Leveraging Machine Learning to Bridge the Gap between Simple and Complex Traits in Job Applicants

Data Science Implementation for Entrepreneurs - Final Report

Submitted by: Liav Slama, Raphael Lasry, Hadar Eshed, Yoni Alush and Ran Zilka

The Problem

During an interview, we can observe various traits such as attentiveness, calmness, confidence, communication skills, and creativity. However, predicting a candidate's persistence, a crucial trait for many roles, is challenging. Can we leverage machine learning to predict a candidate's complex traits like persistence based on their responses and observed traits during an interview?

Highlights

- Development of a machine learning model that connects observable traits from a physical interview to complex, unobservable traits.
- Utilization of XGBoost and Polynomial Regression to predict complex traits based on simple ones.
- Exploration of the potential of machine learning in enhancing traditional interview processes.

Useful terminology

1. Simple Traits: Observable characteristics during an interview, such as attentiveness and communication skills.
2. Complex Traits: Unobservable characteristics that can be inferred from simple traits, such as persistence and proactivity.
3. Trait Mapping: The process of linking simple traits to complex traits using machine learning algorithms.
4. Full model: Model trained on the full set of features
5. Compact model: Model trained on the subset of the most important features

S.M.A.R.T Goal

Specific: Our research question is precise: Can we develop a machine learning model that predicts complex traits such as persistence in job applicants based on simple traits observed during an interview?

Measurable: The success of our research question will be determined by the high predictive accuracy of our model in identifying complex traits. The performance of our model will be measured using appropriate metrics such as accuracy and Mean Squared Error (MSE).

Achievable: With the use of advanced machine learning methods like XGBoost and Polynomial Regression, and a well-prepared and understood dataset, the goal of developing such a model is achievable.

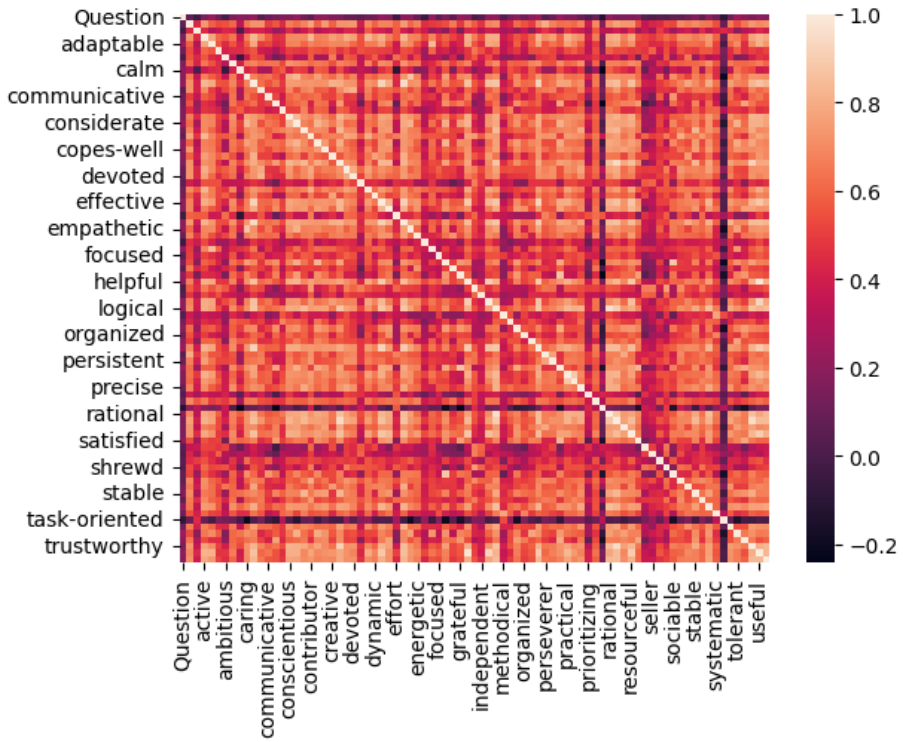
Relevant: This goal is highly relevant in the field of human resources, where predicting unobservable traits like persistence can significantly enhance the recruitment process and lead to better hiring decisions.

Time-Based: The goal is to develop this model by the end of the course, providing a clear timeline for achieving our objective.

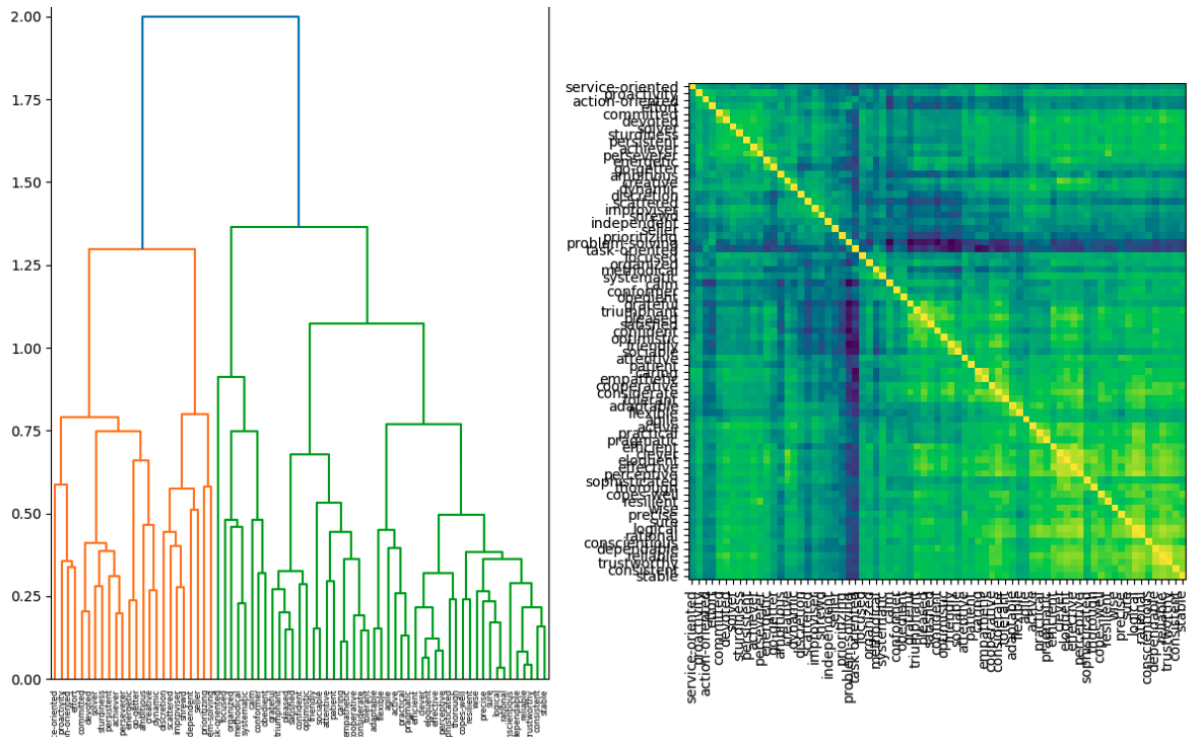
Data Preparation & Exploration

We began our journey by examining the data. We obtained 494 rows (questions asked), 84 applicants, and 83 scores for each question. After comprehending the dataset's traits, we discerned a distinction between simple and complex traits. We manually established a group of 15 complex traits.

It is evident that certain traits exhibit semantic similarity, such as "reliable" and "trustworthy." To explore highly correlated features beyond semantic similarity, we opted to visualize the correlation matrix of the feature set.



Moreover, we aim to verify if there exist clusters of extensively interrelated characteristics, indicating that all characteristics within a cluster are correlated with each other above a predefined threshold. To accomplish this clustering task, we perform hierarchical clustering based on Spearman rank-order correlations[9]:



It’s worth noting that we attempted an alternative approach for the same objective. We represented all features as a graph, with features as vertices and an edge between two vertices if the correlation between them is at least 0.85. We subsequently identified cliques within the graph, where each clique denotes a collection of strongly intercorrelated features.

We have improved our understanding of the dataset and are prepared to proceed with the data preparation phase.

Complex traits selection

The dataset was manually split into simple and complex traits recognized in the EDA phase. The simple traits were used as input for the machine learning models, with the goal of predicting the complex traits.

```
# Manual complex features selection
def split_simple_complex_traits(df):
    complex_traits = [
        "seller",
        "committed",
        "consistent",
        "considerate",
        "service-oriented",
        "action-oriented",
        "conscientious",
        "triumphant",
        "tolerant",
        "proactivity",
        "persistent",
        "independent",
        "pleased",
        "organized",
        "go-getter"
    ]
    target_traits_df = df[complex_traits]
    simple_df = df.drop(complex_traits, axis=1, inplace=False)
    return simple_df, target_traits_df
```

Applicant cohesion in train/test split

We decided to keep all questions from the same applicant together in either the training or test set. This preserves the correlation between an applicant's responses, reflecting their

unique characteristics and communication style. By avoiding a split of questions across sets, we ensure that our model recognizes general patterns applicable to all applicants, rather than specific patterns for individual applicants. This is crucial given our data's structure, with six questions per applicant and around 100 applicants. Our goal is for the model to predict traits based on response content and style, rather than individual idiosyncrasies.

Variance thresholding

We also considered the presence of almost invariant features, which are features that have similar values in all samples. These features were pruned as they do not contribute to the predictive power of the model[10]. For example, if a trait like 'confidence' received a similar score across all applicants, it would be considered an invariant feature and was therefore removed from the dataset. We used a variance threshold of 0.03 to identify and remove these invariant features.

Correlated features clustering

We used the correlated features clusters that were produced in the EDA phase and, for each cluster, we kept a single feature as the “cluster’s leader”. This helps reducing the dimensionality of the dataset and improve the efficiency of the model. For instance, if two traits like 'confidence' and 'assertiveness' were highly correlated, we might choose to represent them with a single feature.

The feature set, after all data preparation and feature pruning steps, looks like the following:

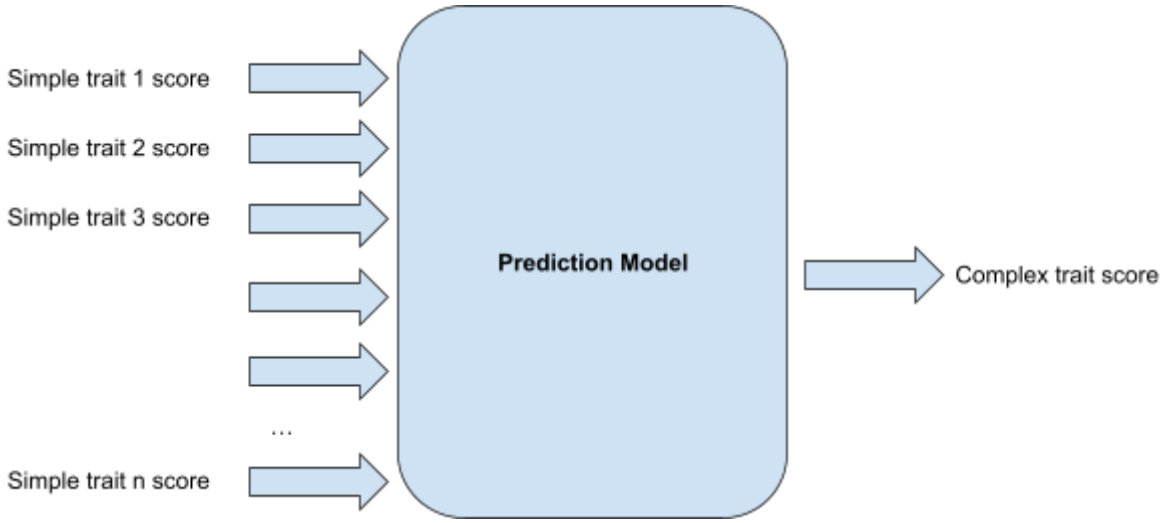
	achiever	active	adaptable	agile	ambitious	attentive	calm	caring	clever	confident	...	proactivity	problem-solving	scattered	shrewd	solver	sophisticated
0	0.080678	0.215110	0.146238	0.085734	0.069208	0.069946	0.037715	0.102040	0.216572	0.158371	...	0.075422	0.245341	0.213784	0.063607	0.075241	0.070486
1	0.627317	0.457680	0.293279	0.407006	0.399087	0.561051	0.191383	0.486489	0.482921	0.485677	...	0.920935	0.911456	0.223088	0.409734	0.540158	0.388744
2	0.474610	0.494967	0.941991	0.211321	0.539131	0.813721	0.097992	0.508986	0.606512	0.939887	...	0.611463	0.817547	0.262650	0.396920	0.677454	0.123203
3	0.465788	0.604448	0.417883	0.242591	0.559507	0.650982	0.106956	0.564496	0.587920	0.577517	...	0.631437	0.898215	0.426968	0.782801	0.423463	0.760178
4	0.621838	0.910976	0.950429	0.716624	0.336634	0.822668	0.845097	0.816513	0.790289	0.941401	...	0.779690	0.128161	0.286399	0.303824	0.449811	0.635565

5 rows × 44 columns

Modeling

Following the EDA and data preparation phase, we have a data set composed of features and target labels, where the labels are the complex traits we want to predict and the

features are the selected simple traits. Our objective is to predict complex traits using simple ones, therefore we will use a prediction model to do the job. Following is an abstract diagram of the model's interface:



More specifically, we train a model for each complex trait we want to predict.

We decided to try two types of prediction models to do the job, each with its own advantages. At the end of the process, we evaluate and compare both techniques under the same metrics and the model that performs the best will be selected. The two types are XGBoost (using the XGBoost Python Package implementation) and Polynomial Regression (using SKLearn implementation). We'll briefly explain the advantages of each.

Decision tree based models, as XGBoost, are considered "best-in-class" for small-medium tabular data sets[6]. Another fact that pushed us to use this algorithm is that XGBoost wins most of Kaggle competitions[7], which is a good general indicator of its performance. It provides high accuracy (which is our main metric) and does some data normalization on its own. Additionally, XGBoost, as well as other decision tree based algorithms, have high explainabilty and visibility which can be of use when explaining to the user why the model predicted the way it did.

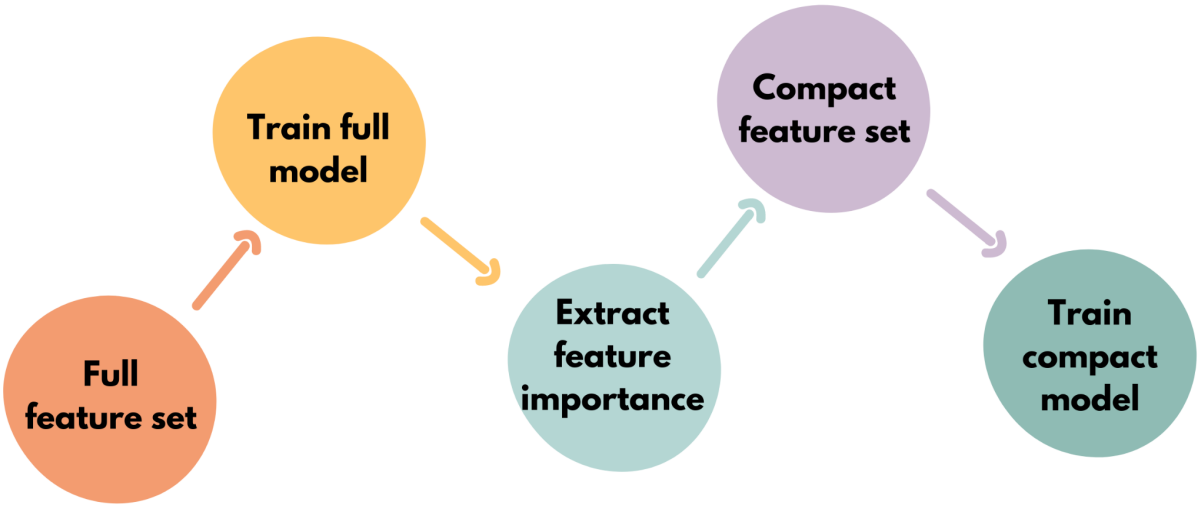
Polynomial regression represents well the importance of relationships between features and can fit a wide range patterns under it.[8] We assume that combinations between features is

highly significant in our use case, making polynomial regression a good candidate. On the other hand, this algorithm might be sensitive to outliers and overfitting.

Compact model

At this point we employ 44 features for prediction. In the real-world scenario, this entails the user (the interviewer) inputting scores for 44 distinct characteristics of the interviewee. In order to turn our model into a valuable and viable solution for the user, we should require much fewer features. We use “compact models” to achieve that.

After we train the model with all available features (the full model), we check for the most affecting features. Then we take the 5 (or any other number of) strongest features and re-train a model only with them (the compact model). If the compact model meets our target metrics, then it is a viable tool to predict interviewees’ complex traits.



Model Evaluation

We measure the performance of both models using two metrics: RMSE (Root Mean Squared Error) and a discrete accuracy. Additionally we measure how the compact model performed

compared to the full model. Following is an explanation on how we measure these metrics and why they're significant.

RMSE

Root-Mean-Square-Error is one measure to estimate the accuracy of our model's predicted values versus the actual or observed values. It measures the error in our predicted values when the target variable is a continuous number.

It is a good metric as it represents "how close" to reality our model is predicting results overall. On the other hand, it might be too sensitive to outliers, as few very bad predictions can have a significant negative effect on the metric.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Discrete accuracy

Additionally to a continuous accuracy metric (RMSE), we want to measure "how many good predictions we're doing". It is of course more abstract than just measuring RMSE as we have to define what a good prediction is, but it represents better our business goal as we want to offer a solution that predicts well for a high percentage of interviewees.

A prediction is considered correct if it deviates from the real value by no more than 0.1.

Then we can simply calculate accuracy as:

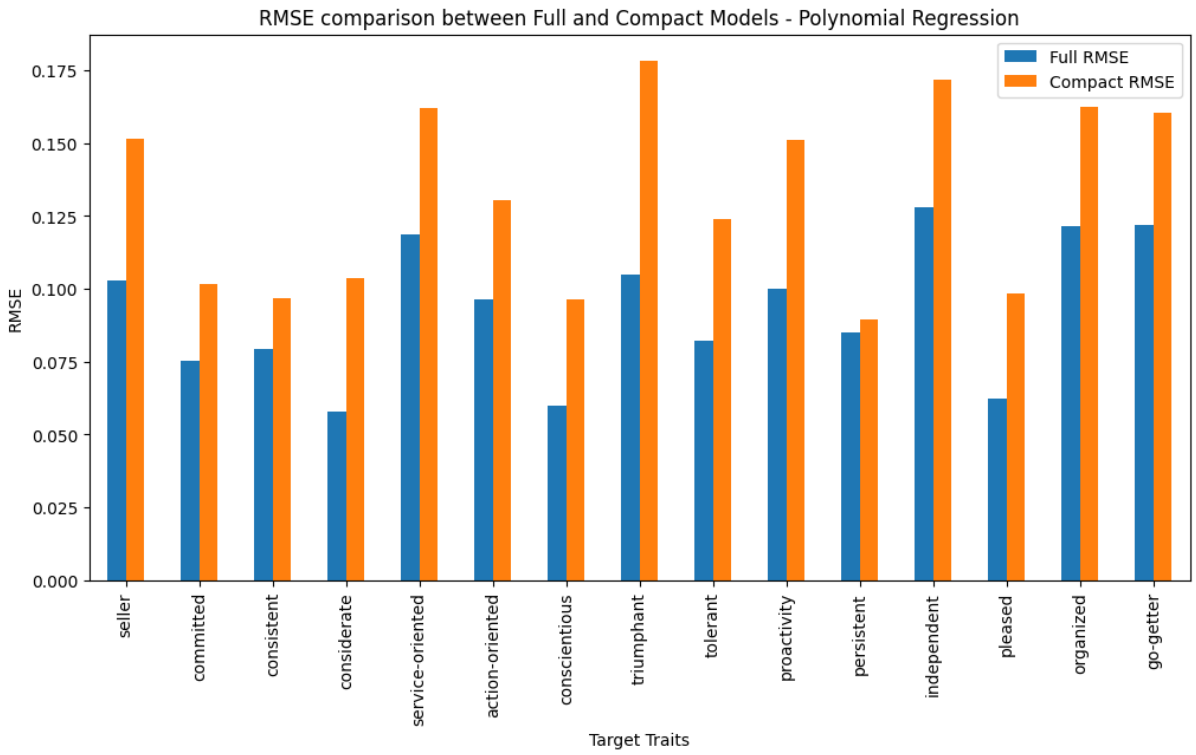
$$\frac{Correct\ Predictions}{All\ Predictions}$$

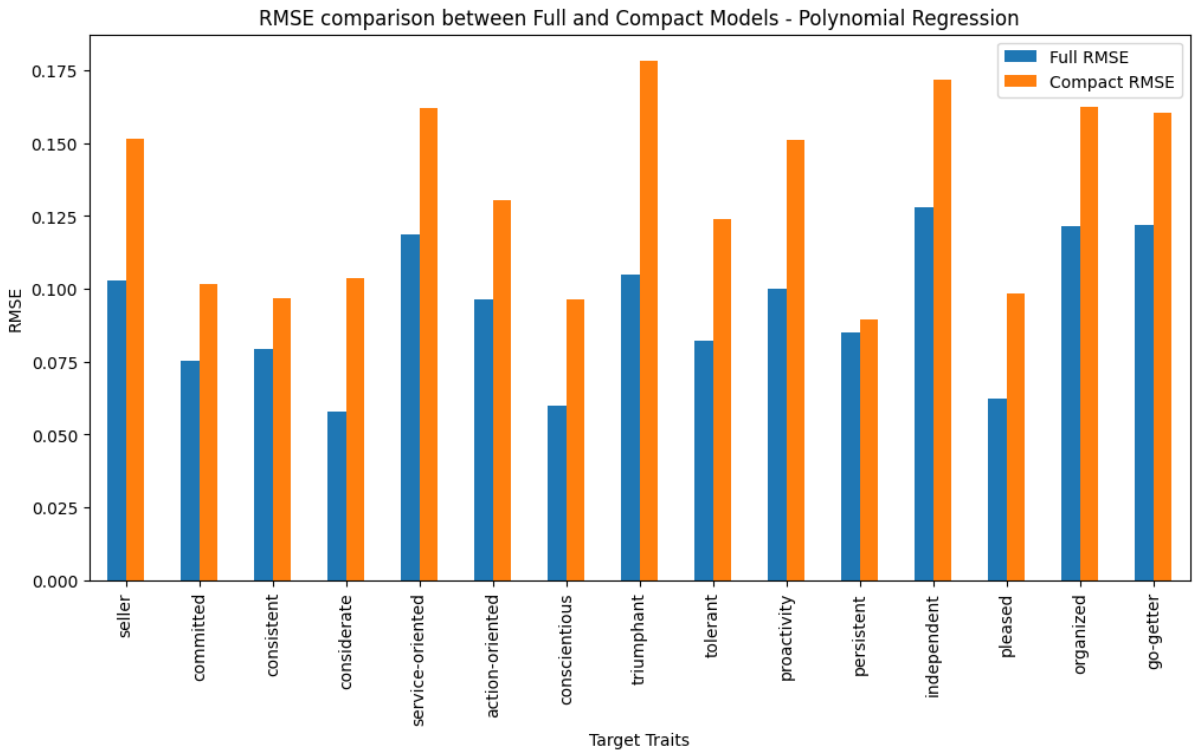
Full-Compact accuracy ratio

We compare the performance of the full and compact models using the RMSE and discrete accuracy ratio between the two. We want to achieve a ratio as close to 1 as possible, that would prove our compact model is as-good-as our full mode..

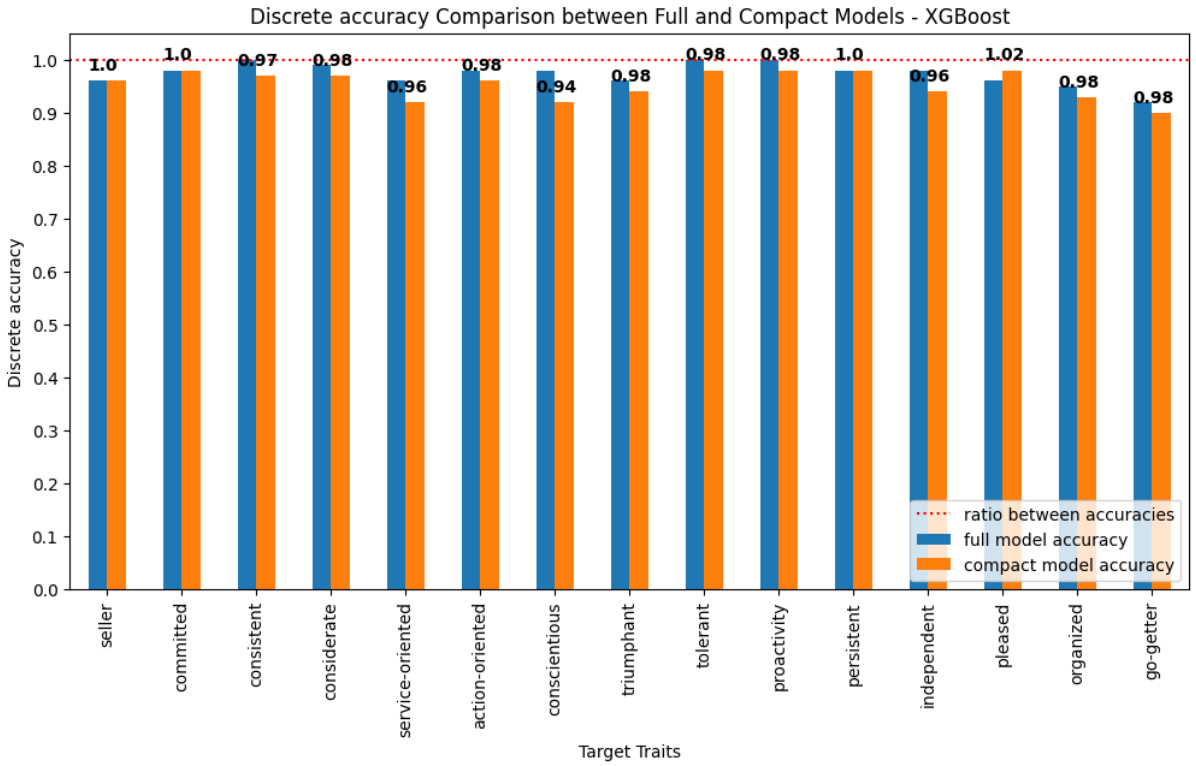
Results

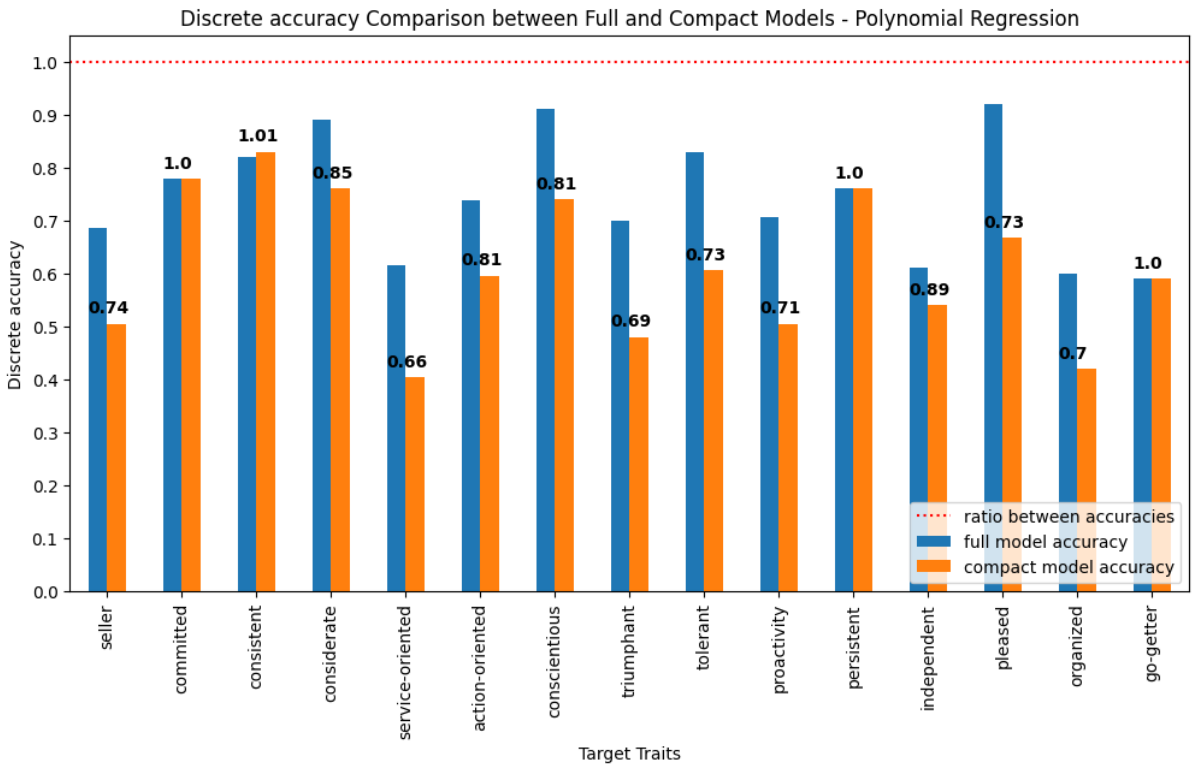
The most significant result is the level of accuracy that the models reached while predicting the target traits. Following are graphs describing the RMSE full and compact models:





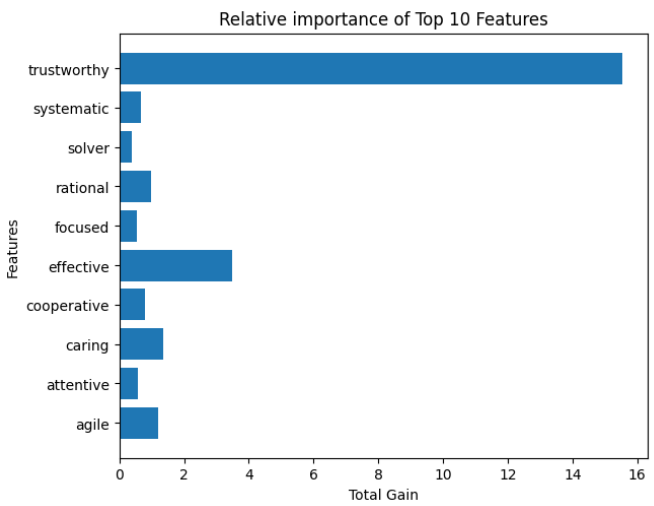
We measure also the discrete accuracy of the models:





Another noteworthy outcome is the importance of features in each model. They provide us with insight into which key features predict a specific label and enhance our visibility of the models. Following is an example of the feature importances of the model that predict how "conscientious" an interviewee is:

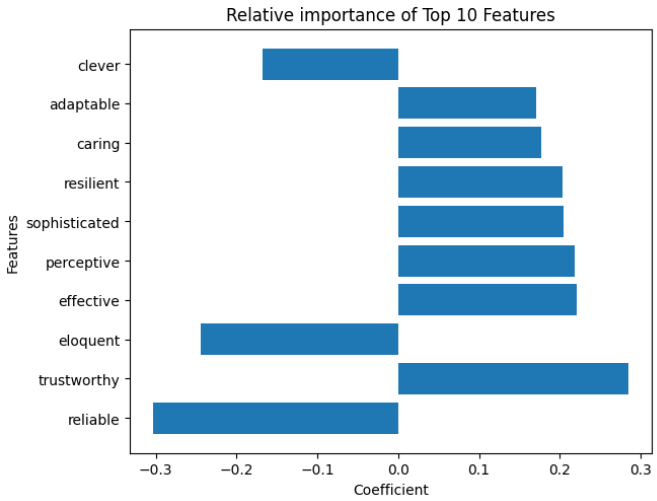
XGBoost - Prediction for 'conscientious'
Pruning : 000



RMSE: 0.02498456218821453

Discrete accuracy: 96.97%

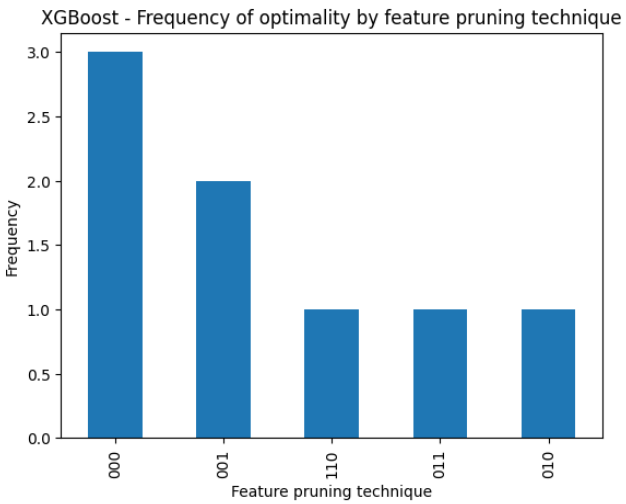
Polynomial Regression - Prediction for 'conscientious'
Pruning : 101

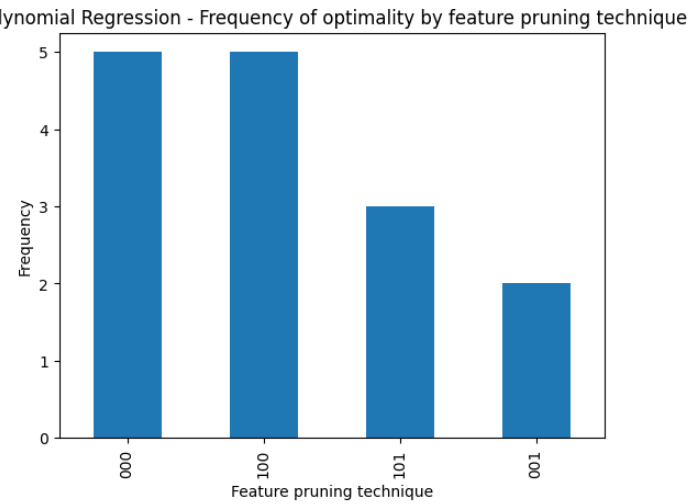


RMSE: 0.05919957944783357

Discrete accuracy: 92.0%

Finally, we analyzed the models' performance by evaluating the impact of data preparation and pruning techniques. For each target label, a model was trained using all possible combinations of data preparation and feature pruning techniques. The combination that yielded the lowest RMSE was chosen as the feature set to train the model. Following is a graph describing how many times each combination yielded the optimal model. For instance, we can see that the combination '000' yielded an optimal model 4 times, where '000' means that no manipulation was made on the original feature set. (A detailed explanation on what each bit means on the notebook).





Conclusions

Our work shows that machine learning can effectively bridge the gap between simple and complex traits in job applicants. This has significant implications for the field of human resources, suggesting that machine learning could be used to gain deeper insights into candidates' suitability for a role.

In our research, XGBoost had a significant advantage over Polynomial Regression yielding an accuracy of over 90% on all target labels even when using a compact set of 5 features. Therefore we recommend the XGBoost model as the tool to predict complex traits.

Furthermore, after our research points out that there is not a single data preparation and feature pruning strategy that is best-suited for all models. Each model benefits from a different strategy.

Further investigation could involve testing the model on publicly available HR datasets or even building our own dataset from real-world interviews. This would provide a more robust validation of the model's effectiveness and applicability in practical HR scenarios.

Bibliography

1. Schmidt, F. L., & Hunter, J. E. (1981). Predicting Job Performance: A Comparison of Expert Opinion and Research Findings. *International Journal of Selection and Assessment, 19*(2), 104-116. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/0169207089900861.
2. Schmidt, F. L., & Hunter, J. (1998). The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings. *Psychological Bulletin, 124*(2), 262-274. Retrieved from https://www.researchgate.net/publication/232564809_The_Validity_and_Utility_of_Selection_Methods_in_Personnel_Psychology.
3. Barrick, M. R., Stewart, G. L., & Piotrowski, M. (2002). Personality and Job Performance: Test of the Mediating Effects of Motivation Among Sales

Representatives. *Journal of Applied Psychology, 87*(1), 43-51. doi: 10.1037/0021-9010.87.1.43.

4. Sheremet, M., Aksimentiev, A., & Tkachenko, V. (2019). Extreme Gradient Boosting as a Method for Quantitative Structure–Activity Relationships. *Journal of Chemical Information and Modeling, 59*(12), 5090-5100. Retrieved from https://pubs.acs.org/doi/abs/10.1021/acs.jcim.6b00591.
5. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). A Gentle Introduction to Polynomial Regression and Overfitting. In *An Introduction to Statistical Learning* (pp. 123-140). Springer. doi: 10.1007/978-1-4614-7138-7_4.
6. Morde, V. (2019, April 8). XGBoost Algorithm: Long May She Reign! Medium; Towards Data Science. Retrieved from https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.
7. Gupta, S. (2020, February 28). Pros and cons of various Classification ML algorithms. Medium; Towards Data Science. Retrieved from https://towardsdatascience.com/pros-and-cons-of-various-classification-ml-algorithms-3b5bfb3c87d6.
8. Pant, A. (2019, January 13). Introduction to Linear Regression and Polynomial Regression. Medium; Towards Data Science. Retrieved from https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb.
9. *Permutation Importance with Multicollinear or Correlated Features*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/auto_examples/inspection/plot_permutation_importance_multicollinear.html
10. T, B. (2023, April 8). How to Use Variance Thresholding For Robust Feature Selection. *Medium*. https://towardsdatascience.com/how-to-use-variance-thresholding-for-robust-feature-selection-a4503f2b5c3f