Growing Beyond Genes: Predicting Adolescent Height with XGBoost

Mapping Growth Trajectories Through Generational Data on a Distant World

Vani Singh, Ajay Kallepalli, Thi Nguyen

March 3, 2025

Contents

List of Tables

List of Figures

0 Executive Summary

This study investigated growth patterns and height prediction in a two-generation dataset of a species with physiological similarities to humans. Using XGBoost, researchers developed a predictive model for adolescent height based on early childhood measurements, parental characteristics, and growth trajectories. The model achieved exceptional accuracy with a root mean squared error of 2.57 cm and explained 96% of the variance in height predictions. Key findings include a strong positive correlation between parent and child heights, with increased variability at higher height ranges suggesting complex genetic-environmental interactions. Age group proved to be a powerful predictor of height (correlation coefficient of 0.97), with early growth velocity, height measurements at ages 6-12, and parental height characteristics serving as the most influential predictors. These findings advance understanding of developmental biology while demonstrating the effectiveness of machine learning approaches for modeling physiological traits.

1 Introduction

The study of growth patterns in biological organisms is critical to understand the developmental processes, genetic inheritance, and environmental influences on physiology. On a distant planet, researchers have conducted a longitudinal study to examine the growth trajectories of a species that exhibits remarkable physiological similarities to humans. This study spans two generations and captures detailed anthropometric measurements, allowing for an in-depth analysis of height and weight development over time.

In the first phase of the study, designated as *Gen1*, researchers recorded standing height measurements at multiple time points from birth until approximately 20 years of age. In the second phase, the progeny of *Gen1*, referred to as *Gen2*, were monitored from birth to around age 18, with both height and weight systematically recorded. This inter-generational dataset provides an opportunity to analyze growth patterns, assess hereditary influences, and develop predictive models for adolescent height based on early-life characteristics.

This study employs extreme gradient boosting (XGBoost) as the primary analytical approach. XGBoost is a powerful machine learning algorithm particularly well-suited for this analysis due to its ability to handle complex nonlinear relationships and interactions between variables. As a tree-based ensemble method, XGBoost excels at capturing the nuanced relationships between early childhood growth metrics, parental characteristics, and adolescent height outcomes.

By focusing exclusively on XGBoost methodology, this research establishes a robust framework for predicting adolescent height from childhood measurements and parental data. The findings improve our understanding of how similar organisms grow and show that tree-based models work well for studying complex body processes. Furthermore, the analysis of feature importance illuminates key predictors of height development, potentially forming future studies of growth trajectories across diverse populations.

2 Exploratory Data Analysis

2.1 Summary Statistics

In the exploratory data analysis (EDA) phase, the key characteristics of the datasets for both Gen1 and Gen2 across training and testing sets were summarized. The **Gen1 Train Summary** (Appendix A) indicates that the `age` variable ranges from 0.1 to 20 years, with a mean age of approximately 10.35 years. The `SHgt_cm` (height) variable ranges from 50.63 cm to 197.08 cm, with a mean height of 135.44 cm. Notably, there are missing values in the `SHgt_cm` column, with 310 entries lacking height data. The **Gen2 Train Summary** (Appendix B) shows similar characteristics but includes additional variables such as `Wgt_kg` (weight) and `AgeGr` (age group). The age variable ranges from 0.1 to 18 years, and the `SHgt_cm` ranges from 49.90 cm to 196.14 cm. Missing data is more prevalent in the `SHgt_cm` (512 missing values) and `Wgt_kg` (2045 missing values) columns.

The Gen1 Test Summary (Appendix A) mirrors the **Gen1 Train dataset**, with slight variations in the mean and standard deviation of the `SHgt_cm` and age variables. Missing values in the `SHgt_cm` column (216 missing values) are also observed. The **Gen2 Test Summary** (Appendix B) reflects the Gen2 Train data but with fewer entries. Here, the `SHgt_cm` and `Wgt_kg` columns have significant missing data, with 132 and 823 missing values, respectively. These summaries highlight important trends, such as the distribution of physical characteristics across different groups and the extent of missing data, which will be critical when considering strategies for imputation or exclusion during data preprocessing.

2.2 Distribution of Target Variable

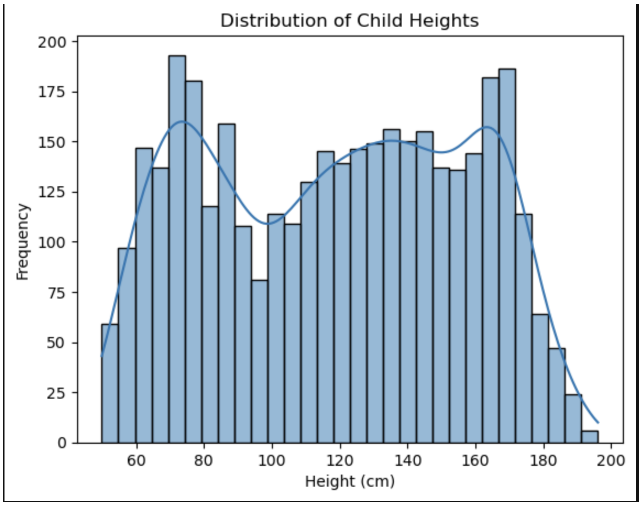


Figure 1: Distribution of Analysis Target Variable

The distribution of child heights depicted in Figure 1 presents a complex pattern characterized by three distinct peaks occurring at different height ranges. The data spans from approximately 50 cm to 200 cm, with frequency counts reaching up to about 200 children at the highest concentrations. The first notable peak appears around 70-80 cm, followed by a relatively consistent middle section between 110-140 cm, and a third prominent peak around 160-170 cm. This pattern strongly suggests that the data encompasses children from various age groups or developmental stages rather than representing a single population with similar characteristics. It is apparent that the data is not normally distributed, as evidenced by the multiple peaks rather than a single bell-shaped curve. Consequently, researchers

cannot appropriately apply statistical models that assume normality without first implementing transformations to the target variable. Proceeding with analysis methods that incorrectly presume a normal distribution would likely result in misleading parameter estimates, unreliable confidence intervals, and compromised hypothesis testing. The distribution captures the complete range of heights typically observed from early childhood through adolescence, with a natural tapering at both the lower and upper extremes.

2.3 Relationship between Parent and Child Heights

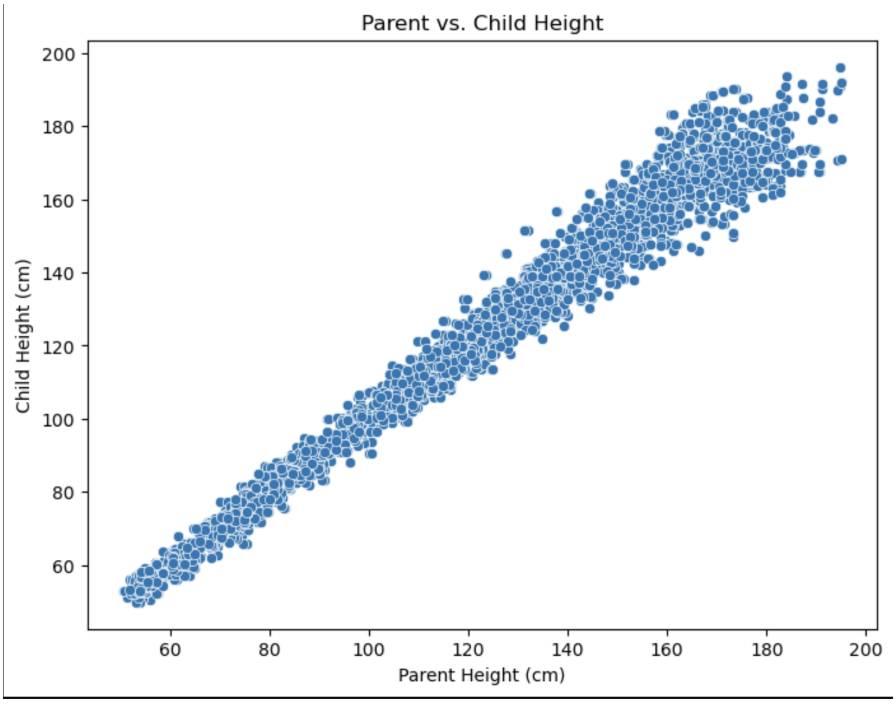


Figure 2: Child vs. Parent Heights

Figure 2 illustrates the relationship between parent and child heights through a scatter plot, with parent height (in centimeters) on the x-axis and child height (in centimeters) on the y-axis. The data reveals a strong positive linear correlation between the two variables, with data points forming a clear diagonal pattern from the lower left to the upper right of the graph. This pronounced linear relationship suggests that taller parents tend to have taller children, while shorter parents tend to have shorter children.

Notably, the correlation pattern displays an interesting characteristic: as both parent and child heights increase, the data points spread out more widely, creating a cone-like shape with greater variability at the upper height ranges. This widening pattern implies that while genetic inheritance remains significant across all height ranges, additional factors may play a more pronounced role in determining the heights of taller children. The increased dispersion at higher heights could suggest that environmental factors, nutrition, or complex genetic interactions have greater influence in families where parents are already tall.

This visualization complements the earlier distribution of child heights by providing insight into one of the key factors influencing the variability observed in the child height data, while also revealing the complexity of height inheritance patterns across different height ranges.

2.4 Correlation Between Variables

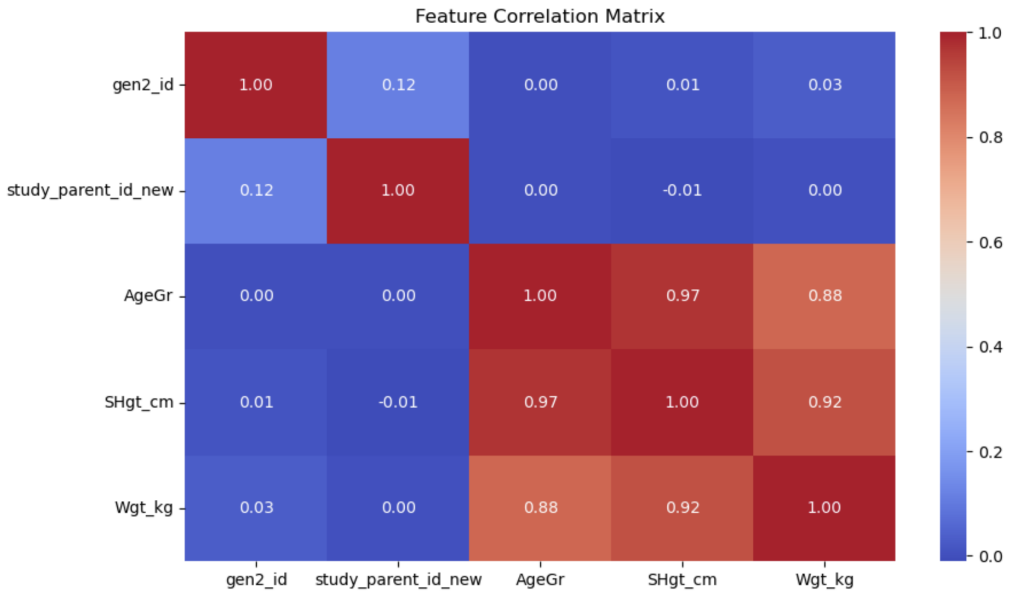


Figure 3: Gen2 Dataset Correlation Matrix

Figure 3 presents a feature correlation matrix that visually represents the relationships between five variables in the dataset. The matrix reveals distinct patterns of correlation, with color intensity and numerical values indicating correlation strength.

The most striking observation is the high positive correlations among three variables: AgeGr (presumably age group), `SHgt_cm` (seemingly standing height in centimeters), and `Wgt_kg` (weight in kilograms). These three variables form a strongly correlated cluster with correlation coefficients ranging from 0.88 to 0.97. Particularly notable is the extremely high correlation (0.97) between `AgeGr` and `SHgt_cm`, suggesting that age group is a very strong predictor of height. Similarly, weight (`Wgt_kg`) shows strong correlations with both height (0.92) and age group (0.88).

3 Feature Engineering

The `create_features` function (Appendix C) was implemented to generate key predictors for height modeling while ensuring robust handling of missing values. First, the function merges the child dataset with parent height data using `study_parent_id_new` and `AgeGr` to align parental information with corresponding child records. It then derives gender-related features, including `is_male` to indicate if the child is male, `parent_is_mother` to distinguish maternal relationships, and `parent_child_same_gender` to capture whether the parent and child share the same gender.

Growth-related features are extracted by identifying height measurements at ages 6, 9, and 12, and the computed height differences between these ages (`height_diff_9_6` and `height_diff_12_9`). The function calculates the the early growth velocity using height changes before age 9, while `peak_growth_rate` and `age_peak_growth` are determined from height changes between ages 10 and 18. Furthermore, `parent_height_18` is recorded as the parent's height at maturity, and the difference and ratio between the childâĂŹs height at age 9 and the parentâĂŹs height at 18 (`parent_child_height_diff` and `parent_child_height_ratio`) are included to assess the influence of family height.

The function also computes BMI (`bmi_at_9`) when weight data is available, ensuring safe division to handle missing values. Growth dynamics is further captured through `growth_rate_change`, defined as the difference between the peak growth rate and the early growth velocity. The function iterates over the data of each child, generating a row for each target age (10âĂŞ18) that includes these engineered characteristics alongside demographic and parental attributes. These transformations provide a structured and informative set of features for predictive modeling.

4 Model Training

The model training process consisted of multiple steps, including data preparation, feature selection, hyperparameter tuning, and evaluation. The goal was to develop a XGBoost model for estimating height using historical growth patterns and demographic attributes.

4.1 Data Processing

To ensure that the training data contained the necessary target variable, the generated feature set was merged with the original dataset containing actual height values. This merge was performed based on the unique identifiers `gen2_id` and `AgeGr`, which represent the individual and their respective age group. Observations with missing height values (`SHgt_cm`) were removed to maintain data integrity and prevent biases during training. (Refer to Appendix D)

4.2 Additional Feature Engineering and Selection

A comprehensive set of features was selected to capture key aspects of growth patterns and biological influences on height using the function `train_data`. These included:

- **Age-based growth indicators:** Age group (`AgeGr`), height measurements at specific ages (`last_height`, `height_at_9`), and growth rate indicators (`growth_velocity`, `height_diff_9_6`, `height_diff_12_9`).
- **Peak growth characteristics:** The maximum growth rate observed (`peak_growth_rate`) and the corresponding age (`age_peak_growth`).
- **Demographic attributes:** Gender (`is_male`), whether the parent in the dataset is the mother (`parent_is_mother`), and whether the child shares the same gender as the parent (`parent_child_same_gender`).
- **Body mass index (BMI):** If available, BMI at age 9 (`bmi_at_9`) was included as an additional predictor.

To handle missing values within these features, an Iterative Imputer based on the Multiple Imputation by Chained Equations (MICE) method was applied. This approach leverages predictive modeling to impute missing values iteratively, improving data quality. The imputed features were then standardized using z-score normalization to ensure that all variables contributed proportionally to the model.

4.3 Train-Validation Split

Since the dataset contained multiple height observations for the same individuals across different ages, a standard random train-test split could have led to data leakage, where information from the same individual appears in both training and validation sets. To prevent this, a group-aware splitting strategy was implemented using `GroupShuffleSplit`. This method ensured that all height records corresponding to the same individual (`gen2_id`) were assigned exclusively to either the training or validation set, preserving the independence of validation samples.

The data was split into 80% training and 20% validation to allow sufficient data for both model learning and evaluation.

4.4 Hyperparameter Tuning

To optimize the XGBoost model, a grid search cross-validation approach was applied. The hyperparameter search space included:

Parameter	Description	Tested Values
max_depth	Controls the depth of each tree	3, 4
learning_rate	Determines the step size at each boosting iteration	0.04, 0.05
min_child_weight	Regulates model complexity to prevent overfitting	3, 4
subsample	Fraction of training data used per boosting round	0.65, 0.7
colsample_bytree	Fraction of features used per tree	0.65, 0.7
gamma	Minimum loss reduction required for further tree splitting	0.3, 0.5
n_estimators	Number of boosting rounds	200, 250
scale_pos_weight	Adjusts weight balancing	1.8, 2
reg_alpha and reg_lambda	L1 and L2 regularization terms to control model complexity	0.15-0.2 and 0.25-0.3

Table 1: Values used for Hyperparameter Tuning

A 4-fold cross-validation approach was used, where the training data was repeatedly split into different subsets to ensure that the model generalized well to unseen data. The evaluation metric for tuning was negative root mean squared error (RMSE), which prioritizes minimizing prediction errors. (Refer to Appendix E)

4.5 Resulting Model

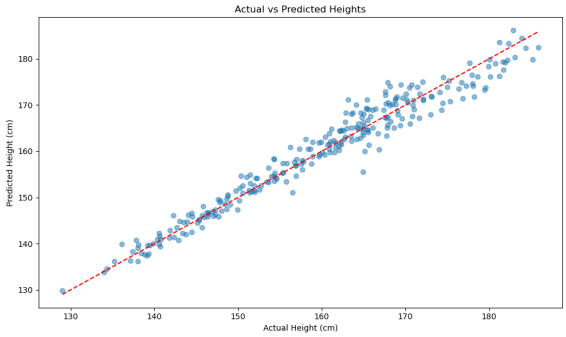
The hyperparameter tuning process involved evaluating 1,024 different combinations across 4-fold cross-validation, resulting in a total of 4,096 model fits. The best-performing parameter set optimized the trade-off between model complexity and generalization. The selected hyperparameters suggest a moderately deep model (max_depth = 4) with a careful balance of regularization (reg_alpha = 0.15, reg_lambda = 0.25) and feature sampling (colsample_bytree = 0.7, subsample = 0.7). The model also benefits from a conservative learning rate (learning_rate = 0.05) and an increased number of boosting rounds (n_estimators = 250) to improve predictive accuracy while avoiding overfitting.

5 Model Evaluation

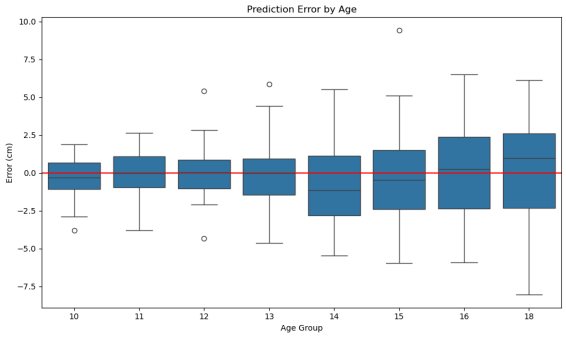
Metric	Value
RMSE	2.57
MAE	1.96
R^2	0.96

Table 2: Evaluation Metrics on Validation Data

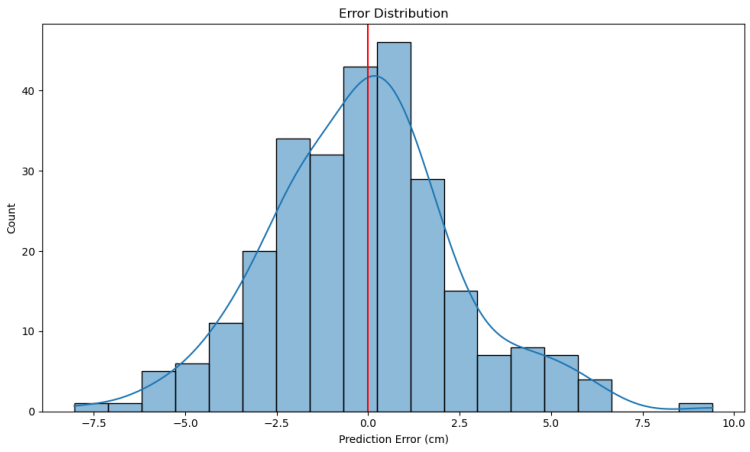
The final trained model demonstrated strong predictive performance on the validation dataset, achieving a root mean squared error (RMSE) of 2.57 cm and a mean absolute error (MAE) of 1.96 cm. These error metrics indicate that the model's height predictions closely align with actual values, with an average absolute deviation of less than 2 cm. Furthermore, the model attained an R^2 score of 0.96, signifying that 96% of the variance in the target variable is explained by the selected features. This high R^2 value, coupled with relatively low error rates, suggests that the model generalizes well to unseen data while effectively capturing key patterns in growth trajectories.



(a) Actual vs. Predicted Height



(b) Prediction Error by Age



(c) Error Distribution

Figure 4: Error Plots

The scatter plot of actual vs. predicted heights (Figure 5a) shows a strong alignment along the diagonal, indicating that the model performs well overall in capturing the relationship between features and height. The points are closely clustered around the red dashed line, reinforcing the high R^2 value of 0.96, which suggests that the model explains most of the variance in the data. However, as actual height

increases, there is slightly more dispersion around the line, indicating that the model's predictions become less precise for taller individuals. Despite this minor variance, the overall performance remains strong, suggesting the model makes accurate predictions with minimal error.

The box plot of prediction error by age group (Figure 5b) shows that the model's accuracy varies slightly across different ages. The median error remains close to zero across all age groups, indicating that the model does not exhibit significant systematic bias. However, as age increases, the spread of the error distribution becomes wider, with larger interquartile ranges and more extreme outliers, suggesting that the model's predictions are less precise for older individuals. This aligns with the earlier observation that prediction dispersion increases with actual height, potentially indicating greater variability in height growth patterns at older ages that the model struggles to fully capture.

The error distribution (Figure 5c) appears approximately normal and centered near zero, as indicated by the overlaid bell curve and the vertical red line at the zero point. This suggests the model is relatively unbiased, with errors distributed fairly symmetrically around zero. Most prediction errors fall within approximately Âś5 cm of the actual values, with the highest frequencies occurring between -2.5 and 2.5 cm. There are some outliers at both extremes, with a few predictions erring by as much as 7.5 to 10 cm. The consistency of the error distribution suggests that the model is capturing the main relationships in the data effectively, though the presence of outliers indicates there may be some cases where additional factors not included in the model are influencing child height.

6 Conclusion

This study successfully developed a robust XGBoost model for predicting adolescent height based on early childhood measurements, parental characteristics, and growth patterns. The model demonstrated excellent predictive accuracy with an RMSE of 2.57 cm, MAE of 1.96 cm, and R^2 score of 0.96. While predictions were generally accurate across all age groups, slight increases in prediction variance were observed for older adolescents and taller individuals. The strong correlation between parent and child heights confirmed the significant role of genetic inheritance, though the increasing dispersion at higher height ranges suggests environmental factors may exert greater influence in families where parents are already tall.

The feature engineering approach effectively captured key aspects of growth dynamics, including growth velocity at different developmental stages and the relationship between early childhood height and parental measurements. These engineered features, combined with comprehensive hyperparameter tuning, created a valuable framework for anthropometric modeling. The findings contribute to the broader understanding of growth patterns and hereditary influences in biological organisms, while suggesting that future research could explore additional features to address the increased variance in older age groups and investigate the complex interactions between genetic and environmental factors influencing growth trajectories.

7 Heredity Analysis: Parental Predictors of a Child's Final Height at Adulthood

To investigate the hereditary influences on final adult height, this study examined which parental growth characteristics are predictive of a child's ultimate height. The analysis involved comprehensive feature engineering, correlation analysis, and predictive modeling.

7.1 Feature Engineering

A robust set of predictive features was derived from the longitudinal height data, emphasizing key parental growth attributes:

- **Height at Key Developmental Stages**: Heights were extracted for each parent at ages 1, 2, 5, 10, 15, and 18 years. Missing data were interpolated to ensure completeness.
- **Final Adult Height**: The last recorded height for each parent, typically around age 18 or older, was used as a critical predictor for offspring height.
- **Peak Growth Velocity**: The maximum yearly height increase was calculated by differentiating successive height measurements, along with the age at which peak growth occurred.
- **Pubertal Growth Magnitude**: The total height gained during puberty (ages 9-15) was computed.
- **Growth Rate in Key Age Intervals**: Growth rates were calculated for the intervals 0-2, 2-5, 5-10, 10-15, and 15-18 years.

7.2 Correlation Analysis

A Pearson correlation analysis was performed to identify associations between parental growth features and the childâĂŹs final height. The following correlations were observed:

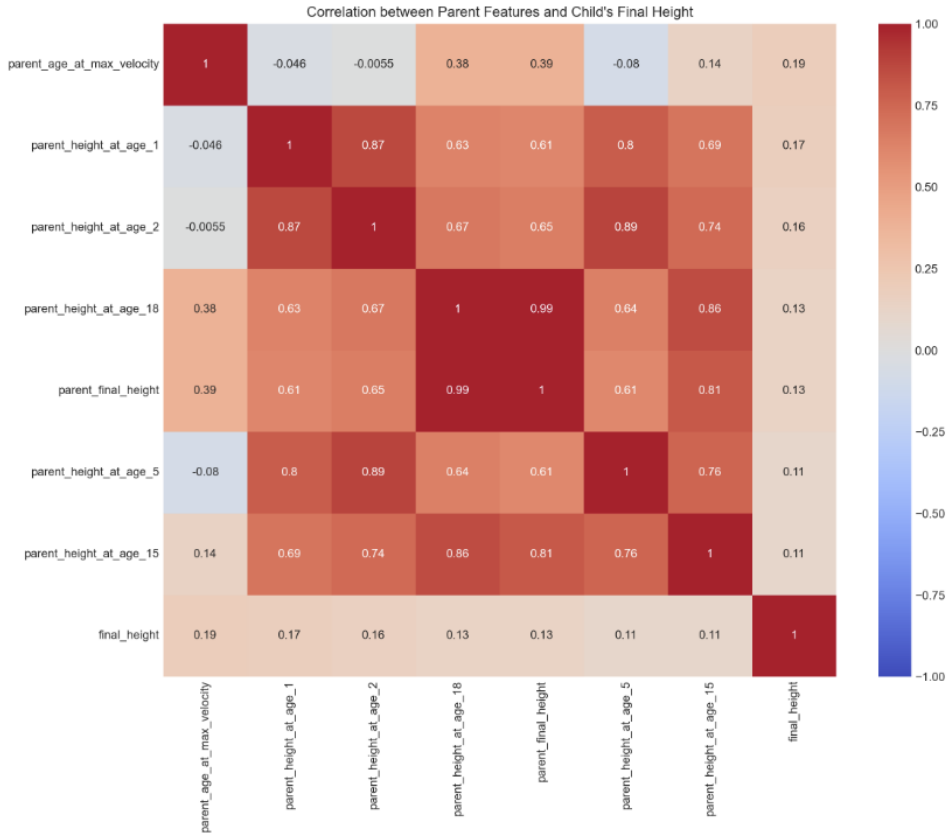


Figure 5: Correlation between Parent Features and Child's Final Height

Figure 5 revealed several significant correlations between parental attributes and a child's growth patterns. Specifically, parental age at peak growth velocity was positively correlated with the child's growth, as indicated by a coefficient of $r = 0.1927$. This suggests that children whose parents reached peak growth at an older age may experience greater growth spurts. Additionally, parental height at age 1 showed a correlation of $r = 0.1740$, indicating that early parental height can influence children's growth outcomes. Similarly, parental height at age 2 had a correlation of $r = 0.1581$, reinforcing the importance of early developmental factors. Moreover, parental height at age 18 and the final parental height were also positively correlated with coefficients of $r = 0.1349$ and $r = 0.1345$, respectively, further supporting the idea that parental final height contributes to the child's ultimate stature. These findings underscore the multifaceted nature of growth, highlighting how various parental characteristics, particularly those observed during key developmental stages, play a crucial role in determining a child's final height.

These findings suggest that a child's final height is moderately influenced by various parental height metrics. Early childhood heights (ages 1 and 2) and final adult height exhibited the strongest associations, underscoring the importance of early development in determining ultimate height. Additionally, the age at which parents reached peak growth velocity appeared to have a moderate influence, suggesting that the timing and tempo of growth in the parents play a role in determining the child's growth pattern.

7.3 Predictive Modeling

Multiple regression-based and ensemble models were trained to assess predictive power:

Model	R^2	MSE
Linear Regression	0.0117	402
Ridge Regression	0.0111	402
Random Forest Regressor	0.1545	344

Table 3: Model Performance Metrics for Final Height at Adulthood Prediction

Table 3 shows that the Random Forest model outperformed other models, though overall predictive power remained modest.

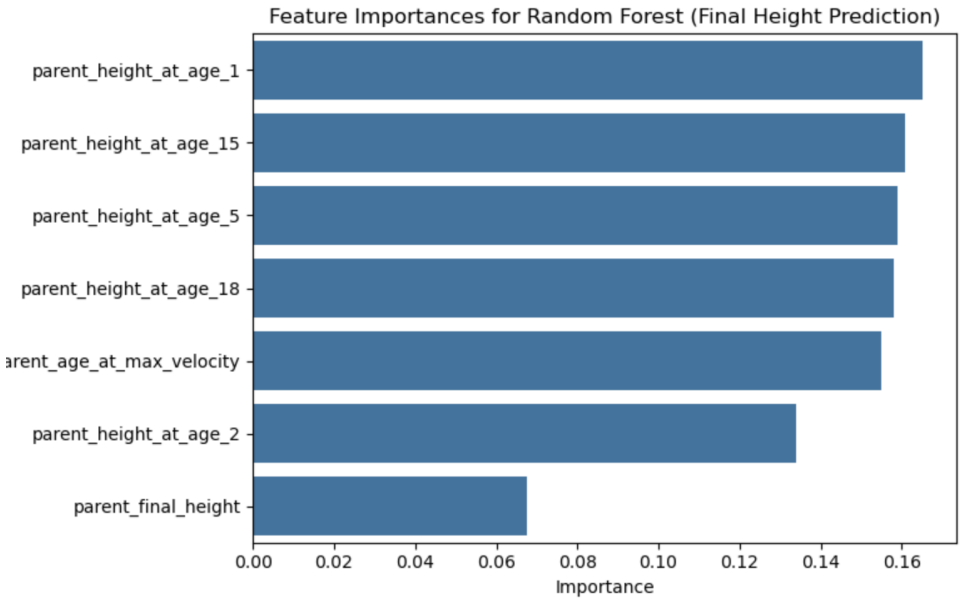


Figure 6: Feature Importances for Random Forest (Final Height)

Figure 6 illustrates the feature importances from a Random Forest model predicting final height. The most influential features are parental heights at various ages, with `parent_height_at_age_1` being the strongest predictor, followed closely by heights at ages 15, 5, and 18. This suggests that a parent's height during early childhood and adolescence significantly impacts final height predictions. Additionally, `parent_age_at_max_velocity`, which represents the age at which the parent experienced peak growth, also plays a key role, indicating that growth timing influences final height.

Interestingly, `parent_final_height` is the least important feature in the model, implying that height at earlier stages already captures most of the predictive power. Overall, the model prioritizes growth trajectory and timing over final parental height when predicting final height outcomes.

7.4 Conclusion

Parental height at different developmental stages plays a modest role in predicting a child's final height. However, predictive models indicate that additional biological, environmental, and genetic factors contribute significantly to height determination.

8 Heredity Analysis: Parental Predictors of a Child's Pubertal Growth Magnitude

The pubertal growth spurt, typically occurring between ages 9 and 15, may also be influenced by hereditary factors. This section explores which parental growth characteristics are most predictive of a child's pubertal growth magnitude.

8.1 Feature Engineering

To analyze hereditary patterns in pubertal growth, the following parental features were engineered:

- **Peak Growth Velocity and Age at Peak Velocity**: Maximum yearly height increase and age at which this peak occurred.
- **Pubertal Growth Magnitude**: Difference in height between ages 9 and 15.
- **Height at Key Developmental Stages**: Heights at ages 1, 2, 5, 10, 15, and 18.
- **Growth Rate Across Age Intervals**: Growth rates for 0-2, 2-5, 5-10, 10-15, and 15-18 years.

8.2 Correlation Analysis

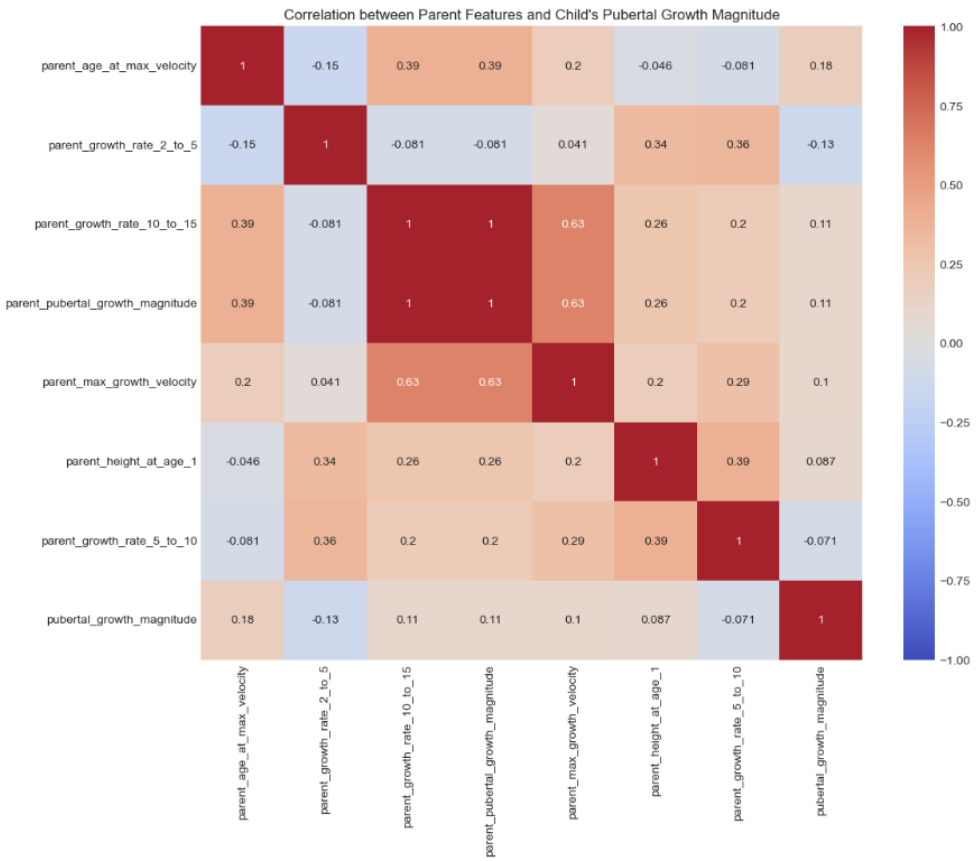


Figure 7: Correlation between Parent Features and Child's Pubertal Growth Magnitude

The analysis identified several parental features that are correlated with a child's pubertal growth magnitude. Looking at Figure 5, parental age at peak growth velocity exhibited a positive correlation

of $r = 0.1784$, suggesting that parents who reach their peak growth later may have children who experience greater growth spurts. Additionally, parental pubertal growth magnitude showed a correlation of $r = 0.1078$, indicating that the extent of parental growth during puberty also relates to the child's growth during the same phase. Furthermore, parental maximum growth velocity was positively associated with the child's growth, with a correlation coefficient of $r = 0.1033$. Conversely, the parental growth rate during early childhood from ages 2 to 5 demonstrated a negative correlation of $r = 0.1289$ which may imply that higher growth rates in this early period could lead to compensatory growth patterns later. Lastly, the growth rate from ages 10 to 15 had a positive correlation of $r = 0.1078$, indicating that parental growth during this key developmental window is also relevant for predicting a child's pubertal growth magnitude. Together, these correlations highlight the complex interplay between parental growth characteristics and the resulting growth patterns observed in their children.

These results suggest that children whose parents experienced later and more pronounced pubertal growth spurts tend to have greater pubertal growth magnitudes themselves. Additionally, the negative correlation with growth rates in early childhood (ages 2-5) may indicate a compensatory growth effect, where slower growth in early years is followed by a more significant growth spurt during puberty.

8.3 Predictive Modeling

Regression-based and ensemble models were developed:

Model	R^2	MSE
Linear Regression	-0.0192	121
Ridge Regression	-0.0198	121
Random Forest Regressor	0.0209	116

Table 4: Model Performance Metrics for Pubertal Growth Prediction

The Random Forest model showed a slight improvement over linear models, but all models exhibited weak predictive power.

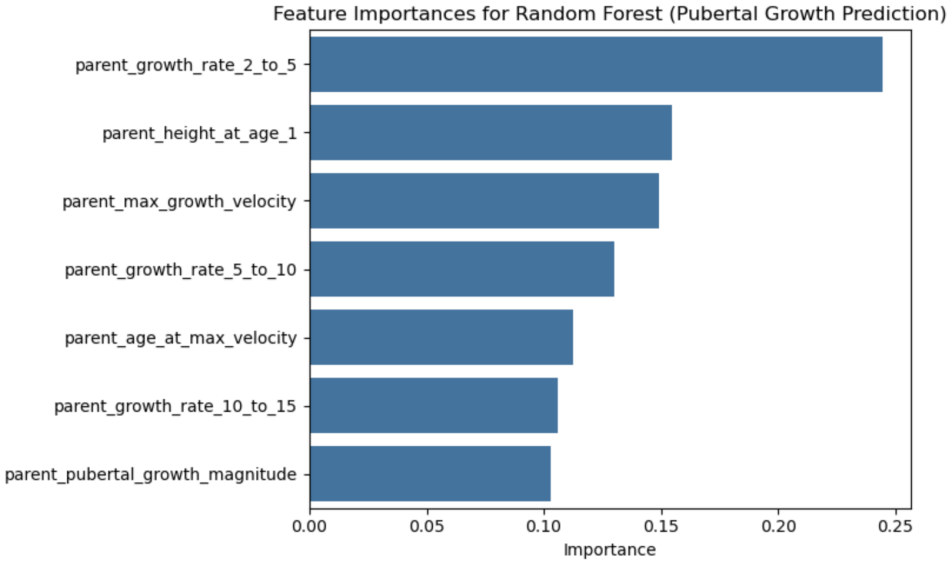


Figure 8: Feature Importances for Random Forest (Pubertal Growth)

Figure 8 presents the feature importances from a Random Forest model predicting pubertal growth. The most influential feature is `parent_growth_rate_2_to_5`, suggesting that early childhood growth rate plays a crucial role in determining pubertal growth outcomes. The second most important feature, `parent_height_at_age_1`, indicates that early height is also a strong predictor. Other significant features include `parent_max_growth_velocity`, which captures the peak rate of growth, and `parent_growth_rate_5_to_10`, reinforcing the importance of pre-pubertal growth trends.

Additionally, `parent_age_at_max_velocity` suggests that the timing of peak growth influences pubertal growth, while `parent_growth_rate_10_to_15` and `parent_pubertal_growth_magnitude` further emphasize the role of sustained growth during adolescence. Overall, the model highlights that early childhood growth patterns and peak growth velocity are key determinants of pubertal growth trajectories.

8.4 Conclusion

Parental growth characteristics, particularly peak growth velocity and pubertal growth magnitude, show moderate associations with a child's pubertal growth. These findings indicate that heredity influences the timing and extent of pubertal growth. However, predictive models reveal that these parental features alone cannot accurately predict a childâĂŹs pubertal growth magnitude, emphasizing the complexity of growth regulation and the importance of considering environmental influences and other biological factors in shaping growth during puberty.

9 Heredity Analysis: Growth Spurt Timing

This section explores which parental growth characteristics if any are most predictive of a child's pubertal growth spurt timing.

9.1 Feature Engineering

To analyze hereditary patterns in pubertal growth, the following parental features were engineered:

- **Parent Peak Growth Rate and Parent Age at Peak Growth Rate**: Maximum yearly height increase and age at which this peak occurred.
- **Parent Child Height Ratio**: Ratio of parent height to child height.
- **Parent Height at 18**: Final recorded height of parent at 18.
- **Parent Child height diff**: Difference between parent and child's height.
- **Age at Peak Growth Rate**: Age of the child's peak growth rate, used to measure predictive power of parent data.

9.2 Correlation Analysis

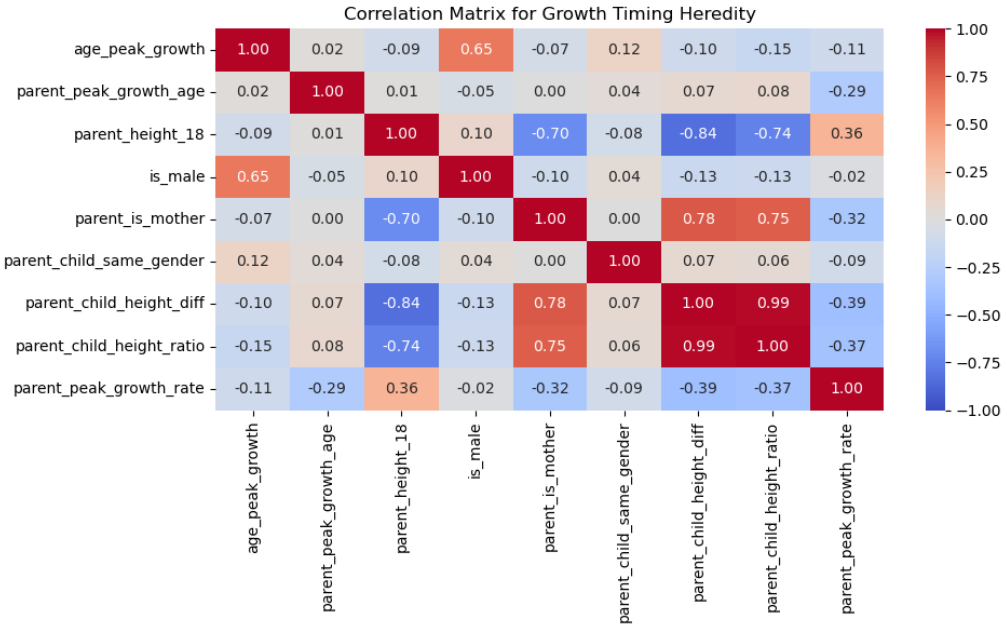


Figure 9: Correlation between Parent Features and Child Pubertal Growth Magnitude

The correlation matrix identified several parental features that correlate with a child's pubertal growth timing. As seen in Figure 9, the child's age at peak growth exhibited a weak correlation with parental peak growth age $r = 0.02$, suggesting minimal direct inheritance of growth timing. However, sex was a strong predictor, with males experiencing later growth spurts $r = 0.65$. Additionally, parental peak growth rate had a negative correlation $r = -0.11$, implying that higher parental growth rates might slightly predict earlier peak growth in children. Height-based metrics also showed modest effects; parent-child height difference $r = -0.10$ and parent-child height ratio $r = -0.15$ correlated weakly with earlier growth timing. While parent-child same-gender pairs exhibited a minor correlation $r = 0.12$,

the overall hereditary influence on growth timing appears limited compared to sex-based differences. These findings suggest that while parental growth traits contribute marginally, the child's sex is the dominant predictor of pubertal growth onset and duration.

9.3 Predictive Modeling

Regression-based and ensemble models were developed:

Model	R^2	MSE
Linear Regression	0.564449	0.749495
Ridge Regression	0.562969	0.752041
Random Forest Regressor	0.354869	1.110139

Table 5: Model Performance Metrics for Predicting Growth Spurt Timing

Displaying feature importance of the Random forest.

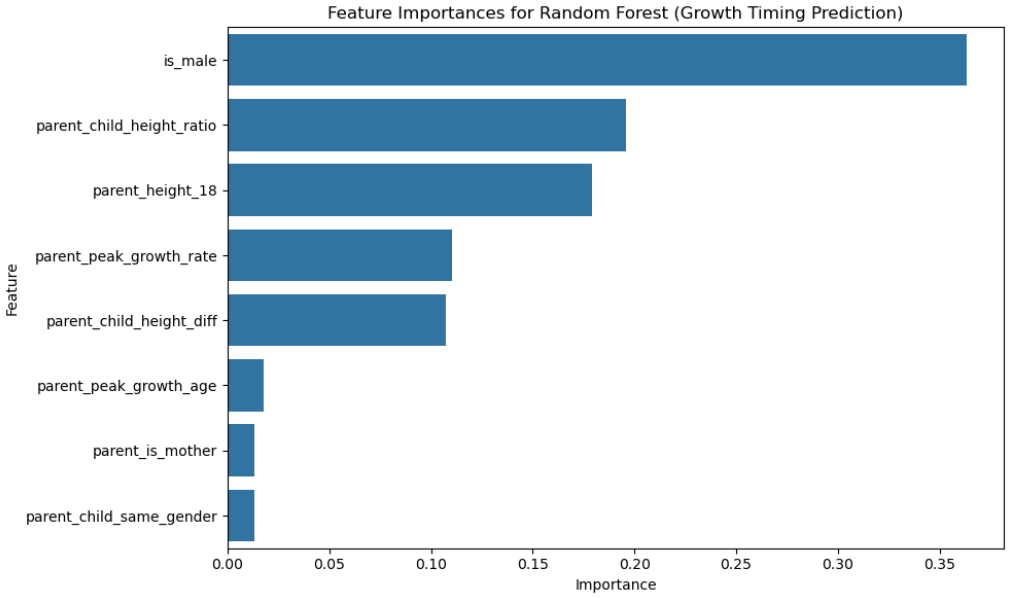


Figure 10: Feature Importances for Random Forest (Pubertal Growth)

Figure 10 presents the feature importances from a Random Forest model predicting `peak_growth_age`. The most influential feature is `is_male`, indicating that biological sex plays a dominant role in determining growth timing, likely due to the known hormonal differences affecting puberty onset and peak growth velocity seen in the above heredity analysis.

The second most important feature, `parent_child_height_ratio`, suggests that a child's height relative to their parent's height is a strong predictor of growth timing, reflecting hereditary influences on stature. Similarly, `parent_height_18` highlights that a parent's height at age 18 is an important factor, likely capturing genetic predisposition to growth patterns.

Other significant features include `parent_peak_growth_rate`, which represents the rate of maximal growth, and `parent_child_height_diff`, reinforcing the importance of height deviations

between parents and children in predicting growth timing.

Surprisingly features such as `parent_peak_growth_age` only have a marginal impact which suggests that the timing of peak growth in parents has more to do with other factors.

9.4 Conclusion

This analysis reveals a complex interplay of hereditary factors influencing a child's pubertal growth spurt timing. While the models used explain approximately 50% of the variance in growth timing using parental characteristics, biological sex emerges as the dominant predictor, with males consistently experiencing later growth spurts ($r = 0.65$). Parent-child height relationships (ratio and difference) demonstrate moderate predictive power, suggesting that hereditary height patterns partially determine when growth spurts occur.

Interestingly, direct transmission of growth timing appears minimal, with parental peak growth age showing only marginal correlation ($r = 0.02$) with child growth timing. Instead, parental growth rate exhibits a weak negative correlation ($r = -0.11$), indicating that parents with rapid growth may have children with earlier growth spurts. These findings suggest that while hereditary components significantly shape developmental trajectories, the mechanisms of transmission are not straightforward and likely involve complex genetic and hormonal interactions.

10 Heredity Analysis: Parent/Child Sex Combinations

This section explores the heredity differences between various parent/child sex combinations.

10.1 Feature Engineering

To analyze hereditary patterns from parent-child sex combinations, the following parental features were engineered:

- **Parent Child Sex Combo**: Combination of sex from parent to child: 'Mother-Son','Mother-Daughter','Father-Son','Father-Daughter'.
- **Parent Child Height Ratio**: Ratio of parent height to child height.
- **Parent Height at 18**: Final recorded height of parent at 18.
- **Parent Child height diff**: Difference between final parent and child's height.
- **Parent peak growth rate and peak growth age**: The age and rate at which the parent grew the most compared to the previous year.
- **Parent Child height diff**: Difference between final parent and child's height.
- **Child Last Height**: Final measurement of child height rate, used to measure predictive power of parent data and parent-child sex combinations.

10.2 Exploratory Data Analysis

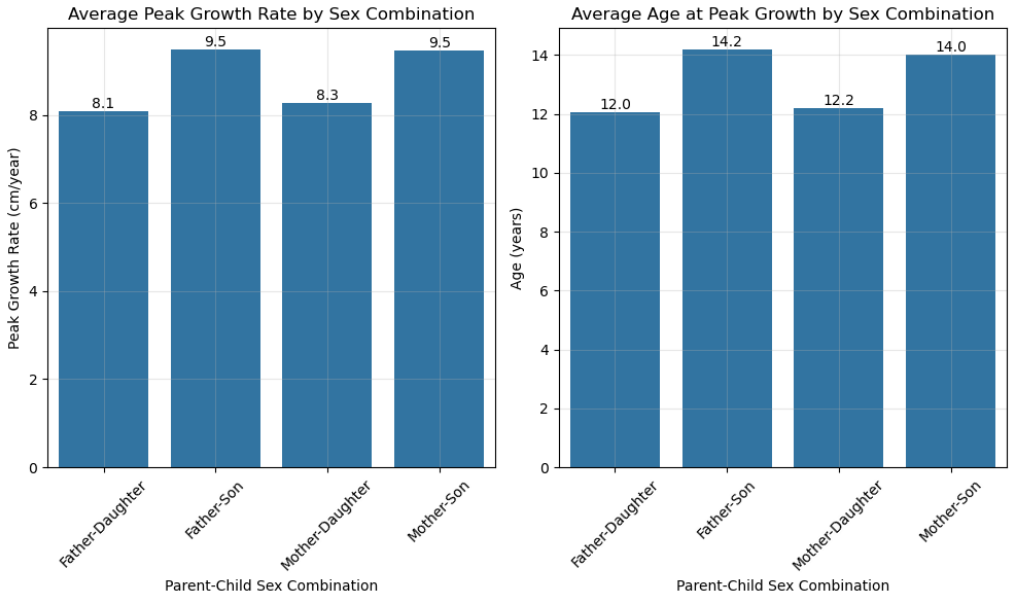


Figure 11: Differences in average growth rate and peak growth age by sex combination

The above bar chart (Figure 11) demonstrates that hereditary growth patterns are primarily influenced by the sex of the child rather than the sex of the parent. The approximately 2-year difference in timing of peak growth between sons and daughters aligns with that especially when compared to the minimal differences between same-sex children with different-sex parents (0.2 years or less).

10.3 Correlation Analysis

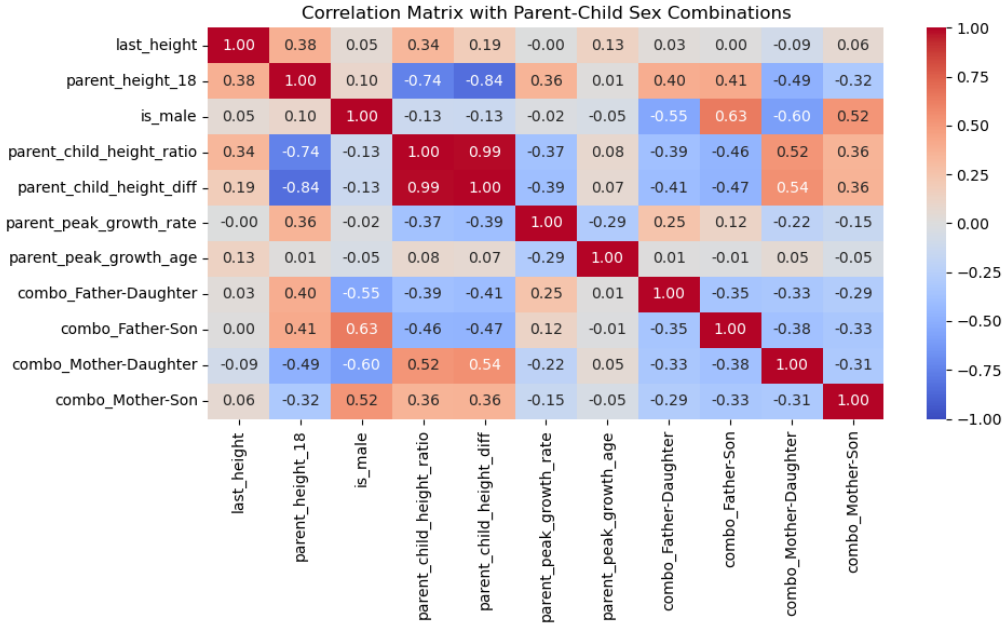


Figure 12: Correlation between sex combinations and last height

The analysis of the correlation matrix (Figure 12) reveals several notable patterns regarding the relationships between parent-child height characteristics and sex combinations.

Looking at the child's final height `last_height`, there is a moderate positive correlation with parent height at age 18 `parent_height_18` with $r = 0.38$, suggesting parental final height is a meaningful predictor of child's adult height. The parent-child height ratio also shows a positive correlation with the child's final height $r = 0.34$.

When examining sex-specific patterns, the data suggests that parent-child sex pairings have minimal impact on a child's final height, with all values being close to zero. The strongest relationship appears to be the Mother-Daughter combination with a slight negative correlation of -0.09, suggesting that this pairing might very slightly predict lower final height compared to other combinations. The Mother-Son combination shows a weak positive correlation of 0.06.

This suggests that paternal height may have a slightly different influence on offspring height compared to maternal height.

10.4 Predictive Modeling

Regression-based and ensemble models were developed:

Model	R^2	MSE
Linear Regression	0.360032	18.091937
Ridge Regression	0.356841	18.182135
Random Forest Regressor	0.432389	16.046390

Table 6: Model Performance Metrics for Pubertal Growth Prediction

Displaying feature importance based on Random Forest.

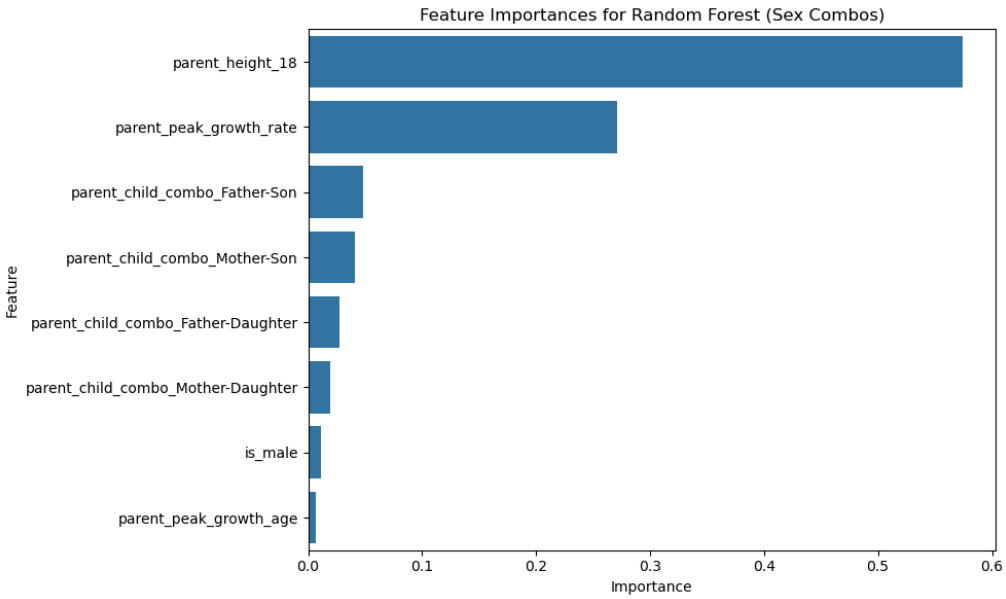


Figure 13: Feature Importances for Random Forest (Sex Combos)

The analysis of feature importances from the Random Forest model (Figure 13) focusing on sex combinations reveals several key insights into the hereditary factors affecting growth patterns.

Parent height at age 18 `parent_height_18` emerges as the most influential feature with an importance score of approximately 0.58, substantially higher than any other feature. This finding strongly suggests that a parent's final adult height is the predominant hereditary factor in predicting a child's growth patterns across sex combinations. The second most important feature is parent peak growth rate `parent_peak_growth_rate` with an importance score of about 0.28, indicating that the velocity of a parent's growth during their pubertal period substantially influences their offspring's growth trajectory. This reflects the hereditary nature of growth velocity patterns across generations.

Among the parent-child sex combinations, the Father-Son pairing shows the highest importance (approximately 0.05), followed closely by Mother-Son combinations. This suggests that male children's growth patterns may be more strongly influenced by parental characteristics than female children's, regardless of which parent is considered.

The Father-Daughter and Mother-Daughter combinations show slightly lower importance scores (around 0.03-0.04), indicating that while these relationships are still relevant, they may have less predictive power than the combinations involving sons.

Interestingly, the child's sex `is_male` and parent's peak growth age `parent_peak_growth_age` show minimal importance in the model, with scores below 0.02. This suggests that once specific parent-child sex combinations are accounted for, the individual sex variable provides little additional predictive value, and the timing of a parent's peak growth has less influence than the magnitude of that growth.

10.5 Conclusion

Our analysis reveals that while parent height at maturity is the dominant hereditary factor in determining a child's growth patterns, the transmission operates largely independent of parent sex. The weak correlations between parent-child sex combinations and final height, coupled with consistent sex-specific growth trajectories, demonstrate that a child's biological sex exerts the primary influence on growth timing and velocity. Parent peak growth rate serves as the second most influential predictor, indicating that both final height and growth velocity mechanisms transfer across generations. Though subtle variations exist, with male children showing slightly stronger parental influence regardless of parent sex, the overall hereditary pattern suggests that growth characteristics are transmitted through complex mechanisms where sex-linked expression in the child outweighs parent-specific contributions.

A Gen1 Summaries

	gen1_id	age	SHgt_cm
count	3636.00000	3636.000000	3326.000000
mean	574.80198	10.350000	135.439189
std	114.14434	6.324272	38.831436
min	370.00000	0.100000	50.627457
25%	490.00000	4.750000	105.990771
50%	569.00000	11.250000	146.593052
75%	679.00000	15.625000	166.793640
max	774.00000	20.000000	197.081343

Table 7: Gen1 Train Summary

	gen1_id	age	SHgt_cm
count	2232.000000	2232.000000	2016.000000
mean	570.661290	10.350000	133.756496
std	114.745977	6.324819	38.298998
min	368.000000	0.100000	48.391836
25%	474.000000	4.750000	103.805282
50%	574.000000	11.250000	145.012864
75%	663.000000	15.625000	165.460237
max	768.000000	20.000000	192.558592

Table 8: Gen1 Test Summary

B Gen2 Summaries

	gen2_id	study_parent_id_new	AgeGr	SHgt_cm	Wgt_kg
count	4224.000000	4224.000000	4224.000000	3712.000000	2179.000000
mean	2650.552083	556.604167	7.140909	118.697202	38.362150
std	139.645270	133.354678	5.646657	37.809589	16.880285
min	1332.000000	262.000000	0.100000	49.896709	4.636903
25%	2574.500000	453.250000	1.500000	83.766677	24.263125
50%	2648.500000	550.500000	6.500000	121.085655	34.270051
75%	2736.250000	668.000000	12.000000	151.367367	50.468043
max	3012.000000	774.000000	18.000000	196.140713	112.812058

Table 9: Gen2 Train Summary

	gen2_id	study_parent_id_new	AgeGr	SHgt_cm	Wgt_kg
count	1232.000000	1232.000000	1232.000000	1100.000000	409.000000
mean	2674.340909	542.215909	3.435714	96.240499	24.205836
std	103.220902	131.443688	2.963793	25.654321	5.097179
min	2332.000000	274.000000	0.100000	49.772812	14.932562
25%	2594.250000	423.250000	0.750000	73.318169	20.279088
50%	2673.500000	530.000000	2.500000	95.548570	23.567309
75%	2762.500000	649.000000	6.000000	118.999817	27.337289
max	2831.000000	768.000000	9.000000	147.217952	46.443758

Table 10: Gen2 Test Summary

C `create_features` Function

```
def create_features(df, parent_data):
    """
    Create features for height prediction with robust NaN handling

    Parameters:
    -----------
    df : DataFrame
        Generation 2 data (children)
    parent_data : DataFrame
        Parent heights data

    Returns:
    --------
    DataFrame
        Features for model training/prediction
    """
    # Merge with parent data
    merged = pd.merge(
        df,
        parent_data,
        left_on=['study_parent_id_new', 'AgeGr'],
        right_on=['gen1_id', 'parent_age'],
        how='left'
    )

    # Extract gender features - categorical to numeric
    merged['is_male'] = (merged['sex_assigned_at_birth'] == 'M').astype(int)
    merged['parent_is_mother'] = (merged['study_parent_sex'] == 'mother').astype(int)

    # Parent-child gender interaction
    parent_child_same_gender = ((merged['sex_assigned_at_birth'] == 'M') &
                                (merged['study_parent_sex'] == 'father')) | \
                               ((merged['sex_assigned_at_birth'] == 'F') &
                                (merged['study_parent_sex'] == 'mother'))
    merged['parent_child_same_gender'] = parent_child_same_gender.astype(int)

    # Calculate BMI if weight data is available (with NaN safety)
    if 'Wgt_kg' in merged.columns:
        # Safe division with proper NaN handling
        merged['bmi'] = np.where(
            (merged['Wgt_kg'].notna()) & (merged['SHgt_cm'].notna()) & (merged['SHgt_cm'
                                                                               ] > 0),
            merged['Wgt_kg'] / ((merged['SHgt_cm']/100)**2),
            np.nan
```

```
    )

features = []
# Process each subject individually
for subject_id in merged['gen2_id'].unique():
    subject_data = merged[merged['gen2_id'] == subject_id].sort_values('AgeGr')

    # Get gender information (these should be consistent within a subject)
    is_male = subject_data['is_male'].iloc[0] if not subject_data.empty else 0
    parent_is_mother = subject_data['parent_is_mother'].iloc[0] if not subject_data.
                                                    empty else 0
    parent_child_same_gender = subject_data['parent_child_same_gender'].iloc[0] if
                                                    not subject_data.empty else 0

    # Extract early measurements (age <= 9) - with robust filtering
    early_data = subject_data[subject_data['AgeGr'] <= 9]
    early_heights = early_data['SHgt_cm'].values if not early_data.empty else np.
                                                    array([])
    early_ages = early_data['AgeGr'].values if not early_data.empty else np.array([]
                                                    )

    # Extract later measurements (10 <= age <= 18) - with robust filtering
    later_data = subject_data[(subject_data['AgeGr'] >= 10) & (subject_data['AgeGr']
                                                     <= 18)]
    later_heights = later_data['SHgt_cm'].values if not later_data.empty else np.
                                                    array([])
    later_ages = later_data['AgeGr'].values if not later_data.empty else np.array([]
                                                    )

    # Growth velocity calculation with NaN safety
    growth_velocity = np.nan
    if len(early_heights) >= 2 and len(early_ages) >= 2:
        age_diff = early_ages[-1] - early_ages[0]
        if age_diff > 0:  # Prevent division by zero
            growth_velocity = (early_heights[-1] - early_heights[0]) / age_diff

    # Peak Growth Rate calculation with NaN safety
    peak_growth_rate, age_peak_growth = np.nan, np.nan
    if len(later_heights) >= 2 and len(later_ages) >= 2:
        # Calculate growth rates between consecutive measurements
        age_diffs = np.diff(later_ages)
        height_diffs = np.diff(later_heights)

        # Only calculate rates where age difference is positive (to avoid division
                                                        by zero)
        valid_indices = age_diffs > 0
        if np.any(valid_indices):
            growth_rates = np.full_like(age_diffs, np.nan, dtype=float)
            growth_rates[valid_indices] = height_diffs[valid_indices] / age_diffs[
                                                            valid_indices]

            # Find max growth rate if there are valid rates
            if not np.all(np.isnan(growth_rates)):
                peak_growth_rate = np.nanmax(growth_rates)
                # Find index of max growth rate
                max_idx = np.nanargmax(growth_rates)
                age_peak_growth = later_ages[max_idx + 1]

    # Extract key height measurements with NaN safety
```

```
        height_at_6 = subject_data[subject_data['AgeGr'] == 6]['SHgt_cm'].iloc[0] if not
                                                    subject_data[subject_data['AgeGr']
                                                    == 6].empty else np.nan
        height_at_9 = subject_data[subject_data['AgeGr'] == 9]['SHgt_cm'].iloc[0] if not
                                                    subject_data[subject_data['AgeGr']
                                                    == 9].empty else np.nan
        height_at_12 = subject_data[subject_data['AgeGr'] == 12]['SHgt_cm'].iloc[0] if
                                                    not subject_data[subject_data['AgeGr
                                                    '] == 12].empty else np.nan

        # Parent height at maturity (age 18)
        parent_height_18 = subject_data[subject_data['parent_age'] == 18]['parent_height
                                                    '].iloc[0] if not subject_data[
                                                    subject_data['parent_age'] == 18].
                                                    empty else np.nan

        # Calculate derived features with NaN safety
        height_diff_9_6 = np.nan
        if pd.notna(height_at_9) and pd.notna(height_at_6):
            height_diff_9_6 = height_at_9 - height_at_6

        height_diff_12_9 = np.nan
        if pd.notna(height_at_12) and pd.notna(height_at_9):
            height_diff_12_9 = height_at_12 - height_at_9

        growth_rate_change = np.nan
        if pd.notna(peak_growth_rate) and pd.notna(growth_velocity):
            growth_rate_change = peak_growth_rate - growth_velocity

        parent_child_height_diff = np.nan
        parent_child_height_ratio = np.nan
        if pd.notna(height_at_9) and pd.notna(parent_height_18) and parent_height_18 > 0
                                                    :
            parent_child_height_diff = height_at_9 - parent_height_18
            parent_child_height_ratio = height_at_9 / parent_height_18

        # Get BMI if available
        bmi_at_9 = subject_data[subject_data['AgeGr'] == 9]['bmi'].iloc[0] if 'bmi' in
                                                    subject_data.columns and not
                                                    subject_data[subject_data['AgeGr'] =
                                                    = 9].empty else np.nan

        # Create a row for each target age (10-18)
        for target_age in range(10, 19):
            row_data = {
                'gen2_id': subject_id,
                'AgeGr': target_age,
                'last_height': early_heights[-1] if len(early_heights) > 0 else np.nan,
                'last_age': early_ages[-1] if len(early_ages) > 0 else np.nan,
                'growth_velocity': growth_velocity,
                'parent_height_18': parent_height_18,
                'height_at_9': height_at_9,
                'height_diff_9_6': height_diff_9_6,
                'height_diff_12_9': height_diff_12_9,
                'growth_rate_change': growth_rate_change,
                'peak_growth_rate': peak_growth_rate,
                'age_peak_growth': age_peak_growth,
                'parent_child_height_diff': parent_child_height_diff,
                'parent_child_height_ratio': parent_child_height_ratio,
                'is_male': is_male,
                'parent_is_mother': parent_is_mother,
```

```
                'parent_child_same_gender': parent_child_same_gender
            }

            # Add BMI if available
            if 'bmi' in subject_data.columns:
                row_data['bmi_at_9'] = bmi_at_9

            features.append(row_data)

    return pd.DataFrame(features)
```

D `prepare_train_data` Function

```
def prepare_train_data(train_features, gen2_train):
    """
    Add actual height values to training features

    Parameters:
    -----------
    train_features : DataFrame
        Features created for training
    gen2_train : DataFrame
        Original gen2 training data with actual heights

    Returns:
    --------
    DataFrame
        Training features with actual height values
    """
    train_with_target = pd.merge(
        train_features,
        gen2_train[['gen2_id', 'AgeGr', 'SHgt_cm']],
        on=['gen2_id', 'AgeGr'],
        how='left'
    )
    return train_with_target.dropna(subset=['SHgt_cm'])
```

E `train_model` Function

```
    def train_model(train_features, feature_cols=None, test_size=0.2, random_state=7):
    """
    Train and evaluate an XGBoost model for height prediction

    Parameters:
    -----------
    train_features : DataFrame
        Training features with target variable
    feature_cols : list, optional
        List of feature columns to use
    test_size : float, optional
        Proportion of data to use for validation
    random_state : int, optional
        Random seed for reproducibility

    Returns:
    --------
```

```
    tuple
        Trained model, preprocessor, and validation metrics
    """
    # Define default feature columns if not provided
    if feature_cols is None:
        feature_cols = [
            'AgeGr', 'last_height', 'growth_velocity',
            'parent_height_18', 'height_at_9', 'height_diff_9_6',
            'height_diff_12_9', 'peak_growth_rate', 'age_peak_growth',
            'parent_child_height_diff', 'parent_child_height_ratio',
            'is_male', 'parent_is_mother', 'parent_child_same_gender'
        ]
        # Add BMI if available
        if 'bmi_at_9' in train_features.columns:
            feature_cols.append('bmi_at_9')

    # Split train and validation sets, grouped by subject
    group_split = GroupShuffleSplit(n_splits=1, test_size=test_size, random_state=
                                                    random_state)
    train_idx, val_idx = next(group_split.split(
        train_features, groups=train_features['gen2_id']
    ))

    # Use MICE imputation for better handling of missing values
    preprocessor = Pipeline([
        ('imputer', IterativeImputer(max_iter=10, random_state=random_state)),
        ('scaler', StandardScaler())
    ])

    # Preprocess data
    X_train = preprocessor.fit_transform(train_features.iloc[train_idx][feature_cols])
    y_train = train_features.iloc[train_idx]['SHgt_cm'].values

    X_val = preprocessor.transform(train_features.iloc[val_idx][feature_cols])
    y_val = train_features.iloc[val_idx]['SHgt_cm'].values

    param_grid = {
    'max_depth': [3, 4],
    'learning_rate': [0.04, 0.05],
    'min_child_weight': [3, 4],
    'subsample': [0.65, 0.7],
    'colsample_bytree': [0.65, 0.7],
    'gamma': [0.03, 0.05],
    'n_estimators': [200, 250],
    'scale_pos_weight': [1.8, 2],
    'reg_alpha': [0.15, 0.2],
    'reg_lambda': [0.25, 0.3]
}

    # Setup cross-validation with group-aware split
    xgb_model = xgb.XGBRegressor(objective='reg:squarederror', random_state=random_state
                                                    )
    grid_search = GridSearchCV(
        estimator=xgb_model,
        param_grid=param_grid,
        cv=4,  # Could use GroupKFold here instead
        scoring='neg_root_mean_squared_error',
        verbose=1
    )

    # Run grid search
```

```
    grid_search.fit(X_train, y_train, groups=train_features.iloc[train_idx]['gen2_id'])
    best_params = grid_search.best_params_
    print(f"Best parameters: {best_params}")

    # Train the final model with optimal parameters
    dtrain = xgb.DMatrix(X_train, label=y_train)
    dval = xgb.DMatrix(X_val, label=y_val)

    params = {
        'objective': 'reg:squarederror',
        'eval_metric': 'rmse',
        'max_depth': best_params['max_depth'],
        'learning_rate': best_params['learning_rate'],
        'min_child_weight': best_params['min_child_weight'],
        'subsample': best_params['subsample'],
        'colsample_bytree': best_params['colsample_bytree'],
        'gamma': best_params['gamma'],
        'random_state': random_state
    }

    evals = [(dtrain, 'train'), (dval, 'eval')]

    model = xgb.train(
        params=params,
        dtrain=dtrain,
        num_boost_round=150,
        evals=evals,
        early_stopping_rounds=10,
        verbose_eval=True
    )

    # Evaluate the model
    y_val_pred = model.predict(dval, iteration_range=(0, model.best_iteration))
    val_rmse = np.sqrt(mean_squared_error(y_val, y_val_pred))
    val_mae = mean_absolute_error(y_val, y_val_pred)
    val_r2 = r2_score(y_val, y_val_pred)

    print(f"Validation RMSE: {val_rmse:.2f}")
    print(f"Validation MAE: {val_mae:.2f}")
    print(f"Validation RÂš: {val_r2:.2f}")

    # Create evaluation results
    val_results = pd.DataFrame({
        'gen2_id': train_features.iloc[val_idx]['gen2_id'].values,
        'AgeGr': train_features.iloc[val_idx]['AgeGr'].values,
        'actual': y_val,
        'predicted': y_val_pred,
        'error': y_val - y_val_pred
    })

    return model, preprocessor, feature_cols, val_results
```