and College.

Sources: Private

Education data

This data contains the education level data from the American Community Survey for adults 25+. Counts are broken down by sex. And there are 5-year estimates shown by tract, county, and state boundaries.

Sources: U.S. Census Bureau's American Community Survey (ACS) 2016-2020 5-year estimates, Table(s) B15002
https://services.arcgis.com/P3ePLMYs2RVChkJx/arcgis/rest/services/ACS_Educational_Attainment_Boundaries/FeatureServer

Population data

This data provides boundaries for the States of the United States in the 50 states and the District of Columbia. And the attribute fields include estimated 2017 total population. Sources: Esri, TomTom, U.S. Department of Commerce, U.S. Census Bureau
https://services.arcgis.com/P3ePLMYs2RVChkJx/arcgis/rest/services/USA_States_Generalized/FeatureServer

Data cleaning:

- Missing value check: none
- Lower case for future matching
- String revising : delete 'county', replace state full name to abbreviation
- Filter year: use the latest info, only us universities,

Data integration:

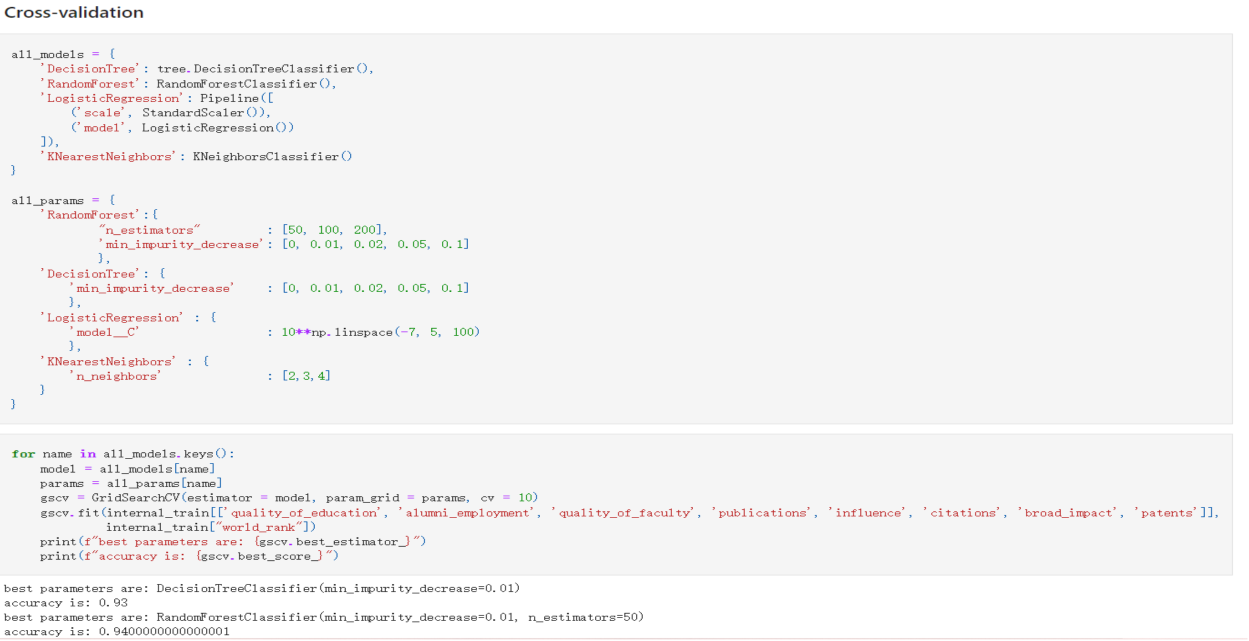
- Merge university info to university rank
- Merge education data to demographic data
- Merge local demographic data to university rank(county level)

Part II: Model Analysis

We use four machine learning models for each group of factors we extracted-internal, external and mixed. Next, we employ cross-validation to select which model is best for each group of factors. Finally, we gather the accuracy for each group of factors to decide which group is the best for us to predict.

Machine learning for each group of factors:

- Split function (80% for training & 20% for testing)
- Decision Tree
- Logistic regression
- Random forest
- Knn
- Cross-validation



Cross-validation

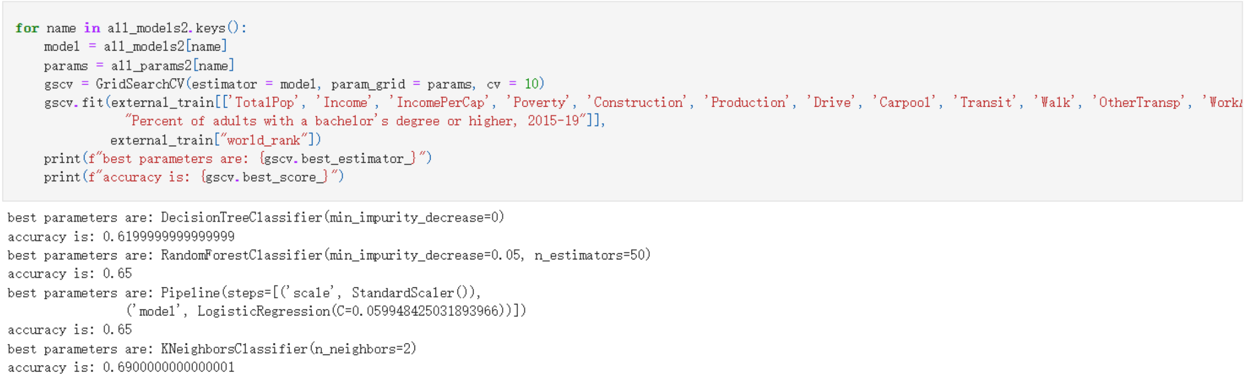
```
all_models = {
    'DecisionTree': tree.DecisionTreeClassifier(),
    'RandomForest': RandomForestClassifier(),
    'LogisticRegression': Pipeline([
        ('scale', StandardScaler()),
        ('model', LogisticRegression())
    ]),
    'KNearestNeighbors': KNeighborsClassifier()
}

all_params = {
    'RandomForest':{
            "n_estimators"          : [50, 100, 200],
            'min_impurity_decrease': [0, 0.01, 0.02, 0.05, 0.1]
            },
    'DecisionTree': {
        'min_impurity_decrease'     : [0, 0.01, 0.02, 0.05, 0.1]
        },
    'LogisticRegression' : {
        'model__C'                  : 10**np.linspace(-7, 5, 100)
        },
    'KNearestNeighbors' : {
        'n_neighbors'               : [2,3,4]
    }
}
```

```
for name in all_models.keys():
    model = all_models[name]
    params = all_params[name]
    gscv = GridSearchCV(estimator = model, param_grid = params, cv = 10)
    gscv.fit(internal_train[['quality_of_education', 'alumni_employment', 'quality_of_faculty', 'publications', 'influence', 'citations', 'broad_impact', 'patents']],
             internal_train["world_rank"])
    print(f"best parameters are: {gscv.best_estimator_}")
    print(f"accuracy is: {gscv.best_score_}")
```

```
best parameters are: DecisionTreeClassifier(min_impurity_decrease=0.01)
accuracy is: 0.93
best parameters are: RandomForestClassifier(min_impurity_decrease=0.01, n_estimators=50)
accuracy is: 0.9400000000000001
```

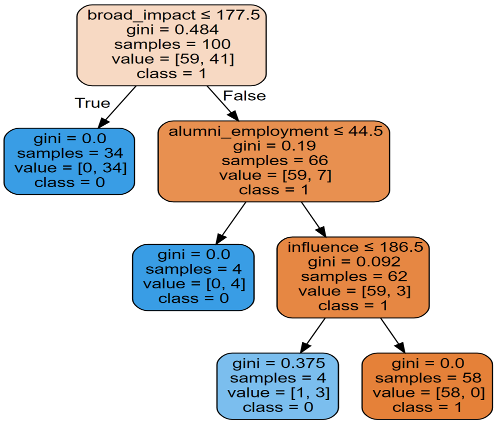
Cross-validation analysis for external factors:

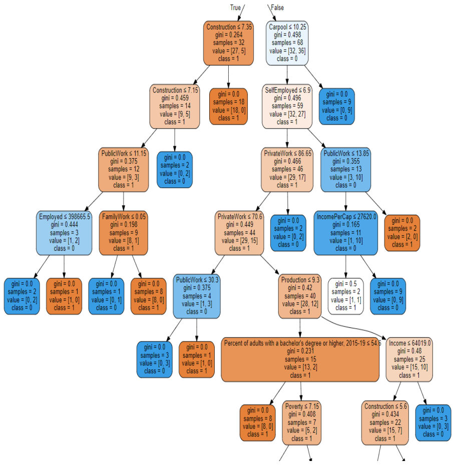


```
for name in all_models2.keys():
    model = all_models2[name]
    params = all_params2[name]
    gscv = GridSearchCV(estimator = model, param_grid = params, cv = 10)
    gscv.fit(external_train[['TotalPop', 'Income', 'IncomePerCap', 'Poverty', 'Construction', 'Production', 'Drive', 'Carpool', 'Transit', 'Walk', 'OtherTransp', 'Work
               "Percent of adults with a bachelor's degree or higher, 2015-19"]],
             external_train["world_rank"])
    print(f"best parameters are: {gscv.best_estimator_}")
    print(f"accuracy is: {gscv.best_score_}")
```

```
best parameters are: DecisionTreeClassifier(min_impurity_decrease=0)
accuracy is: 0.6199999999999999
best parameters are: RandomForestClassifier(min_impurity_decrease=0.05, n_estimators=50)
accuracy is: 0.65
best parameters are: Pipeline(steps=[('scale', StandardScaler()),
                ('model', LogisticRegression(C=0.059948425031893966))])
accuracy is: 0.65
best parameters are: KNeighborsClassifier(n_neighbors=2)
accuracy is: 0.6900000000000001
```

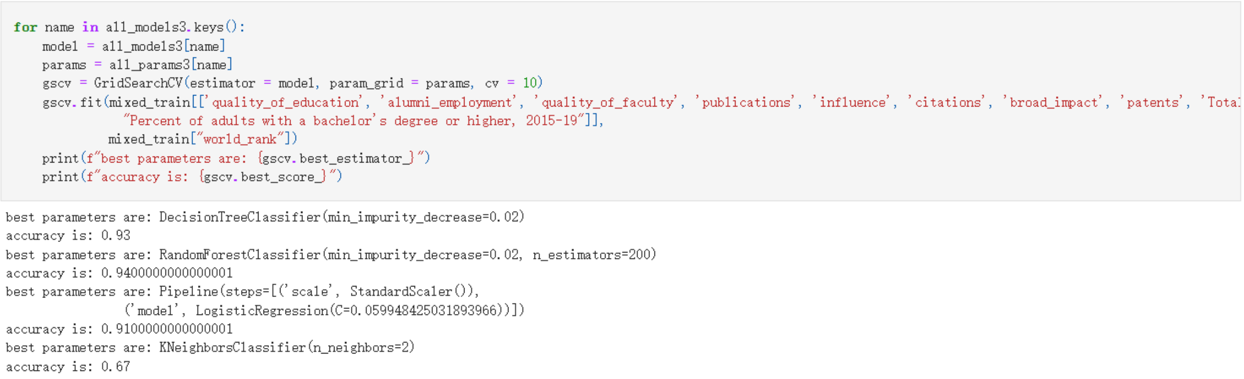
Decision tree for internal and external:





Two decision trees are shown above, due to the fact that external factors have much more features than internal factors, the maximum depth of the decision tree is much deeper than the internal. We can see the gini of each split and know how the decision tree wolks.

Machine learning for mixed factors:

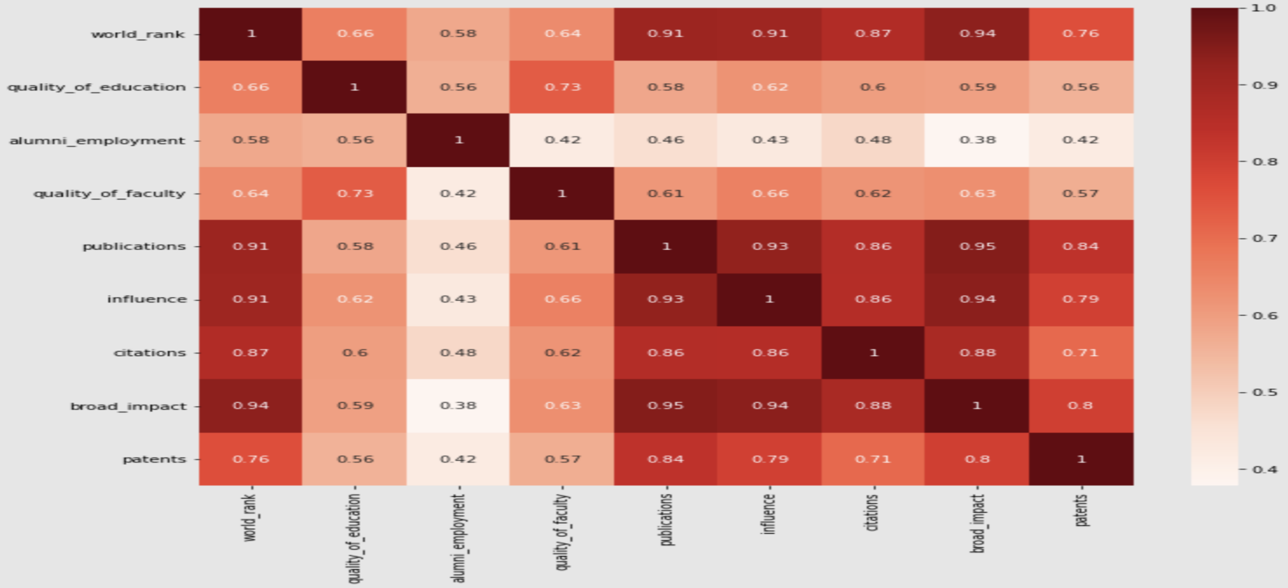


```
for name in all_models3.keys():
    model = all_models3[name]
    params = all_params3[name]
    gscv = GridSearchCV(estimator = model, param_grid = params, cv = 10)
    gscv.fit(mixed_train[['quality_of_education', 'alumni_employment', 'quality_of_faculty', 'publications', 'influence', 'citations', 'broad_impact', 'patents', 'Total
             "Percent of adults with a bachelor's degree or higher, 2015-19"]],
             mixed_train["world_rank"])
    print(f"best parameters are: {gscv.best_estimator_}")
    print(f"accuracy is: {gscv.best_score_}")
```

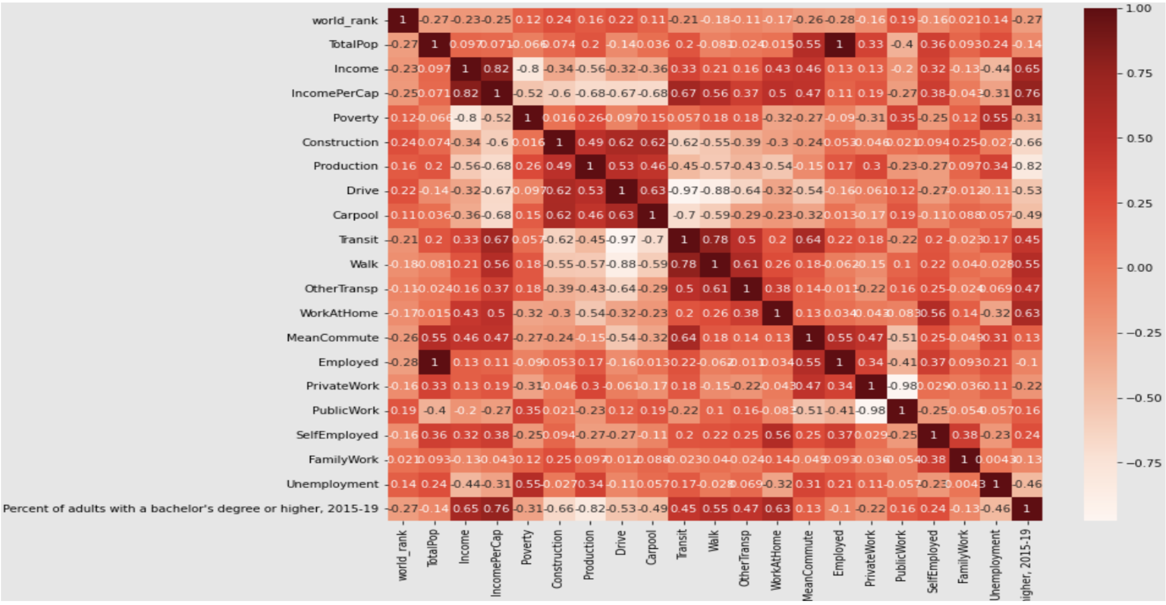
```
best parameters are: DecisionTreeClassifier(min_impurity_decrease=0.02)
accuracy is: 0.93
best parameters are: RandomForestClassifier(min_impurity_decrease=0.02, n_estimators=200)
accuracy is: 0.9400000000000001
best parameters are: Pipeline(steps=[('scale', StandardScaler()),
                ('model', LogisticRegression(C=0.059948425031893966))])
accuracy is: 0.9100000000000001
best parameters are: KNeighborsClassifier(n_neighbors=2)
accuracy is: 0.67
```

Correlation between features and label:

The graph shows the relationship between each feature and label. As the color becomes deeper, the feature has more relationship with the label. The correlation method we use is Pearson correlation. The higher the value, the stronger correlation between two variables.
And the correlation further indicates that internal factors are better predictors because variables in internal factors have stronger correlations with world rank than those in external factors.
Internal factor:



External factor:

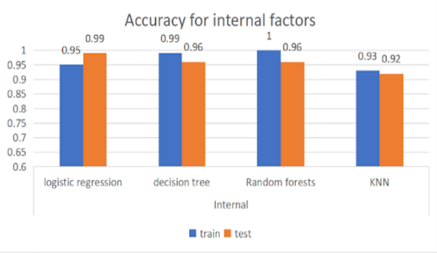


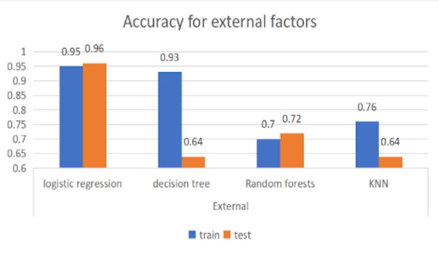
Accuracy for each type of factor:
Highest for Internal: 0.94（random forest）
Highest for External: 0.69(random forest & logistic regression
Highest for Mixed: 0.94(random forest)

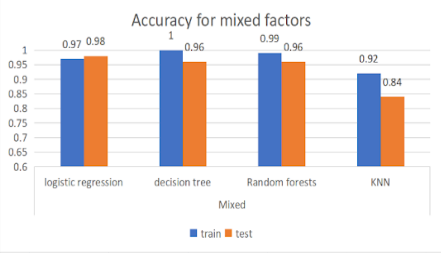
Findings:

1.Internal prediction >mixed prediction > better than external

Both training accuracy and test accuracy is over 90%. In external factors we can see that logistic regression is all over 90% but the other three groups, especially random forest and KNN, are below 80%. In accuracy of mixed factors, although most of them are kind of the same with internal factors, KNN in the accuracy of mixed factors is lower than the KNN from internal factors.

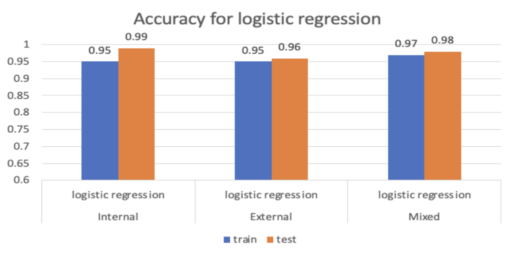


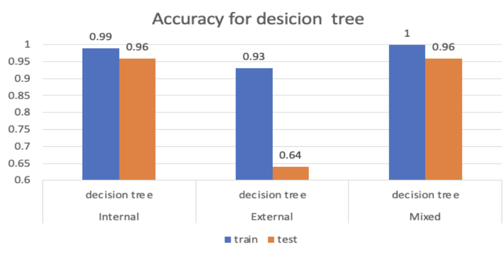


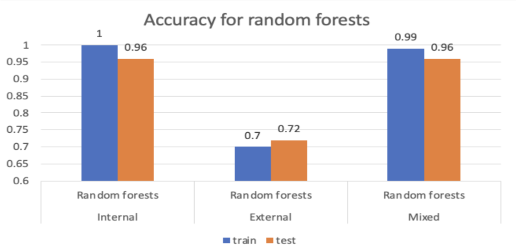


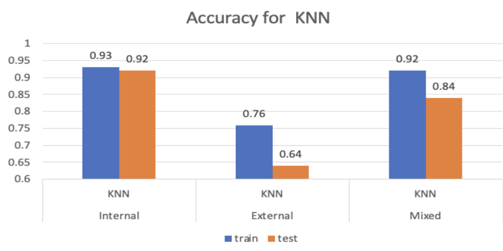
2.Logistic regression is the best overall.

There are four different modules as we can see in the screenshot below. Both training accuracy and test accuracy in logistic regression are over 95% from our module. However, in the accuracy of the decision tree, internal and mixed factors are better than the external one because the test accuracy of external factors is only 64%. So compared with the other three modules, accuracy of logistic regression is the best module we found.

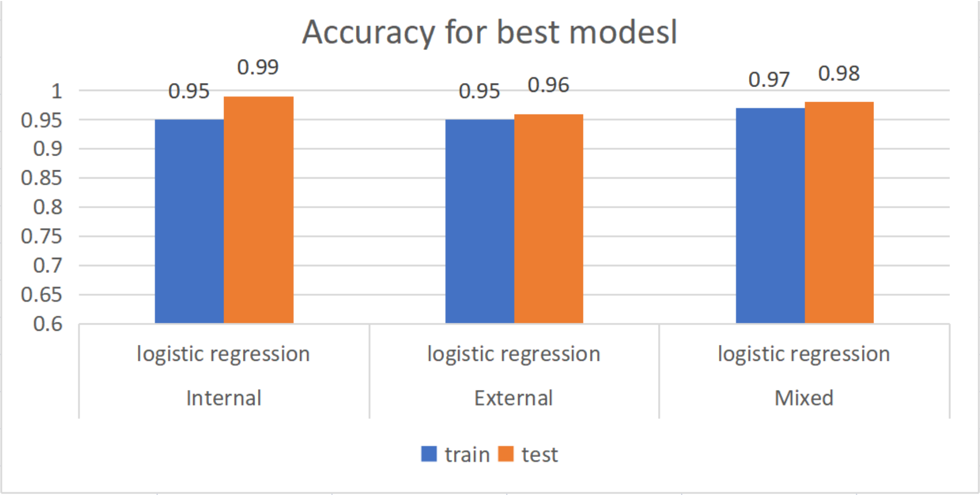








3. Logistic regression, all over 95% test accuracy



Results:

1.Internal prediction is better than mixed prediction and mixed prediction better than external because internal prediction is the best indicator or factor for university rank.

2. Logistic regression is the best overall.

3.Pearson Correlation proves our outcomes because internal features are better than external features.

Limitation:

1.Only use one rank data, different rank data may have different results.

2.Data of internal factors may be different from the real situation. (man-made data, subjective to the people who create the data). For example, quality_of_faculty is an abstract concept which lacks acknowledged standards to evaluate.

Conclusion:

Based on the findings, results and limitations we talked about above. We need more data to help us with training modules.