Pavit Singh MathLogic Summer Internship Report

Work Done and Documentation

Abstract

During the course of my internship I worked on evaluation of feature selection algorithms to better facilitate model training and evaluation

Contents

1. Business Objective

The primary business objective of implementing feature selection algorithms is to enhance the predictive performance and interpretability of machine learning models. By identifying and selecting the most relevant features, we aim to achieve the following goals:

1. **Improve Model Accuracy**: Feature selection helps in removing irrelevant or redundant data, thereby improving the model's accuracy and predictive power. This ensures that the insights derived from the model are more reliable and actionable.
2. **Reduce Computational Cost**: By minimizing the number of features, we can significantly reduce the computational resources required for model training and prediction. This leads to faster model development and deployment, optimizing operational efficiency.
3. **Enhance Interpretability**: A model with fewer, more meaningful features is easier to interpret and understand. This transparency is crucial for gaining stakeholder trust and facilitating informed decision-making processes.
4. **Mitigate Overfitting**: By focusing on the most relevant features, feature selection helps in reducing the risk of overfitting, where the model performs well on training data but poorly on unseen data. This enhances the model's generalizability to new, unseen data.
5. **Facilitate Compliance and Governance**: In sectors with strict regulatory requirements, such as finance and healthcare, feature selection ensures that the models are compliant by using only permissible and justifiable data inputs. This helps in meeting compliance standards and avoiding legal complications.

By leveraging feature selection algorithms, we aim to deliver robust, efficient, and interpretable machine learning solutions that drive value for our clients and support their strategic business objectives.

2. Approach

In this section, we discuss the methodology implemented for feature selection based on a paper and a reinforcement learning (RL) library. The objective was to benchmark the performance of these methods on two large datasets using XGBoost as a baseline.

1. Literature Review

I referred to the paper "Feature Selection: A Data Perspective" by Jundong Li et al. from Arizona State University. This paper provides a comprehensive overview of recent advances in feature selection research, especially in the context of big data. The paper categorizes feature selection methods into four main groups: similarity-based, information-theoretic-based, sparse learning-based, and statistical-based methods. These methods aim to build simpler models, improve data mining performance, and prepare clean, understandable data .

2. Implementation of Feature Selection Methods

Based on the insights gained from the paper, I implemented a few of the feature selection algorithms provided by the author namely the methods: CFS, CIFE, CMIM, DISR, F_score, Fisher_score, Gini_index, ICAP, JMI, MIFS, MRMR, MIM, reliefF in supervised learning tasks.I also tried the following methods for unsupervised tasks, MCFS, NDFS, Laplacian-Score, SPEC.This is the link to the library https://github.com/jundongl/scikit-feature/tree/ master

Example codes can be found here as well as my zipped code folder: https://github.com/ jundongl/scikit-feature/tree/master/skfeature/example

The data format may not match up, It needs to be indexed to 0 (remove headers) and it only takes numerical features, change accordingly.This library can be found as an environment in my zipped folder I have provided that as well

Activating python env:

```
source test_venv/bin/activate
```

3. Reinforcement Learning for Feature Selection

Additionally, I explored a reinforcement learning approach for feature selection using the RL library. The RL-based method formulates the feature selection process as a Markov Decision Process (MDP) where the agent iteratively selects features to maximize a reward function, which in this case is the model accuracy. This approach is detailed in an article on Towards Data Science: Reinforcement Learning for Feature Selection.

The RL library had issues being installed in python so I had to import the folder with the working code from GitHub and run my experiments within that folder. The folder is by the name “FSRLearning”.

This library just gave the rankings with had to be run separately in other notebooks, this is also a very computationally expensive library to run and takes a long time.

4. Benchmarking and Performance Evaluation

The selected feature selection methods were benchmarked on two large datasets. The performance was evaluated using XGBoost, a gradient boosting framework that is highly efficient and widely used for machine learning tasks. The evaluation metrics included accuracy, AUC-ROC, AUC-PR, and computational efficiency.

XgBoost parameters chosen:

```
xgb.XGBClassifier( colsample_bytree=
0.7,learning_rate=0.01,subsample= 0.8, max_depth=
2, n_estimators= 500, objective="binary:logistic")
```

The initial data explorations on data_4 were done with RandomForrest classifiers, Logistic Regression and Support Vector Machines. The metrics for evaluation used there were Accuracy and ROC-AUC. For unsupervised learning, the mean values of features were chosen for seeing if there is any appreciable difference in clustering capabilities.

Datasets

1. **Data_4**: It is a data with 43 features and 3 lakh rows, this was first used to benchmark the libraries computational load.
2. **Data_5:** It is a data with 5000+ features and 1lakh rows, this was highly compute expensive, further evaluation of data was done using features selected by xgboost and a smaller list of 250 features which greatly reduces computation needs. The unsupervised evaluation was done by

Experimental Setup

1. **Baseline Model**: XGBoost was used as the baseline model to evaluate the effectiveness of the feature selection methods. The model was trained and tested on the original datasets without any feature selection.
2. **Feature Selection and Model Training**: Each feature selection method was applied to the datasets, these either returned a ranking or a subset of features the selected features were used to train the XGBoost model in intervals. The performance of the models with selected features was compared to the baseline model.

3. Detailed Explana.on on Algorithms

Supervised Learning Feature Selection Methods

1. **CFS (Correlation-based Feature Selection)**:
    - **Purpose**: Identifies feature subsets that are highly correlated with the target variable while having low inter-correlation.
    - **Approach**: Evaluates subsets of features based on a heuristic that combines the predictive ability of each feature and the redundancy among them. It uses a correlation-based heuristic to select features that are useful in predicting the class.
2. **CIFE (Conditional Infomax Feature Extraction)**:
    - **Purpose**: Selects features by maximizing the conditional mutual information.
    - **Approach**: Selects features that provide the maximum information about the target variable, conditioned on already selected features. It aims to maximize the information gained about the class while minimizing redundancy.
3. **CMIM (Conditional Mutual Information Maximization)**:
    - **Purpose**: Maximizes the mutual information between features and the target variable, conditioned on previously selected features.
    - **Approach**: Evaluates each feature's ability to provide unique information about the target variable that isn't already provided by the selected features.

This method ensures that the selected features are both relevant and non-redundant.

4. **DISR (Double Input Symmetrical Relevance)**:
    - **Purpose**: Considers the symmetrical relevance of features.
    - **Approach**: Evaluates the relevance of a feature in the context of other features and the target. It uses a symmetrical relevance score to ensure that the selected features collectively provide significant information about the target.
5. **F-Score**:
    - **Purpose**: Measures the discrimination power of features.
    - **Approach**: Calculates the F-score for each feature, which is the ratio of between-class variance to within-class variance. Features with higher F-scores are better at distinguishing between classes.
6. **Fisher Score**:
    - **Purpose**: Evaluates feature importance based on class separability.
    - **Approach**: Similar to the F-score, but specifically designed for binary classification tasks. It calculates the ratio of variance between classes to variance within classes for each feature.
7. **Gini Index**:
    - **Purpose**: Measures the impurity or purity of features.
    - **Approach**: Used in decision tree algorithms, the Gini index measures how often a randomly chosen element would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.
8. **ICAP (Interaction Capping)**:
    - **Purpose**: Considers both feature relevance and redundancy.
    - **Approach**: Evaluates features based on their mutual information with the target variable and penalizes redundancy with already selected features. It aims to balance relevance and redundancy.
9. **JMI (Joint Mutual Information)**:
    - **Purpose**: Maximizes the joint mutual information between features and the target.
    - **Approach**: Selects features that collectively provide the maximum amount of information about the target variable. It considers the mutual information of feature pairs with the target.
10. **MIFS (Mutual Information Feature Selection)**:
    - **Purpose**: Selects features based on mutual information while penalizing redundancy.
    - **Approach**: Evaluates features by their mutual information with the target and subtracts a redundancy term to penalize features that provide redundant information.
11. **MRMR (Minimum Redundancy Maximum Relevance)**:
    - **Purpose**: Balances feature relevance with minimal redundancy.

- **Approach**: Selects features that have high mutual information with the target and low mutual information with each other. It ensures that selected features are both relevant and non-redundant.

12. **MIM (Mutual Information Maximization)**:
    - **Purpose**: Selects features that maximize mutual information with the target.
    - **Approach**: Evaluates features based solely on their mutual information with the target variable, ignoring redundancy considerations.
13. **ReliefF**:
    - **Purpose**: Robustly evaluates feature importance by considering the ability to distinguish between instances that are near each other.
    - **Approach**: Iteratively selects features by calculating their ability to distinguish between instances of different classes in a neighborhood. It calculates the difference between feature values of nearest neighbor instances of the same and different classes.

Unsupervised Learning Feature Selection Methods

1. **MCFS (Multi-Cluster Feature Selection)**:
    - **Purpose**: Identifies features that preserve the multi-cluster structure of the data.
    - **Approach**: Selects features based on their ability to maintain the clustering structure of the data. It uses spectral clustering techniques to identify features that contribute most to the separation of clusters.
2. **NDFS (Nonnegative Discriminative Feature Selection)**:
    - **Purpose**: Selects features that are non-negative and discriminative.
    - **Approach**: Combines discriminative analysis and L2,1-norm minimization to select features that are useful for clustering tasks. It ensures that selected features are both informative and non-negative, which is useful for certain types of data like text or image data.
3. **Laplacian Score**:
    - **Purpose**: Evaluates feature importance by their locality-preserving power.
    - **Approach**: Measures the feature's ability to preserve the local structure of the data. Features that keep similar instances close to each other in the transformed space are given higher scores.
4. **SPEC (Spectral Feature Selection)**:
    - **Purpose**: Uses spectral graph theory to evaluate feature importance.
    - **Approach**: Constructs a similarity graph of the data and uses eigenvalues and eigenvectors to identify features that best preserve the graph structure. It aims to select features that capture the intrinsic geometry of the data manifold.

4. Results

For Data 4

1. Introduction

This report presents the results of applying various feature selection techniques on a dataset and evaluating the performance using the XGBoost algorithm. The performance metrics considered are the Area Under the Receiver Operating Characteristic curve (ROC AUC) and the Logistic Regression Precision-Recall AUC (LR-PR).

2. Baseline Model

The baseline model was trained using all 43 features available in the dataset.

Performance:

- **Train ROC:** 0.7371
- **Train LR-PR:** 0.2201
- **Test ROC:** 0.7355
- **Test LR-PR:** 0.2155

3. Feature Selection Techniques and Results

3.1. Correlation-based Feature Selection (CFS)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6819	0.1697	0.6847	0.1668
6	0.6859	0.1732	0.6888	0.1700

3.2. Conditional Infomax Feature Extraction (CIFE)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6048	0.1123	0.6024	0.1094
6	0.6134	0.1172	0.6141	0.1156
7	0.6270	0.1242	0.6271	0.1234
8	0.6306	0.1295	0.6299	0.1279
9	0.6308	0.1357	0.6293	0.1328
10	0.6838	0.1682	0.6797	0.1629
15	0.6902	0.1732	0.6855	0.1669
20	0.6907	0.1733	0.6861	0.1667

25	0.7236	0.2071	0.7209	0.2031
30	0.7234	0.2070	0.7207	0.2031
35	0.7253	0.2093	0.7230	0.2052
40	0.7267	0.2108	0.7246	0.2068

3.3. Conditional Mutual Information Maximisation (CMIM)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6442	0.1390	0.6405	0.1383
6	0.6490	0.1402	0.6456	0.1395
7	0.6491	0.1378	0.6454	0.1371
8	0.6493	0.1379	0.6458	0.1375
9	0.6497	0.1381	0.6460	0.1374
10	0.6510	0.1396	0.6476	0.1402
15	0.6602	0.1449	0.6548	0.1444
20	0.6571	0.1430	0.6512	0.1409
25	0.6623	0.1498	0.6572	0.1467
30	0.6994	0.1832	0.6954	0.1758
35	0.7032	0.1871	0.6996	0.1798
40	0.7036	0.1874	0.7002	0.1803

3.4. Double Input Symmetrical Relevance (DISR)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6701	0.1633	0.6718	0.1616
6	0.6709	0.1647	0.6727	0.1612
7	0.6791	0.1695	0.6801	0.1664
8	0.6793	0.1696	0.6801	0.1663
9	0.6793	0.1698	0.6802	0.1670
10	0.6793	0.1701	0.6801	0.1672
15	0.7251	0.2056	0.7233	0.2028
20	0.7254	0.2057	0.7233	0.2028
25	0.7288	0.2113	0.7270	0.2080
30	0.7288	0.2112	0.7268	0.2077

35	0.7290	0.2115	0.7269	0.2079
40	0.7305	0.2133	0.7287	0.2094

3.5. F-Score

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.7276	0.2103	0.7259	0.2069
6	0.7274	0.2100	0.7255	0.2064
7	0.7290	0.2110	0.7273	0.2079
8	0.7284	0.2103	0.7266	0.2072
9	0.7285	0.2108	0.7268	0.2077
10	0.7291	0.2112	0.7274	0.2081
15	0.7294	0.2111	0.7275	0.2078
20	0.7294	0.2114	0.7274	0.2082
25	0.7301	0.2121	0.7282	0.2086
30	0.7300	0.2124	0.7282	0.2089
35	0.7300	0.2127	0.7284	0.2092
40	0.7304	0.2131	0.7287	0.2095

3.6. Gini Index

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.7280	0.2099	0.7262	0.2068
6	0.7285	0.2105	0.7268	0.2071
7	0.7294	0.2110	0.7278	0.2079
8	0.7295	0.2111	0.7275	0.2078
9	0.7298	0.2119	0.7277	0.2083
10	0.7298	0.2121	0.7278	0.2086
15	0.7306	0.2132	0.7286	0.2095
20	0.7303	0.2129	0.7282	0.2091
25	0.7307	0.2135	0.7284	0.2094
30	0.7306	0.2126	0.7287	0.2098
35	0.7303	0.2127	0.7284	0.2095
40	0.7304	0.2129	0.7287	0.2098

3.7. Interaction Capping (icap)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6442	0.1390	0.6405	0.1383
6	0.6490	0.1402	0.6456	0.1395
7	0.6491	0.1378	0.6454	0.1371
8	0.6493	0.1379	0.6458	0.1375
9	0.6497	0.1381	0.6460	0.1374
10	0.6510	0.1396	0.6476	0.1402
15	0.6602	0.1449	0.6548	0.1444
20	0.6571	0.1430	0.6512	0.1409
25	0.6623	0.1498	0.6572	0.1467
30	0.6994	0.1832	0.6954	0.1758
35	0.7032	0.1871	0.6996	0.1798
40	0.7036	0.1874	0.7002	0.1803

3.8. Joint Mutual Information (JMI)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6694	0.1628	0.6659	0.1610
6	0.6702	0.1642	0.6671	0.1625
7	0.6788	0.1691	0.6748	0.1672
8	0.6787	0.1691	0.6745	0.1671
9	0.6794	0.1695	0.6752	0.1675
10	0.6867	0.1752	0.6826	0.1731
15	0.7252	0.2055	0.7217	0.2018
20	0.7251	0.2055	0.7215	0.2018
25	0.7292	0.2119	0.7258	0.2082
30	0.7290	0.2116	0.7254	0.2076
35	0.7290	0.2114	0.7252	0.2074
40	0.7306	0.2131	0.7270	0.2090

3.9. Mutual Information Feature Selection (MIFS)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.5892	0.1206	0.5881	0.1189
6	0.5926	0.1237	0.5918	0.1219
7	0.6101	0.1224	0.6083	0.1209
8	0.6233	0.1273	0.6215	0.1250
9	0.6247	0.1277	0.6228	0.1256
10	0.6280	0.1293	0.6260	0.1269
15	0.6349	0.1317	0.6332	0.1297
20	0.6447	0.1384	0.6429	0.1359
25	0.6976	0.1818	0.6951	0.1783
30	0.6984	0.1827	0.6960	0.1794
35	0.7007	0.1841	0.6982	0.1809
40	0.7080	0.1903	0.7053	0.1876

3.10. Mutual Information Maximisation (MIM)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6778	0.1702	0.6759	0.1689
6	0.6807	0.1721	0.6788	0.1705
7	0.6861	0.1761	0.6835	0.1742
8	0.6899	0.1810	0.6876	0.1789
9	0.6899	0.1815	0.6875	0.1789
10	0.6895	0.1811	0.6870	0.1785
15	0.6939	0.1872	0.6912	0.1847
20	0.6999	0.1914	0.6975	0.1883
25	0.7023	0.1943	0.6998	0.1912
30	0.7022	0.1944	0.6997	0.1913
35	0.7148	0.2023	0.7122	0.1989
40	0.7154	0.2032	0.7126	0.2001

3.11. Minimum Redundancy Maximum Relevance (MRMR)

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6701	0.1633	0.6680	0.1616

6	0.6709	0.1647	0.6689	0.1612
7	0.6791	0.1695	0.6778	0.1664
8	0.6793	0.1696	0.6779	0.1663
9	0.6793	0.1698	0.6780	0.1670
10	0.6793	0.1701	0.6780	0.1672
15	0.7251	0.2056	0.7233	0.2028
20	0.7254	0.2057	0.7233	0.2028
25	0.7288	0.2113	0.7270	0.2080
30	0.7288	0.2112	0.7268	0.2077
35	0.7290	0.2115	0.7269	0.2079
40	0.7305	0.2133	0.7287	0.2094

3.12. ReliefF

No. of Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
5	0.6798	0.1696	0.6780	0.1670
6	0.6815	0.1706	0.6798	0.1682
7	0.6850	0.1730	0.6830	0.1703
8	0.6884	0.1765	0.6862	0.1738
9	0.6884	0.1766	0.6863	0.1739
10	0.6888	0.1769	0.6866	0.1742
15	0.6940	0.1831	0.6916	0.1798
20	0.6958	0.1865	0.6935	0.1826
25	0.6991	0.1894	0.6964	0.1855
30	0.6992	0.1894	0.6966	0.1857
35	0.6990	0.1895	0.6963	0.1858
40	0.6995	0.1901	0.6968	0.1864

4. Conclusion

The performance metrics reveal that the feature selection techniques F-Score, and Gini Index generally provide the best results in terms of ROC AUC and LR-PR AUC. The baseline model's performance is competitive, but the models with fewer features often show similar or improved performance, indicating effective feature selection.

5. Recommendations

- Use Voting, F-Score, or Gini Index for feature selection in future model development for optimal performance.
- Validate feature selection techniques on different datasets to ensure generalizability.

Data 5

Using the best techniques in data 4 we calculated for data 5

1. Baseline

- **Model**: XGBoost
- **Dataset**: data_5
- **Features**: All
  - **Train ROC**: 0.6694
  - **Train LR-PR**: 0.2473
  - **Test ROC**: 0.6534
  - **Test LR-PR**: 0.2258
- **Features**: 1525 (reduced by XGBoost)
  - **Train ROC**: 0.6686
  - **Train LR-PR**: 0.2464
  - **Test ROC**: 0.6521
  - **Test LR-PR**: 0.2256

2. Gini Index

- **Model**: XGBoost
- **Dataset**: data_5

Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
20	0.6467	0.2322	0.6307	0.2138
40	0.6486	0.2341	0.6321	0.2155
60	0.6523	0.2353	0.6355	0.2165
80	0.6536	0.2363	0.6360	0.2167
100	0.6555	0.2377	0.6379	0.2186
120	0.6558	0.2379	0.6383	0.2192
140	0.6581	0.2395	0.6396	0.2201
160	0.6586	0.2398	0.6397	0.2202
180	0.6589	0.2400	0.6404	0.2204
200	0.6588	0.2398	0.6407	0.2206
400	0.6647	0.2432	0.6471	0.2227
600	0.6657	0.2439	0.6475	0.2228
800	0.6695	0.2469	0.6533	0.2259
1000	0.6695	0.2470	0.6532	0.2264
1200	0.6695	0.2470	0.6536	0.2261
1400	0.6694	0.2468	0.6527	0.2257

1525	0.6693	0.2466	0.6528	0.2257

3. F-Score

- **Model**: XGBoost
- **Dataset**: data_5

Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
20	0.6261	0.2203	0.6146	0.2059
40	0.6303	0.2218	0.6191	0.2071
60	0.6321	0.2239	0.6198	0.2081
80	0.6447	0.2302	0.6330	0.2149
100	0.6540	0.2341	0.6407	0.2178
120	0.6551	0.2351	0.6414	0.2183
140	0.6564	0.2362	0.6420	0.2181
160	0.6600	0.2395	0.6456	0.2213
180	0.6602	0.2397	0.6458	0.2214
200	0.6603	0.2396	0.6455	0.2214
400	0.6648	0.2428	0.6488	0.2227
600	0.6685	0.2461	0.6537	0.2260
800	0.6684	0.2460	0.6535	0.2257
1000	0.6692	0.2462	0.6539	0.2267
1200	0.6693	0.2466	0.6537	0.2267
1400	0.6695	0.2468	0.6530	0.2256
1525	0.6692	0.2469	0.6526	0.2258

4. T-Score

- **Model**: XGBoost
- **Dataset**: data_5

Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
20	0.6320	0.2231	0.6228	0.2094
40	0.6379	0.2229	0.6273	0.2117
60	0.6399	0.2235	0.6359	0.2148
80	0.6426	0.2315	0.6376	0.2153
100	0.6546	0.2319	0.6392	0.2174

120	0.6591	0.2338	0.6417	0.2182
140	0.6592	0.2346	0.6426	0.2194
160	0.6592	0.2364	0.6456	0.2210
180	0.6595	0.2375	0.6464	0.2212
200	0.6605	0.2385	0.6473	0.2213
400	0.6646	0.2419	0.6499	0.2221
600	0.6680	0.2449	0.6527	0.2227
800	0.6682	0.2462	0.6528	0.2230
1000	0.6677	0.2463	0.6529	0.2237
1200	0.6688	0.2462	0.6532	0.2242
1400	0.6685	0.2461	0.6529	0.2241
1525	0.6683	0.2464	0.6547	0.2247

RL library results for data_5

Baseline Model

The baseline model was trained using all 248 features available in the dataset.

Performance:

- Train ROC: 0.6694
- Train LR-PR: 0.2473
- Test ROC: 0.6534
- Test LR-PR: 0.2258

3. Feature Selection Results

Features	Train ROC	Train LR-PR	Test ROC	Test LR-PR
20	0.6076	0.1974	0.5962	0.1879
40	0.6304	0.2125	0.6118	0.1960
60	0.6333	0.2162	0.6098	0.1971
80	0.6350	0.2181	0.6107	0.1963
100	0.6376	0.2189	0.6150	0.1978
120	0.6397	0.2207	0.6148	0.1990
140	0.6413	0.2220	0.6163	0.2001
160	0.6415	0.2223	0.6163	0.2006
180	0.6434	0.2238	0.6196	0.2021
200	0.6450	0.2251	0.6207	0.2026

220	0.6470	0.2288	0.6239	0.2057
240	0.6542	0.2327	0.6312	0.2106

5. Conclusions

Statistical based measures for feature selection, namely gini, f_score and t_score were standout performers which showed near baseline metrics at much reduced features. These techniques are also good as they use less compute and are much faster, they also take a lot less ram.

For unsupervised learning tasks effec*ve clustering was shown with mean values of features being different, although they followed a paBern of having clear separa*on of 2 classes (eg 5 clusters 3 will have nega*ve mean values 2 will have posi*ve) the mean values while being different have shown this separa*on,

The feature selec*on through Reinforcement Learning is unreliable and computa*onally very expensive as its pseudo random walk using the Markov decision process is not very efficient, it took 18hrs for data_4, and around 5 days of running for data_5.

The unsupervised learning algorithms are fast but require a lot of RAM since they try to map rela*onships between each data point, so an abundance of ram is necessary. In general the informa*on theory based models like CIFE CMIM JMI MIFS were unreliable and not effec*ve while sta*s*cal models based on simpler techniques fared much beBer.

6. References

Paper:

- Brown, G., Pocock, A., Zhao, M. J., & Luján, M. (2012). Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection. Journal of Machine Learning Research, 13(Jan), 27-66.

Article:

- Bouzin, H. (2020, November 3). Reinforcement Learning for Feature Selection. Towards Data Science. Retrieved from https://towardsdatascience.com/reinforcement-learning-for-feature-selection-be1e7eeb0acc.

Link to results excel for data_4, data_5: https://fnmathlogic-my.sharepoint.com/:x:/p/pavit_singh/EW3CT-_1BkNDnGIhDi6S6lMBwGqeN2H9dU7CpSAziL0yXQ?e=QuLSBJ&wdOrigin=TEAMS-MAGLEV.p2p_ns.rwc&wdExp=TEAMS-TREATMENT&wdhostclicktime=1720216838864&web=1

RL library results for data_5 with reduced 248 features

RLresults 1.xlsx

Laplacian_Score unsupervised for data_5

lapscore.xlsx
Spec unsupervised for data_5

spec_unsup

NDFS unsupervised for data_5

spec_unsup

MCFS unsupervised for data_5

unsupMCFS

The notion site below contains detailed graphs for data_4 with AUC_ROC, accuracy plotted over 3 different models
https://pait.notion.site/results-for-data4-9d1825cea30b4e47a908ebfc0f375a4b?pvs=4

The notion site below contains info about the outputs of the RL library

https://pait.notion.site/results-for-data4-9d1825cea30b4e47a908ebfc0f375a4b?pvs=4