The Crucial Role of Humans in Harnessing ChatGPT for Machine Learning Model Development

Emily Chan
Department of Computer Science
Western University
London, Canada
echan392@uwo.ca

Scott Quinn
Department of Computer Science
Western University
London, Canada
squinn43@uwo.ca

Yael Shteyer
Department of Computer Science
Western University
London, Canada
yshteyer@uwo.ca

***Abstract*—The rise of artificial intelligence tools, including ChatGPT, in recent years has vastly changed the way individuals complete tasks. Namely, artificial intelligence tools can be used to build machine learning models. Addressing the novelty of using such tools for machine learning applications, this research compares the performance of a human-created model against models generated by machine intelligence. To conduct this investigation, a human-made model with no artificial intelligence involvement, a model generated using ChatGPT, and a model generated using a fine-tuned version of ChatGPT were built and evaluated against a baseline problem. These models predict house prices using a scikit-learn dataset and are evaluated based on the mean squared error, coefficient of determination, and ease of use. Our findings show that, based on the identified criteria, the human-created model, an Extreme Gradient Boosting model, outperforms the other artificial intelligence models in technical and non-technical ways. This paper discovers that the best machine learning models will come from the combination of human expertise and artificial intelligence tools rather than the dominance of one. Further research might attempt to evaluate new artificial intelligence tools as well as expand on the findings of this research, by combining the power of human knowledge and artificial intelligence.**

***Index Terms*—AI, LLM, ML, MSE, R-squared**

I. Introduction

In the evolving landscape of artificial intelligence (AI), the development and implementation of machine learning (ML) models have been predominantly human-driven. However, recent advancements in AI, specifically in the capabilities of models like ChatGPT, suggest a potential shift in this industry. ChatGPT is a groundbreaking large language model (LLM) that has rapidly become the most quickly adopted of its kind. This study explores the creation and comparison of three distinct regression models: one traditionally created by human experts, another generated by ChatGPT, and a third using a finely tuned version of ChatGPT. The primary objective is to scrutinize the effectiveness and efficiency of these models in a standardized setting.

The focus of these models is to predict house prices, utilizing a dataset provided by scikit-learn, a machine learning library in Python. This research aims to limit human interaction in the construction of both the ChatGPT and fine-tuned ChatGPT models to gauge their autonomous capabilities in generating accurate and reliable predictions.

Despite the interest and discussion around the capabilities of AI tools like ChatGPT, there is a lack of empirical studies focusing on their direct comparison with human-generated models, especially in well-defined and controlled environments. This research seeks to fill this gap by employing a clean, well-documented dataset to facilitate an objective comparison of a popular machine learning problem.

The primary goal of this study is to identify the most effective model among the three models and to determine the extent of human intervention required to create a usable and accurate model using ChatGPT. This involves a comprehensive analysis of each model's performance metrics, drawing insights into their predictive accuracy, efficiency, and ease-of-use.

II. Background and Related Work

Currently, there is a lack of studies comparing the use of AI tools against human-generated models. However, a study published in 2018 compared human-made models to machine learning models [1]. The paper presents an empirical study comparing the performance of software engineers and machine learning algorithms in the context of a specific software development task: synthesizing the control structure of an autonomous streetlight application. The results of the study revealed that in some cases, software engineers outperform machine learning algorithms while in other cases, they do not. These indeterminate findings suggest the importance of understanding when automation is more effective than human involvement in performing specific software development tasks.

Research from 2020 attempts to bridge the gap between human-involvement for machine learning tasks by suggesting a web-based tool that allows users with no technical background to easily build ML models [2]. This enables any user to build a predictive model without necessarily understanding the background logistics or computations. This research projects aims to achieve a similar goal in testing whether AI tools are self-sufficient in building reliable ML models with minimal human interference. Likewise an AI tool, this web Graphical User Interface is easy to use, personalizable, and flexible for non-technical users. Using this resource, users are only able to edit parameters such as epochs, batch size, and the learning rate. Given that the overall complexity is hidden from the users, this tool mimics the process of using an AI tool to build a model with little human involvement.

OpenAI's ChatGPT operates as a generative AI, capable of understanding inputs in natural language and generating fresh, unique responses that include text, visual content, and code. This innovation represents the pinnacle of decades of AI research dating back to the 1960s. The distinctiveness of ChatGPT lies in its exceptional accuracy and focus on showcasing the potential of LLMs [3]. ChatGPT possesses the advanced ability to keep track of context within a conversation, which enables it to refine its responses and learn from the ongoing dialogue, thereby enhancing the relevance and accuracy of its answers.

A study from 2023 scrutinized ChatGPT's proficiency in executing data science tasks when guided by a specialist in the field [3]. The objective was to determine whether ChatGPT could augment these tasks' quality, efficiency, and maintenance. The evaluation centered on ChatGPT's capabilities in generating code, retrieving information, translating code across different languages, and the extent to which it could collaborate with humans. In this case study, a domain expert provided directives to ChatGPT. The principal results were the development of logistic regression and random forest classification models. The research concluded that ChatGPT positively impacts the set objectives, yet it still relies on human guidance and expertise to ensure continued effectiveness. The research was similar to this study, which evaluates the involvement of ChatGPT for a specific ML task.

III. Research Objectives

1) Create an accurate model utilizing class notes and other resources found, without the use of AI
2) Build a model with minimal human-intervention using ChatGPT
3) Build a model using a fine-tuned version of ChatGPT
4) Determine a criteria of metrics and properties of good model performance
5) Evaluate and compare the performance of non-AI and AI models

IV. Research Methodology

A. Methodology Background

1) Dataset: scikit-learn's California housing dataset was used for regression modelling [4]. The dataset consists of eight numeric variables. The target variable is the median house price of California districts. The data originates from a 1990 United States census and includes 20640 observations. The variables are shown in Figure 1 with descriptions from scikit-learn [4]. 20% of the dataset was randomly reserved as a holdout set to ensure a fair evaluation of the three final models. This prevented overfitting and bias in the training, resulting in a true final estimate of the models' performance on a dataset that is guaranteed to be unseen.

Variable Name	**Description**
MedInc	median income in block group
HouseAge	median house age in block group
AveRoom	average number of rooms per household
AveBedrms	average number of bedrooms per household
Population	Population block group population
AveOccup	average number of household members
Latitude	block group latitude
Longitude	block group longitude
MedHouseVal	median house value (in $100,000)

Fig. 1. Dataset Variable Descriptions

2) Metrics: To evaluate the regression models, the Mean Squared Error (MSE) and the coefficient of determination (R-squared) were used due to their popularity in regression evaluation.

MSE measures how close a regression line is to the data points [5]. The goal is to minimize MSE. It is calculated as:

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

(n represents the sample size, y represents the actual values, and $\hat{y}$ represents the predicted values)

R-squared is the proportion of variation in the response variable that is explained by the regression model [6]. Its equation relies on the Residual Sum of Squares (SSR) and Total Sum of Squares (SST). It is defined as:

$$R^2 = \frac{SSR}{SST}$$

$$\text{SSR} = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

$$\text{SST} = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

(n represents the sample size, y represents the actual values, and $\hat{y}$ represents the predicted values)

It assesses the "fit" of a regression model. R-squared = 0 means there is no fit, while R-squared = 1 means there is a perfect fit. A large SSR is usually preferred.

B. Non-AI Process

1) Exploratory Data Analysis: Exploratory data analysis was conducted to gain a better understanding of scikit-learn's California housing dataset. Comprehensive summary statistics were generated for each variable, which included the count, mean, standard deviation, minimum value, 25th percentile, 50th percentile, 75th percentile, and maximum value. This allowed us to identify patterns and uncover any anomalies. Histograms for each variable were plotted to visualize the distribution of data in Figure 2.

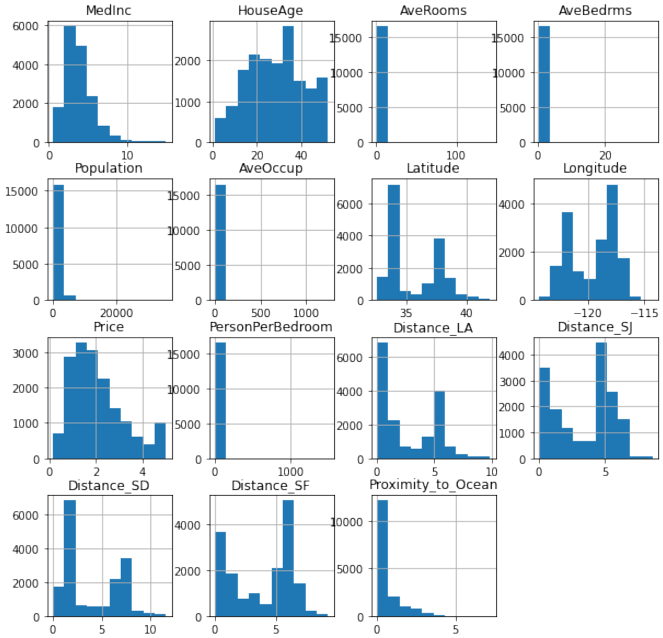

Fig. 2. Distribution of Variables

2) Data Preprocessing: Based on the insights from the exploratory data analysis, data preprocessing measures were undertaken to enhance the quality of the data and improve the performance and reliability of the machine learning models. From the summary statistics, the maximum values of the variables AveRooms, AveBedrms, and AveOccup were significantly higher than the values of the 75th percentile. Therefore, the observations above a certain threshold for AveRooms and AveOccup were removed. This process not only rectified the outliers for AveRooms, but also eliminated the need for further removal of outliers associated with AveBedrms.

Feature engineering was used to improve the performance of the model. Evaluating the correlations between the variables and applying context, several new features were created. Firstly, for each of the major cities: San Diego, Los Angeles, San Jose, and San Francisco, the distances between the cities and houses were computed and stored in new variables. This was motivated by the impact that location has on housing price, as seen in Figure 3. The magnitude of these distances was stored in the new variable for each respective city. Another feature added computes the proximity of the houses to the coastline because locations closer to the water are associated with higher house prices. Lastly, a variable which computes people per bedroom was engineered. All the new features added can be seen in Figure 4. The pairwise correlation between all variables, including the new ones, can be seen in Figure 5. From the correlation analysis, it is revealed that MedInc has the highest correlation with Price, indicating that it is an important predictor.

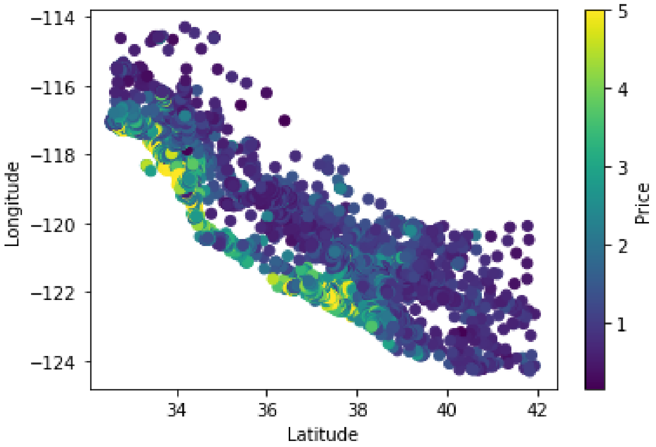

Fig. 3. Latitude vs. Longitude by Price

Variable Name	Description
distance_LA	distance to Los Angeles
distance_SJ	distance to San Jose
distance_SD	distance to San Diego
distance_SF	distance to San Francisco
Proximity_to_Ocean	distance to coastline
PersonPerBedroom	average number of people per bedroom

Fig. 4. Features Added

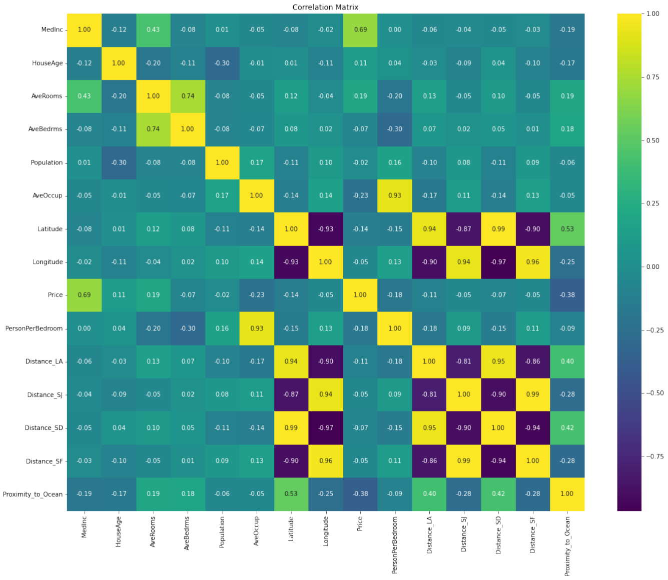

Fig. 5. Correlation Heatmap

3) Models and Hyper-Parameter Tuning: In order to determine the best non-AI model, different models and parameter values were tested. Models that were investigated include Linear, LASSO, Ridge, Decision Tree, Random Forest, and Extreme Gradient Boosted regression:

Linear Regression was evaluated first. For this model, an h is chosen from the hypothesis space $\mathcal{H}$, where

$$h_w(x) = w^T x + b, \quad \text{with } w = [w_1, ...w_n]^T$$

Using this notation, each w_i is called a parameter or a weight and b is the bias or intercept term. In order to determine the values of the parameters, w, an error function is used which measures the difference between predictions and true values. The goal is to minimize this error function. The Linear Regression model in the scikit-learn package uses Ordinary Least Squares (OLS) to compute the error function. This means that w is chosen by minimizing:

$$J(w) = \frac{1}{2} \sum_{i=1}^{n} (h_w(x_i) - y_i)^2$$

Ridge and LASSO regression apply the idea of regularization to penalize the model complexity and deflate large parameter values of w. With $J(w)$ as the training loss, $R(w)$ as the regularization function, and the regularization parameter $\lambda \geq 0$, $L(w)$ is computed as:

$$L(w) = J(w) + \lambda R(w)$$

This equation is computed as to achieve a balance between data fitting and model complexity. This research attempted both L1-norm and L2-norm regularization methods. L2-Norm, also known as Ridge, pushes the parameters towards 0 by adding the penalizing term: $\frac{\lambda}{2} \sum_{j=1}^{n} w_j^2$. Comparatively, L1-norm, known as Least Absolute Shrinkage and Selection Operator (LASSO), achieves regularization with the term: $\lambda \sum_{j=1}^{n} |w_j|$ [8]. To determine the value of λ, cross-validation is used. The difference between these two approaches is that while L2-Norm approaches weight values of 0, L1-Norm is able to achieve exact weights of 0. Therefore, L1-Norm can behave as a method for feature selection by setting weights to 0. To visualize this difference, Figure 6 shows how the intersection of weight $w*$ can be exactly 0 with L1-norm [8].

The decision tree is a flowchart-like tree structure, where each internal node represents a feature, each branch embodies the decision rules, and each leaf node signifies the algorithm's outcome. Throughout training, the decision tree algorithm identifies the optimal attributes and values to split on. Predictions are generated by simulating which leaf node would be reached given an input vector and the values of its attributes [9].

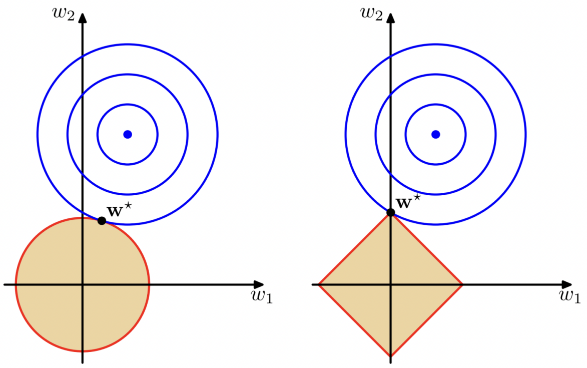

Fig. 6. Regularization: L2-Norm vs. L1-Norm

Model Name	MSE	R-squared
Linear Regression	0.445	0.652
Ridge Regression	0.445	0.652
LASSO Regression	0.620	0.515
Decision Tree Regressor	0.405	0.683
Random Forest Regressor	0.236	0.815
XGBoost Regression	0.203	0.841
XGBoost Regression (Parameter Fine-Tuning)	0.188	0.853

Fig. 7. Metrics for Non-AI Models

Random forest is an ensemble technique that utilizes multiple decision trees. It uses bootstrap aggregation, also known as bagging. Each tree in the forest is trained on a random subset of the data features and the final prediction is obtained by aggregating the predictions of individual trees. It employs averaging as a mechanism to enhance predictive accuracy and mitigate overfitting [10].

Extreme Gradient Boosting, known as XGBoost, is an ensemble technique that utilizes multiple decision trees and uses a method called boosting. Boosting combines weak learners sequentially, with subsequent trees aiming to correct errors made by the previous ones [11].

4) Model Evaluations: To assess the performance of the non-AI models, MSE and R-squared metrics were used. The metrics computed for each model can be seen in Figure 7. In addition, plots were created showcasing the true values alongside the predicted values for the initial 100 observations. These visualizations were helpful in depicting the disparities between the model predictions and actual values, offering valuable insights into the overall performance of the models. For the best model, the XGBoost regressor was selected based on its performance.

5) Conclusion: With XGBoost regressor as the best non-AI model, hyper-parameter tuning was applied to this model. Selecting the optimal values for the hyper-parameters: max_depth, n_estimators, and learning_rate was a key aspect of improving performance. To determine these values, a

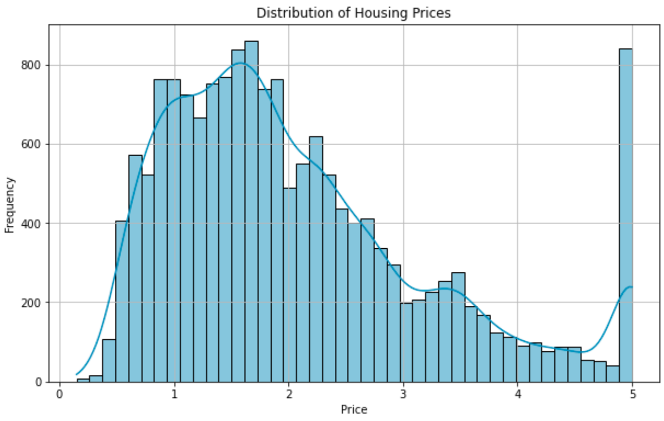

Fig. 8. Model B: Distribution of Price

cross-validated grid-search was performed, where all possible combinations of the hyper-parameter values are tested in order to determine the best results. Using these values, the final model selected had max_depth of 6, n_estimators of 700, and learning_rate of 0.1. This was the final model selected to evaluate the performance of human-created models against AI models. It had an MSE of 0.199 and an R-squared value of 0.848 when evaluated on the holdout set.

C. AI Process

GPT-4 was used to create two regression models to train on the California housing dataset. Model A is the fine-tuned model created by prompting a fine-tuned version of ChatGPT and following its instructions and outputted code. To fine-tune this model, it was given lecture slides on regression modelling from CS4442 as input [7]. Model B refers to the model created by prompting the original version of ChatGPT, and following its instructions and outputted code. The AI process involved 6 base prompts for both models. The following is a summary of the conversation inputs and outputs with ChatGPT for each of the models, A and B:

Prompt 1: ‘I have supplied a CSV file containing data about California housing prices per neighbourhood. The end goal is to create a regression model in Python. The target variable is ‘Price’. Please perform exploratory data analysis. When done, please ask what to do next.’

Model A: Model checks for missing values and data types and analyzes a statistical summary of the dataset.

Model B: Model checks for missing values, analyzes descriptive statistics, looks at the distribution of the target variable ‘Price’ as seen in Figure 8, generates a pair plot to understand relationships between variables as seen in Figure 9, and explores a correlation heatmap as seen in Figure 10.

Prompt 2: ‘Please perform feature engineering. When done, please ask what to do next.’

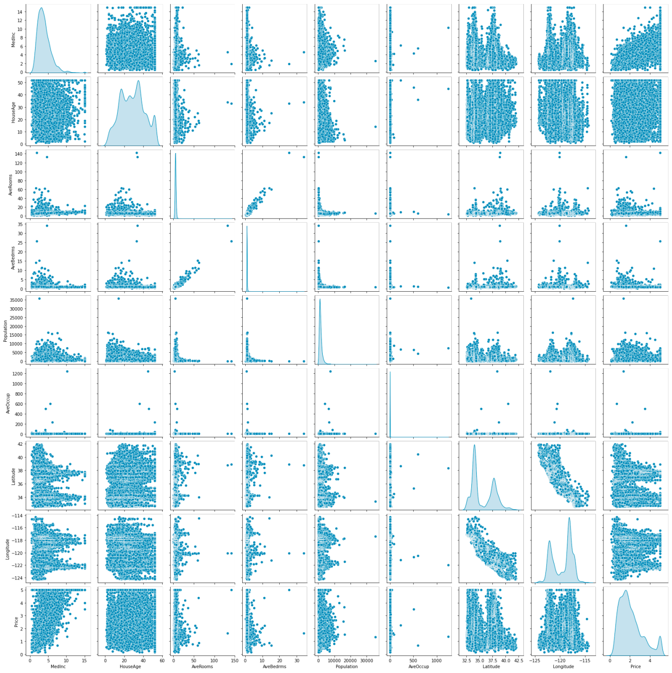

Fig. 9. Model B: Pair Plot

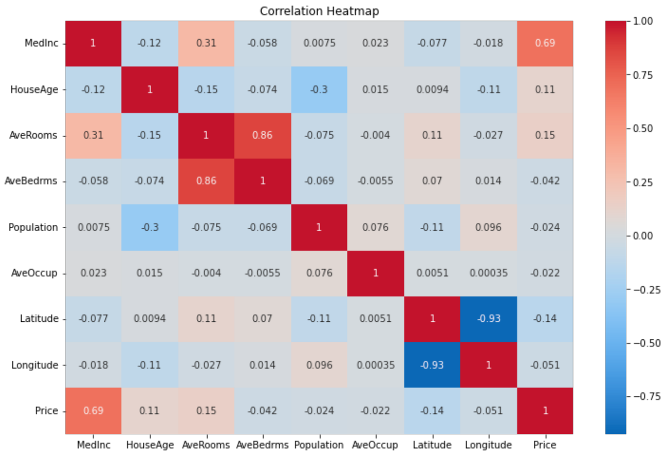

Fig. 10. Model B: Correlation Heatmap

Model A: Model removes unnecessary columns and performs outlier detection and handling using IQR for columns: AveBedrms and AveOccup. It proceeds with feature transformation and creation of new features as well as scaling. The following are the new features created: RoomsPerHousehold, BedroomProportion, PopulationPerHousehld.

Model B: Model drops the unnecessary columns and it creates the new features: RoomsPerHousehold, BedroomProportion, PopulationPerHousehld.

Prompt 3: ‘Please perform data preprocessing. When done, please ask what to do next.’

Model A: Model rechecks for missing values, then splits the dataset into train and test sets. 80% of the data is allocated for training and 20% is for testing purposes.

Model B: Model handles outliers by imposing limits on values at the 1st and 99th percentiles. It then splits the dataset into train and test sets. 80% of the data is allocated for training and 20% is for testing purposes. It also scales the features and checks for multicollinearity.

Prompt 4: 'Please perform model selection and identify the most promising model. When done, please ask what to do next.'

Model A: Model trains and evaluates the following models using MSE:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Elastic Net Regression
- Random Forest Regressor
- Gradient Boosting Regressor

Model B: Model trains and evaluates the following models using root mean squared error (RMSE):

- Linear Regression
- Lasso Regression
- Elastic Net Regression
- Random Forest Regressor

Prompt 5: 'With the selected model, please perform any hyperparameter tuning. When done, please ask what to do next.'

Model A: Model selected the Gradient Boosting Regressor model and performed a grid search cross-validation, using scikit-learn's GridSearchCV, to identify optimal values for the following hyper-parameters:

- n_estimators: [50, 100, 200]
- learning_rate: [0.01, 0.1, 0.2]
- max_depth: [3, 4, 5]

5-fold cross-validation for each of the 27 combinations found that the best values for the hyper-parameters are:

- n_estimators: 200
- learning_rate: 0.1
- max_depth: 5

Model B: Model selected the Random Forest Regressor model and performed a grid search cross-validation, using scikit-learn's GridSearchCV, to identify optimal values for the following hyper-parameters:

- n_estimators: [50, 100, 150],
- max_depth: [None, 10, 20, 30]
- min_samples_split: [2, 4, 6]
- min_samples_leafs: [1, 2, 4]

3-fold cross-validation for each of the 108 combinations found that the best values for the hyper-parameters are:

- n_estimators: 150,
- max_depth: 20
- min_samples_split: 4
- min_samples_leafs: 2

Model	MSE	R-squared
Non-AI Model	0.199	0.848
Model A (fine-tuned)	0.252	0.808
Model B	1.454	-0.110

Fig. 11. Ranked Model Metrics

Prompt 6: 'Please create a feature that accepts the model and a dataset with the same structure as the original one provided and have the output be the accuracy of predictions on the new dataset.'

Model A and Model B both created a function that accepts a dataset and a model, performs the feature engineering and preprocessing, and evaluates the model on a new dataset using MSE and R-squared.

Each of these models trained by ChatGPT was retrained on the entire dataset, using the optimal hyperparameters, and was evaluated on the unseen holdout set. The results are as follows:

Model A:
MSE: 0.252
R-squared: 0.808

Model B:
MSE: 1.454
R-squared: –0.110

V. Results

For the non-AI process, several models were built and evaluated based on the MSE and R-squared scores. The process of finding the best human-created model involved the application of context, feature engineering, and hyper-parameter tuning. Due to the number of models considered, the non-AI process was lengthy. However, the ability to apply context to the problem was crucial in achieving good model results. Upon analysis, the XGBoost model was determined to have the best performance. The XGBoost model selected produced an MSE score of 0.199 and an R-squared of 0.848 on the holdout dataset.

In the AI process, comparative analysis revealed contrasting results between the two AI models when evaluated on the holdout dataset. The standard ChatGPT model demonstrated overfitting or significant error, evident by its negative R-squared value of -0.110 and high MSE of 1.454. In contrast, the fine-tuned ChatGPT model excelled, with an R-squared value of 0.808 and a considerably lower MSE of 0.252 on the holdout set. The results of all three models are shown in Figure 11.

Notably, ChatGPT's aptitude in constructing accurate regression models was much higher when using the fine-tuned ChatGPT model. The fine-tuned model, enriched with contextual information, significantly outperformed the standard model. This disparity highlights ChatGPT's potential when supplemented with targeted knowledge. ChatGPT's computational power is particularly advantageous in model creation, allowing for more nuanced and contextually relevant outputs.

However, this study highlights an area of the tool that requires further examination: the effectiveness of ChatGPT's model-building when constrained by a rigid prompt structure. A user's ability to continuously add context-rich statements can enhance the model's performance, implying that having domain expertise is crucial in guiding successful ChatGPT outputs. This limitation is a new discovery in the realm of building ML models using AI tools such as ChatGPT.

The key takeaway is the vast improvement in model accuracy with context-specific fine-tuning. The significantly better MSE and R-squared values of the fine-tuned model underscore the impact of fine-tuning and contextual enrichment on model performance in ChatGPT. The stark contrast in performance metrics between the two models is an exciting case study for the necessity of domain knowledge and the potential limitations of ChatGPT's model creation capabilities when domain knowledge is lacking. This research suggests how users can leverage ChatGPT's capabilities more effectively by providing rich, relevant context and domain expertise. Further research into ChatGPT's model creation abilities should focus on relaxing the prompt structure and allowing a domain expert to have a more natural dialogue with a fine-tuned ChatGPT. Such research will continue to quantify the effect that improved context has on ChatGPT's output.

VI. Conclusions and Future Work

Among the various models tested, the human-created model outperformed both AI models in terms of the MSE and R-squared. This outcome highlights a critical aspect of model creation: the ability of humans to critically think about variables, interpret metrics and results, and provide contextual information. These human capabilities proved to be highly beneficial and were key factors in the superior performance of the non-AI model. The findings of this study illustrate the limitations of AI models. Specifically, ChatGPT models when constrained by rigid prompt structures. Despite their efficiency in terms of time, AI models, like those generated by ChatGPT, require significant human input to reach their full potential.

The results of this study pave the way for future research in several key areas. Future research could explore the development of hybrid models that integrate AI capabilities with human expertise. Such models could harness the computational power of AI while benefiting from the critical and contextual insights provided by human experts. Additionally, further studies could aim to enhance the ability of AI tools like ChatGPT to understand and incorporate contextual information. This advancement would reduce the dependency on human input for optimal performance. Expanding the scope of research to include a broader range of models, diverse datasets, or other AI tools could also offer more comprehensive insights into the comparative performance of human-created models and ChatGPT models.

This study highlights several important lessons in the field of data science. One of the key lessons is the irreplaceable value of human expertise. Attributes including critical thinking and contextual understanding, inherent in humans, are crucial for the success of machine learning models. This study reinforces the idea that AI and machine learning tools should be viewed as aids to human expertise, not replacements. Achieving the best results in data modeling requires a balanced approach that utilizes both AI's computational efficiency and the understanding of context that humans provide.

References

[1] Nascimento, N., Alencar, P., Lucena, C., & Cowan, D. (2018). Toward human-in-the-loop collaboration between software engineers and Machine Learning Algorithms. 2018 IEEE International Conference on Big Data (Big Data). https://doi.org/10.1109/bigdata.2018.8622107

[2] Carney, M., Webster, B., Alvarado, I., Phillips, K., Howell, N., Griffith, J., Jongejan, J., Pitaru, A., & Chen, A. (2020). Teachable machine: Approachable web-based tool for Exploring Machine Learning Classification. Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems. https://doi.org/10.1145/3334480.3382839

[3] Walsh, M., Ross, D. A., Worrel, C., & Gomez, A. (2023). DEMONSTRATING THE PRACTICAL UTILITY AND LIMITATIONS OF CHATGPT THROUGH CASE STUDIES

[4] scikit-learn. 7.2. real world datasets: California Housing Dataset. https://scikit-learn.org/stable/datasets/real_world.html#california-housing-dataset

[5] Simplilearn. (2023, October 19). Statistics tutorial, Lesson 13: Mean Squared Error: Overview, Examples, Concepts and More. Simplilearn.com. https://www.simplilearn.com/tutorials/statistics-tutorial

[6] Abraham, B., & Ledolter, J. (2006). Introduction to regression modeling. Thomson Brooks/Cole.

[7] Wang, B. Lecture 3: Supervised Learning: Linear Regression. Artificial Intelligence II (CS4442 & CS9542). London; University of Western Ontario, Department of Computer Science.

[8] Wang, B. Lecture 4: Overfitting, Cross-Validation, and Regularization. Artificial Intelligence II (CS4442 & CS9542). London; University of Western Ontario, Department of Computer Science.

[9] GeeksforGeeks. (2023, August 20). Decision tree. GeeksforGeeks. https://www.geeksforgeeks.org/decision-tree/

[10] GeeksforGeeks. (2023b, December 6). Random Forest regression in python. GeeksforGeeks. https://www.geeksforgeeks.org/random-forest-regression-in-python/

[11] Dhingra, C. (2020, December 28). A visual guide to gradient boosted trees. Medium. https://towardsdatascience.com/a-visual-guide-to-gradient-boosted-trees-8d9ed578b33