MINI PROJECT REPORT

CA5304 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

EXPLAINABLE AI TEAM

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY, CHENNAI

Submitted by

VINOTHKUMAR A (2022179041)
SANJAYPRAKASH M (2022179043)
ARAVINDAMBALAVANAN J (2022179044)

Submitted to

DR. DEIVAMANI M
Assistant professor
(DIST)

ACKNOWLEDGEMENT

We would like to take this opportunity to express our sincere gratitude and appreciation to everyone who has supported and encouraged us throughout the completion of this mini project. First and foremost, we offer our heartfelt thanks to the Almighty God for His boundless blessings, unwavering guidance, and endless love, which have been the source of strength and inspiration throughout this journey. We are deeply indebted to the Management of Anna University for providing the necessary resources and facilities that enabled us to carry out this project effectively. Their support has been instrumental in our learning and growth as students. We extend our profound appreciation to the Head of the Department, Dr. S. Sridhar, for his invaluable guidance and encouragement. A special word of thanks goes to the Staff in charge, Dr. Deivamani M and Mr. Muthumani M (Industry Expert), for their constant support, patience, and willingness to assist whenever needed. Their expertise and insights have been invaluable in enhancing the quality of this work. We cannot forget to acknowledge the unwavering support and encouragement we received from our Parents. Their love, belief, and sacrifices have been the driving force behind our academic pursuits, and we are eternally grateful for their presence in our life. To our Friends and classmates, thank you for being the pillars of strength and motivation. Your camaraderie and the exchange of ideas made this journey enjoyable and memorable. Lastly, we want to thank all the individuals who have directly or indirectly contributed to the successful completion of this mini project. Your support, encouragement, and belief in our abilities have meant the world to us. In conclusion, this project has been a significant learning experience, and we are thankful to everyone who has been a part of it.

Thank you all!

Sincerely,

VINOTHKUMAR A (2022179041)
SANJAYPRAKASH M (2022179043)
ARAVINDAMBALAVANAN J (2022179044)

TABEL OF CONTENT

1.INTRODUCTION

Explainable Artificial Intelligence (XAI) refers to the development of AI systems that can provide clear, understandable, and interpretable explanations for their decision-making processes. As artificial intelligence systems become more complex and pervasive, there is a growing need for transparency and accountability in their decision-making mechanisms, especially in critical areas like healthcare, finance, and criminal justice.

Traditional AI models, particularly deep learning models, are often considered "black boxes" because their internal workings are intricate and difficult to comprehend. This lack of transparency can be a significant barrier, as users may be hesitant to trust or adopt AI systems when they cannot understand the reasoning behind their outputs.

XAI aims to address this issue by incorporating mechanisms that enable users to grasp the logic and factors influencing an AI system's decisions. This transparency not only enhances trust but also allows users to identify biases, errors, or unexpected behavior in the AI model.

ABSTRACT

Explainable AI (XAI) aims to demystify the decision-making process of complex models, providing users and stakeholders with insights into the factors influencing predictions.This project explores and implements three widely-used explainability techniques - SHAP (SHapley Additive exPlanations
,ELI5 (Explain Like I'm 5), and LIME (Local Interpretable Model-agnostic Explanations) - to enhance the interpretability of AI models.

The project involves integrating these three techniques into a cohesive framework for XAI. A diverse set of AI models, spanning from classical machine learning algorithms to deep neural networks, will be employed to demonstrate the versatility of the proposed approach. The project aims to showcase the strengths and limitations of each technique in different contexts, providing a comparative analysis of their effectiveness.
Additionally, the user interface will be developed to facilitate the interaction with the XAI framework, allowing users to input data, view explanations, and gain insights into the model's decision-making.

2.BACKGROUND

2.1.Problem Domain:

Salary prediction and explainability: addressing a significant real-world challenge by developing a model to predict whether an individual's salary is above or below 50K, and prioritizing explainability to understand the model's reasoning and ensure fairness.

UCI Adult dataset:tilizing a well-known dataset with demographic and employment data, making your findings applicable to broader research and potential real-world applications.

2.2.Model and Explainability:

2.2.1.Decision tree model: Chosen a model that balances interpretability with reasonable accuracy, allowing for a balance of prediction performance and understanding of decision-making.

2.2.2.Explainability techniques: You're going beyond basic model interpretation by employing multiple techniques for comprehensive insights
LIME: Local explanations for individual predictions, revealing how features influence specific outcomes.
SHAP: Global feature importance and contribution scores, providing overall understanding of feature impact.
eli5: Simplified explanations for broader accessibility, making results understandable for those with less technical expertise.

2.3.Model Comparison:

Lazypredict: You're exploring potential model improvements by efficiently evaluating different algorithms to identify those that might offer better accuracy or interpretability trade-offs.

2.4.Model Deployment and Accessibility:

Streamlit: You're making your model accessible and interactive through a user-friendly web application, enabling others to explore predictions and explanations, promoting understanding and potential real-world use cases.

Key Points:

- Chosen a model that balances interpretability and performance.
- Using diverse explainability techniques for comprehensive understanding.
- Comparing models to optimize results.
- Deploying the model effectively for accessibility and potential real-world use.

3.Data

The dataset used for analysis comprises a diverse set of demographic and socioeconomic attributes, offering valuable insights into the factors influencing individual income levels. The dataset contains information on the following variables:

Age: The age of individuals, providing a crucial demographic factor that can influence income.

Work Class: Categorization of individuals based on their employment, like private,federal gov,self emp etc.

Education: The educational attainment of individuals, offering insights into the correlation between education levels and income.

Marital Status: The marital status of individuals, a sociodemographic factor that can impact financial stability.

Occupation: The type of work or profession individuals are engaged in, shedding light on the diversity of job roles and their associated income levels.

Race: Information about the racial background of individuals, which may contribute to understanding potential disparities in income.

Sex: Gender information, providing insights into gender-based income variations.

Capital Gain and Capital Loss: Financial indicators that can impact an individual's overall income and financial health.

Hours per Week: The number of hours individuals work per week, a crucial factor influencing income levels.

Native Country: The country of origin or residence, offering insights into the influence of geographical location on income.

Income: The target variable indicating whether an individual's income is above 50k or lower than 50k, providing the basis for predictive modeling or analysis.

3.1.Data preparation

3.1.1.Data Loading and Exploration:

- Load the dataset using a suitable library like pandas.
- Examine its structure (shape, columns, data types).
- Visualize distributions and relationships between features using descriptive statistics and visualizations (e.g., histograms, scatter plots).
- Identify potential issues like missing values, outliers, and inconsistencies.

3.1.2.Data Cleaning:

Handle missing values: Use appropriate imputation methods (e.g., mean, median, mode, or more sophisticated techniques like KNN imputation) based on data characteristics and missingness patterns.
Consider removing rows with excessive missingness if appropriate.

Address inconsistencies: Correct errors and standardize formats for categorical variables.

Detect and address outliers: Use statistical methods or visualizations to identify outliers.
Decide whether to remove, cap, or winsorize them based on their impact and domain knowledge

3.1.3.Data preprocessing:

LabelEncoder transforms categorical data into numerical form, enabling machine learning models to process it effectively.

SMOTETomek is an ensemble technique for imbalanced classification that combines oversampling using SMOTE with undersampling using Tomek links.

3.1.4.Data Splitting:

Divide the dataset into training and testing sets (e.g., 80/20 split) for model development and evaluation.

4.APPROACH USED

4.1.Model choice:

Model name: Decision tree

Inherent Interpretability: Decision trees are naturally interpretable. Their structure, resembling a flowchart, clearly shows the rules and conditions used to make predictions. This makes them ideal for XAI because you can easily understand how a specific prediction was made and which features were most important.

Transparency and Fairness: The transparency of decision trees promotes responsible AI development. You can readily identify potential biases in the model's decision-making and address them, as opposed to "black box" models where understanding reasoning is difficult.

Explainability Techniques Synergy: Decision trees work well with various explainability techniques like LIME and SHAP. These techniques leverage the tree structure to provide further insights into feature importance and local explanations for individual predictions.

Performance Trade-off: While decision trees might not always achieve the highest accuracy compared to more complex models, they offer a good balance between accuracy and interpretability. This makes them suitable for situations where understanding the model's reasoning is a priority.

Resource Efficiency: Decision trees are generally computationally efficient and require less training data compared to some other algorithms. This makes them a good choice for smaller datasets or situations where resources are limited.

Overall, choice of a decision tree aligns well with project's focus on XAI. It allows you to build a model that is not only accurate but also easily understandable and transparent.

4.2.Explainability Techniques:

4.2.1.SHAP (Shapley Additive Explanations):SHAP to quantify the global importance of features and their contribution to individual predictions, offering a comprehensive view of feature impact.

Summary plot (shap.summary_plot):

Purpose:

- Creates a visual summary of feature importance based on SHAP values, a method for explaining model output by calculating the contribution of each feature to individual predictions.
- Provides insights into which features have the most influence on the model's decisions and how they impact predictions globally.

Key Features:

Feature Ranking: Features are ordered vertically based on their global importance, with the most important at the top.

SHAP Value Distributions: Each feature has a distribution plot showing its SHAP values across the dataset, indicating how it contributes to different predictions.

Color Coding:

- Red indicates a feature's value positively contributes to the model's output (pushing it towards a higher prediction).
- Blue indicates a negative contribution (pushing it towards a lower prediction).

Summary Bar: Displays the average impact of each feature across all instances, with positive contributions on the right and negative on the left.

Dependence plot(shap.dependence_plot):

Purpose:

- Uncovers how a feature's value influences model output, revealing patterns and relationships that might not be evident from global summaries.
- Helps understand feature interactions and non-linear relationships that impact predictions.

Key Features:

- Horizontal Axis: Represents the range of values for the chosen feature.
- Vertical Axis: Shows the corresponding SHAP values, indicating feature impact on model output.
- SHAP Value Scatter: Each dot represents a single instance, positioned based on its feature value and SHAP value.
- Trend Line: Captures the general pattern of feature impact, indicating positive or negative influence and potential non-linearities.
- Color Coding (Optional): Can represent a second feature to visualize interactions, revealing how the impact of one feature varies depending on the value of another.

4.2.2.Eli5: Leveraging eli5 to simplify model explanations, making them more accessible to those with less technical expertise.

Eli5 show weights(eli5.show_weights)

Purpose:

- Provides a simplified view of how a model's learned weights contribute to feature importance.
- Helps identify which features have the most influence on predictions.

Functionality:

- Calculates feature importance based on model weights.
- Displays the results in a clear and concise format, often using text-based explanations.

Key Points:

Model-Specific: The exact interpretation of weights and feature importance depends on the type of model being used.

Global Importance: Shows the overall importance of features across the entire dataset.

Not Local Explanations: Doesn't provide explanations for individual predictions.

4.2.3.LIME (Local Interpretable Model-agnostic Explanations):Using LIME to provide granular explanations for individual predictions, revealing how specific features impact the model's output for each instance.

Explain instance(explainer.explain_instance)

Purpose:

- Generates a local explanation for a specific prediction made by a machine learning model.
- Reveals how individual features contributed to that particular prediction, providing insights into the model's decision-making process for that instance.

Output:

Feature importance scores: Indicate the extent to which each feature influenced the prediction, either positively or negatively.

Explanation visualization (optional): LIME can often visualize explanations as bar charts or force plots for intuitive understanding.

Functionality:

Local approximation: LIME creates a simpler, interpretable model (often a linear model) around the specific instance to approximate the original model's behavior locally.

Feature perturbation: It perturbs the instance's features to observe how the model's prediction changes, determining feature importance.

4.3.Model Comparison:

4.3.1.Lazypredict: You're exploring potential model enhancements by efficiently comparing different algorithms, seeking to identify models that might balance interpretability with accuracy more effectively.

Lazy classifier(LazyClassifier)

Purpose:

- Simplifies the process of building and comparing multiple classification models within LazyPredict.
- Facilitates quick model selection and evaluation for classification tasks**.**

Key Features:

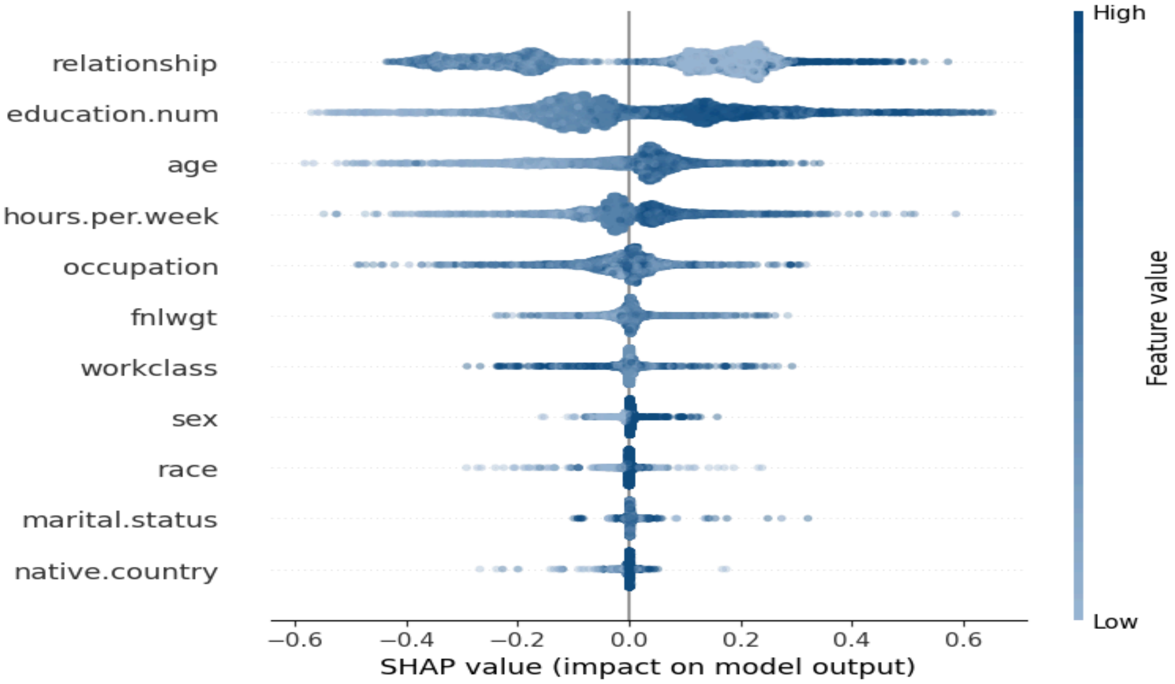
- Automatic Model Training: Trains a variety of classification algorithms on your dataset with a single line of code.
- Performance Evaluation: Calculates and displays various performance metrics (accuracy, precision, recall, F1-score, AUC-ROC) for each model.
- Hyperparameter Tuning: Automatically tunes hyperparameters for some algorithms to optimize performance.
- Best Model Identification: Recommends the best-performing model based on the chosen evaluation metrics.
- Model Saving and Loading: Enables saving trained models for later use and loading them back for predictions.

4.4.Deployment :

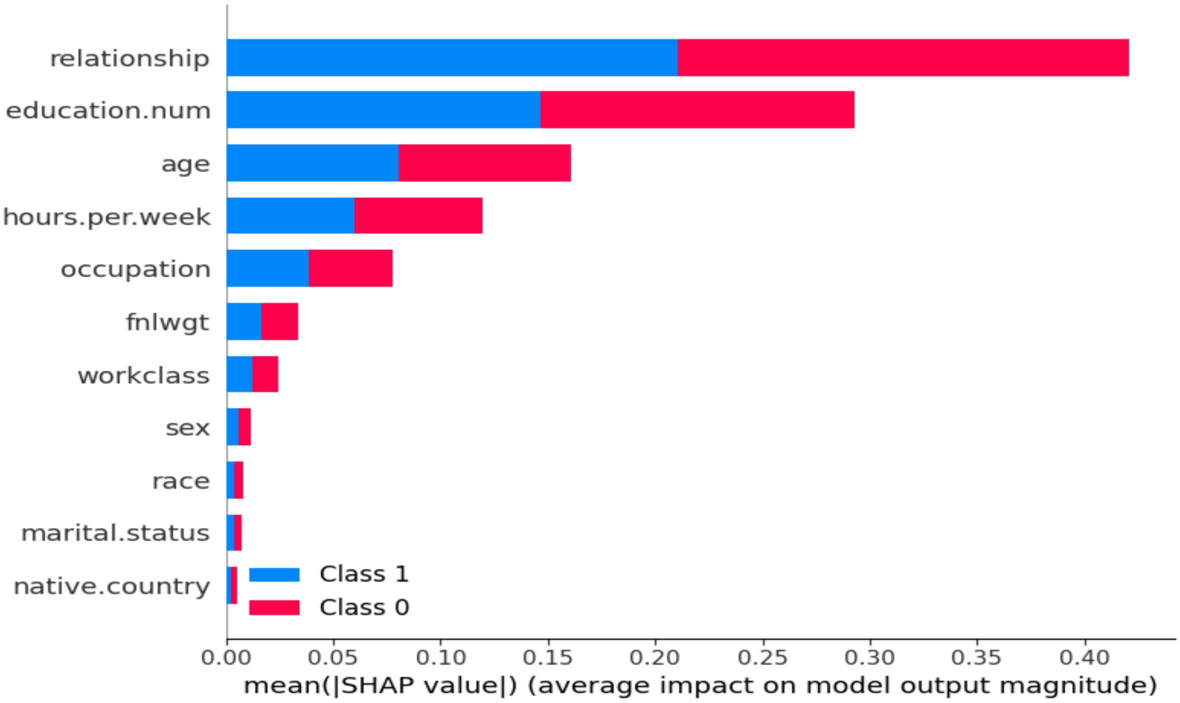
Streamlit: Making model and explanations interactive and accessible through a user-friendly web application, enabling others to explore predictions and understand model reasoning.

5.RESULT

Summary ploy:

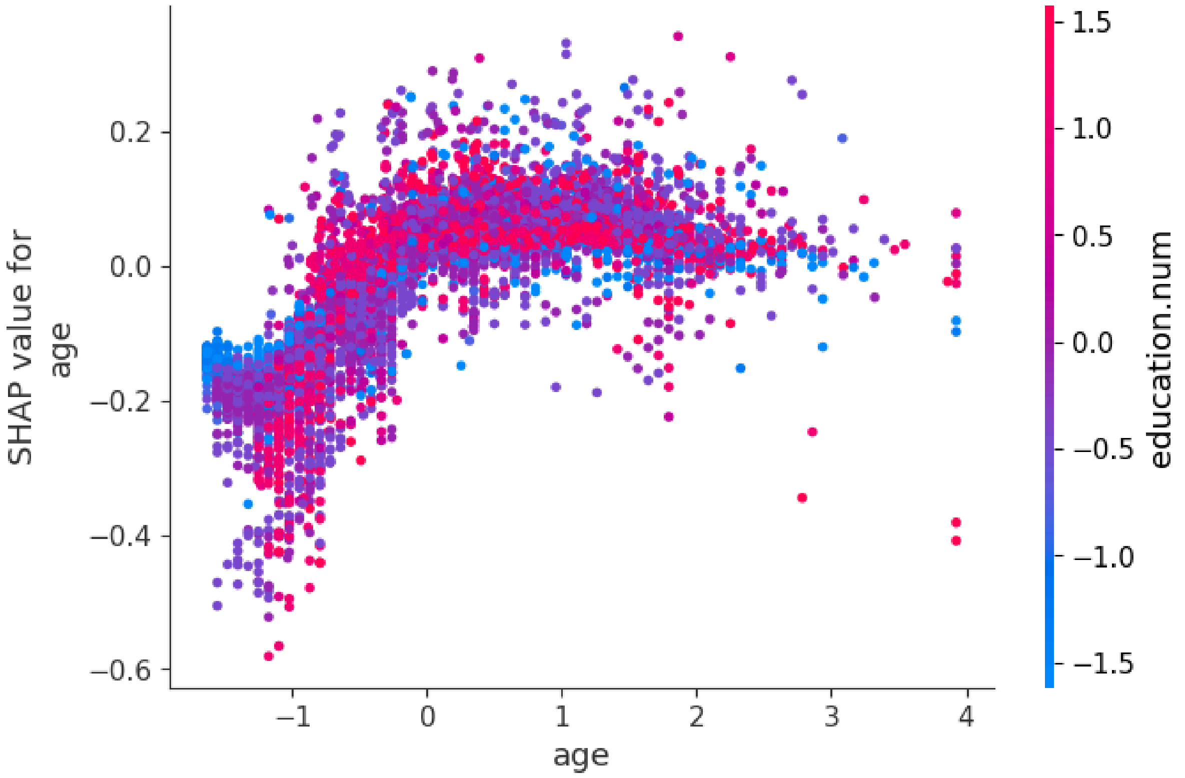


Summary bar plot:

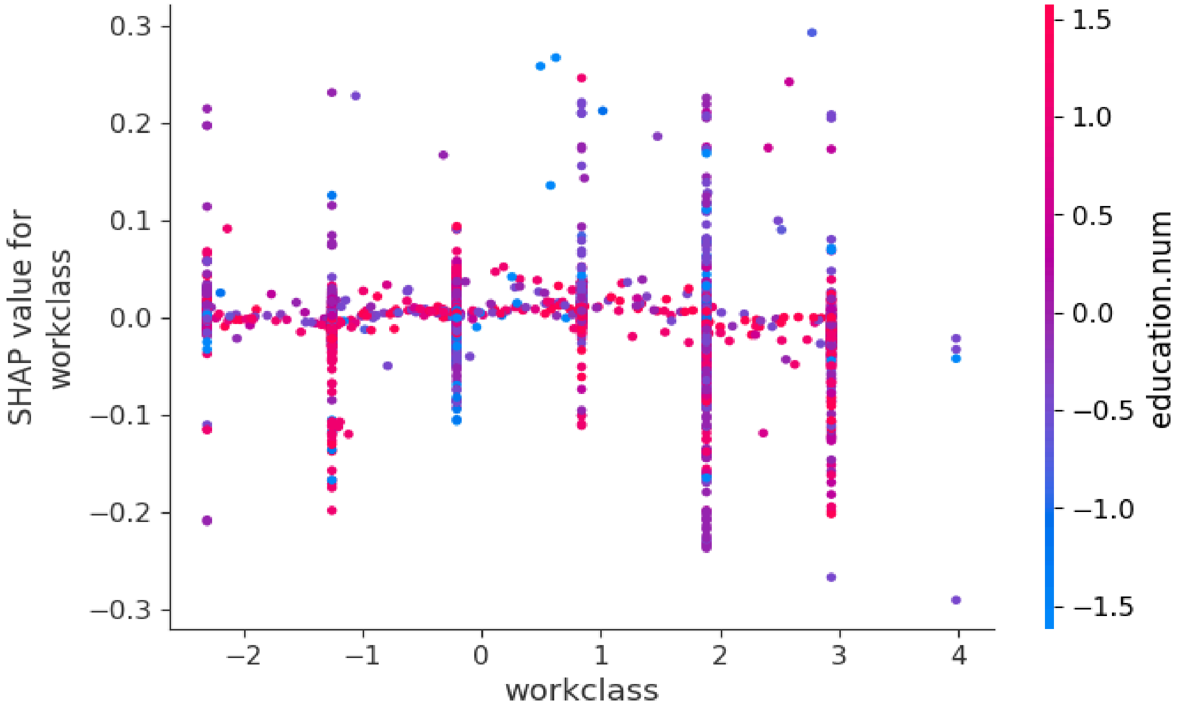


Dependency plot:

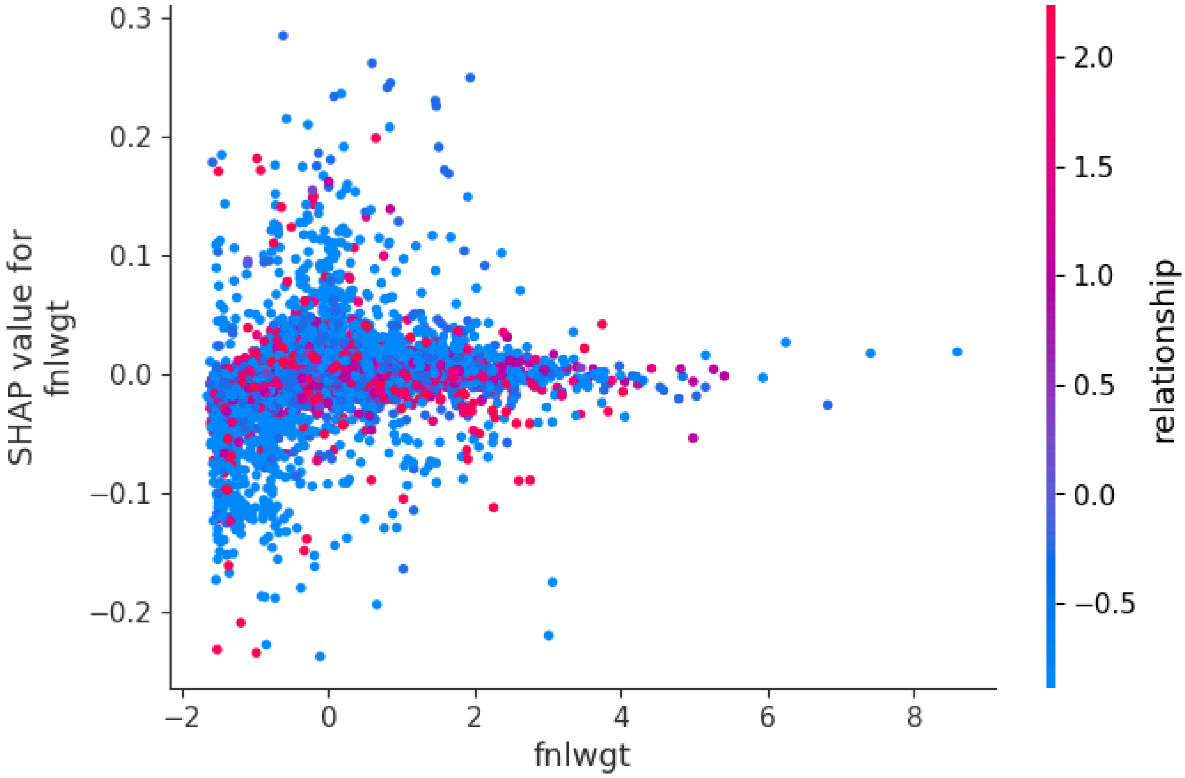
Age:



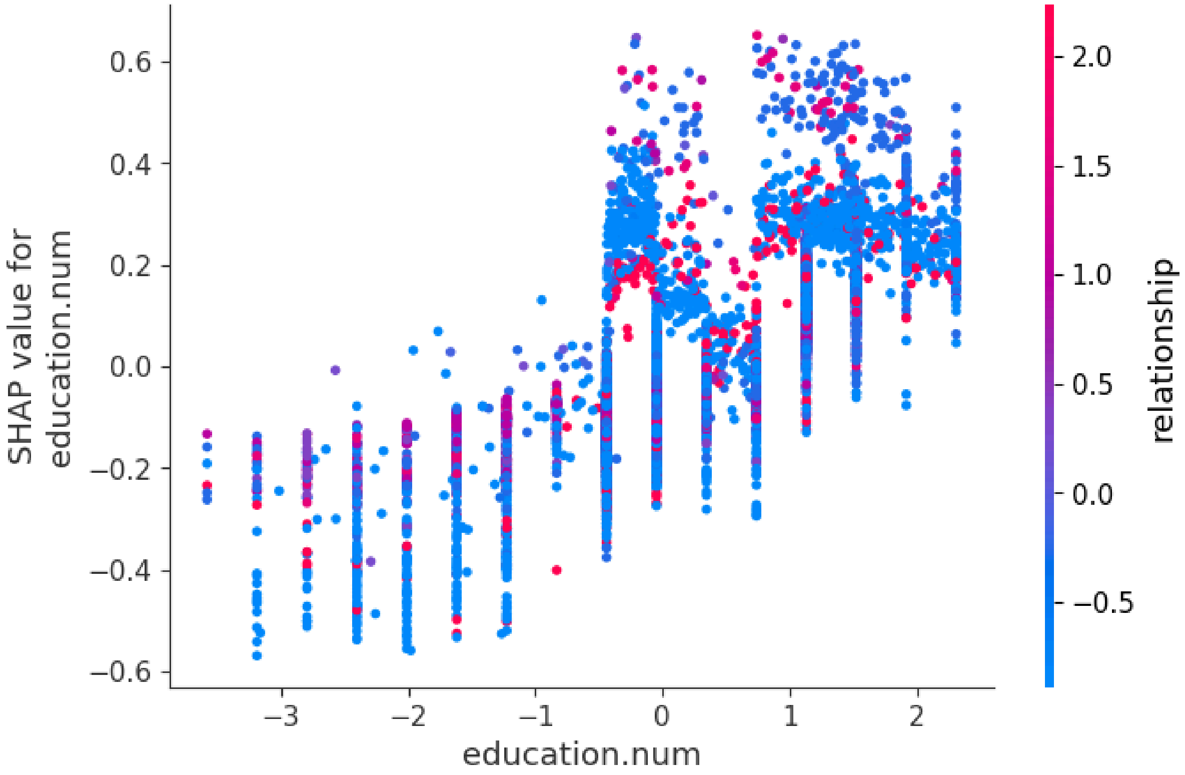
Work class:



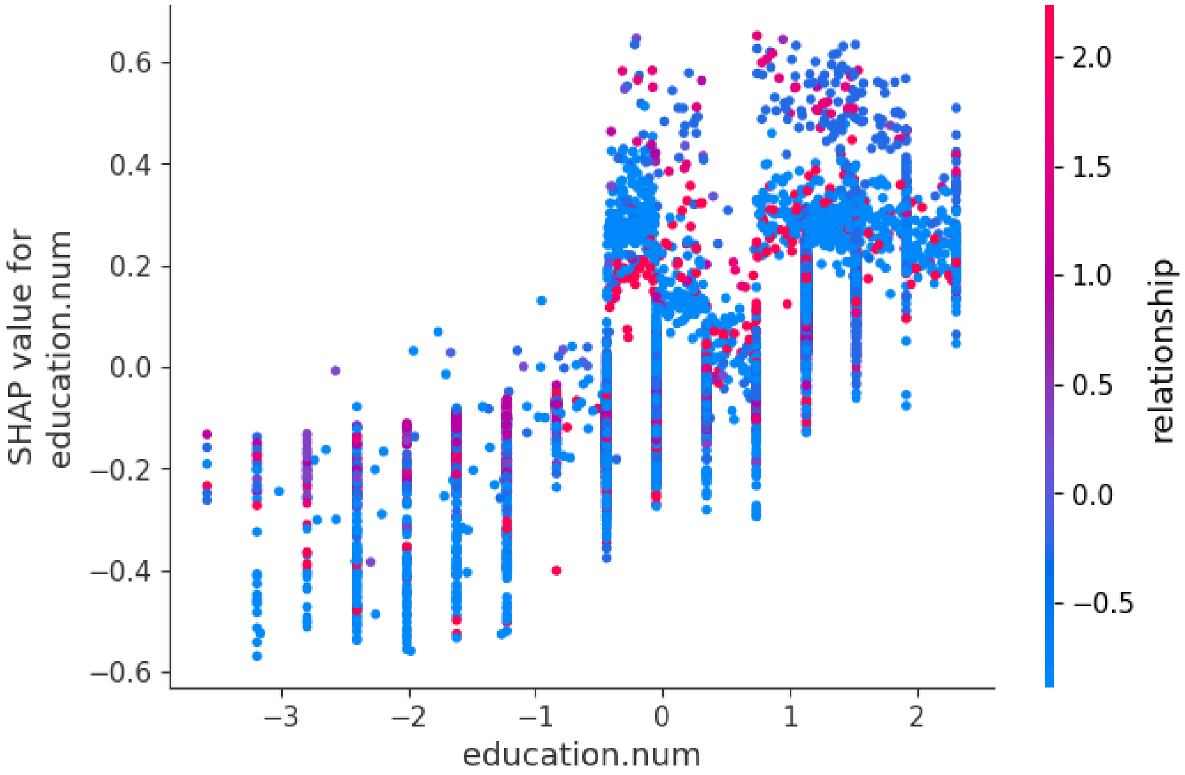
Fnlwgt:



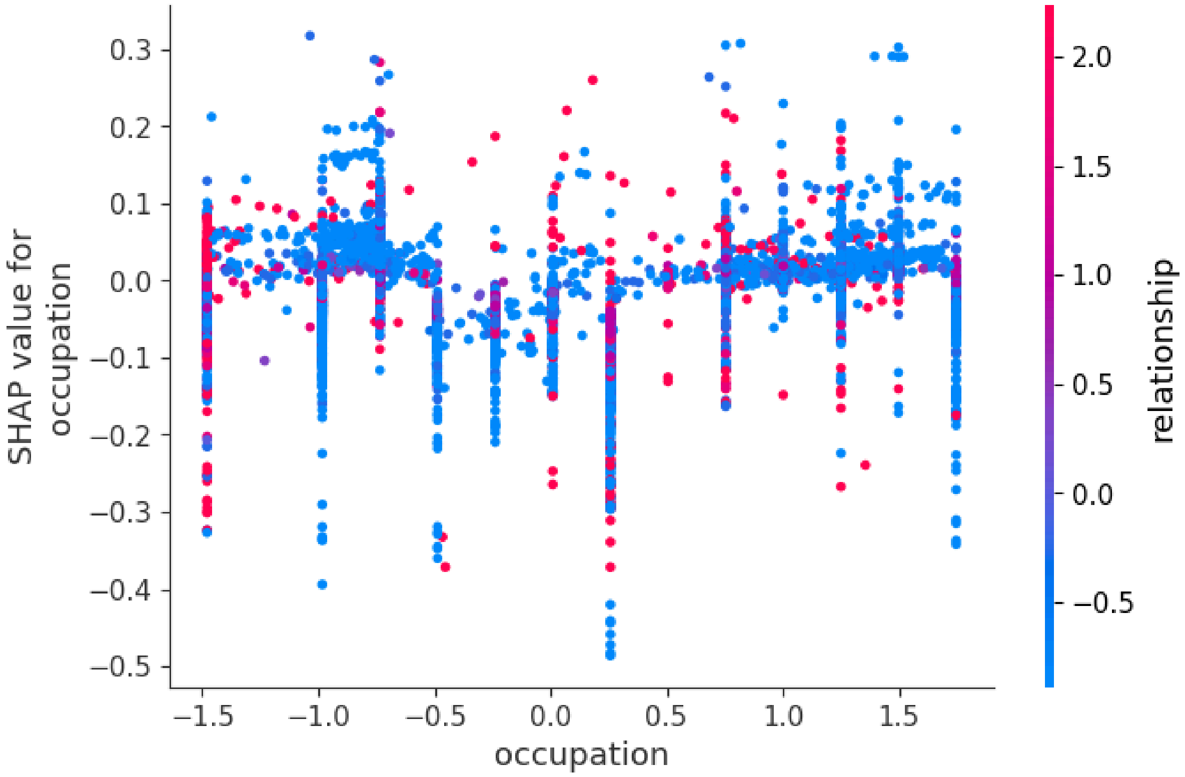
Education num:



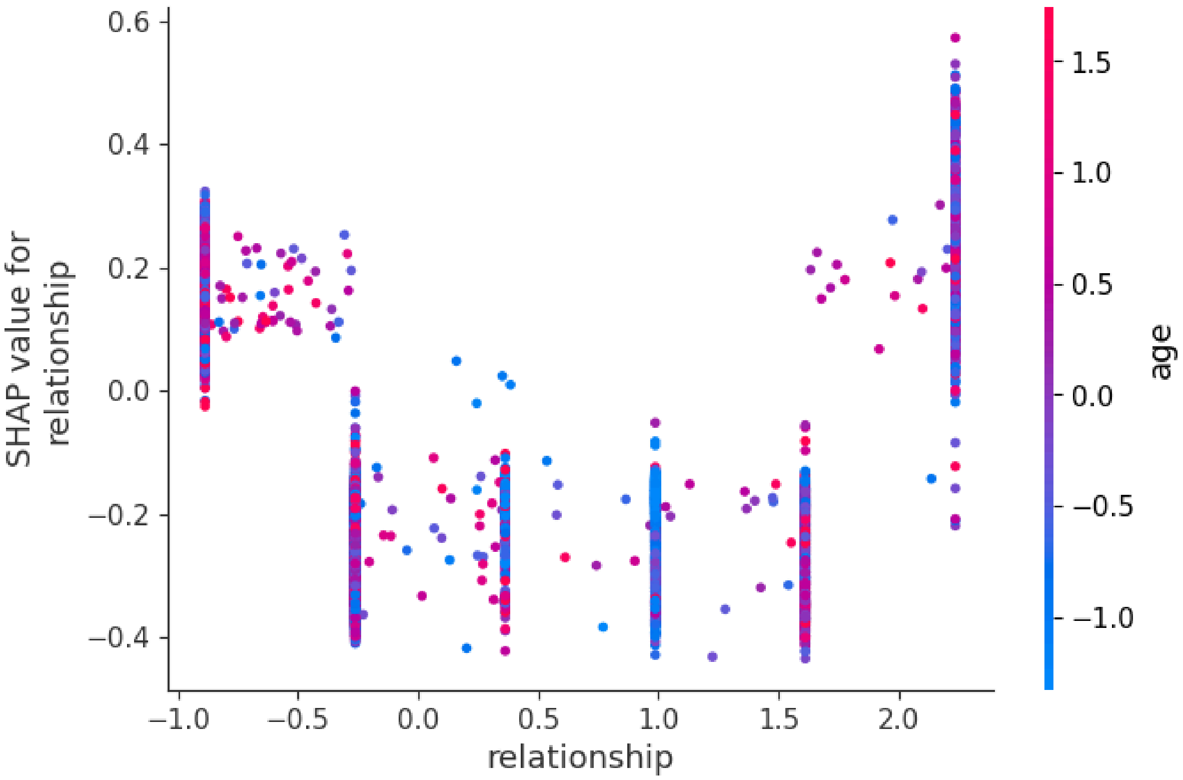
Martial status:



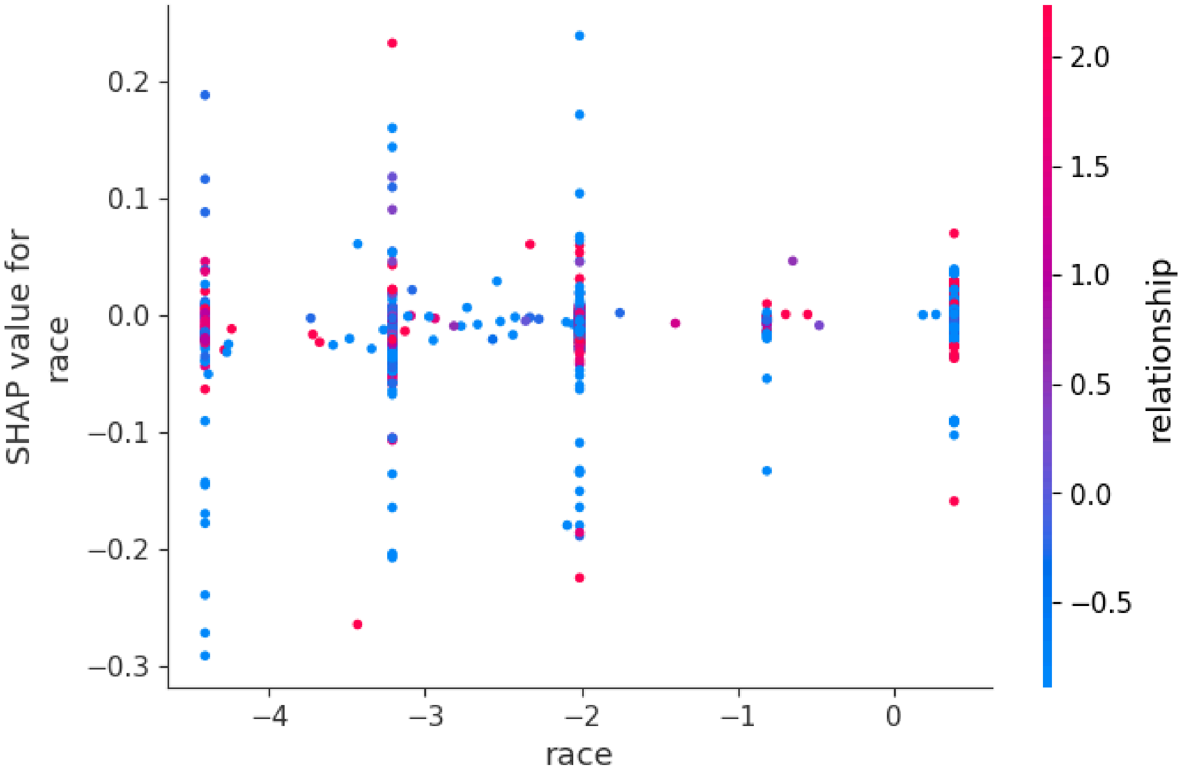
Occupation:



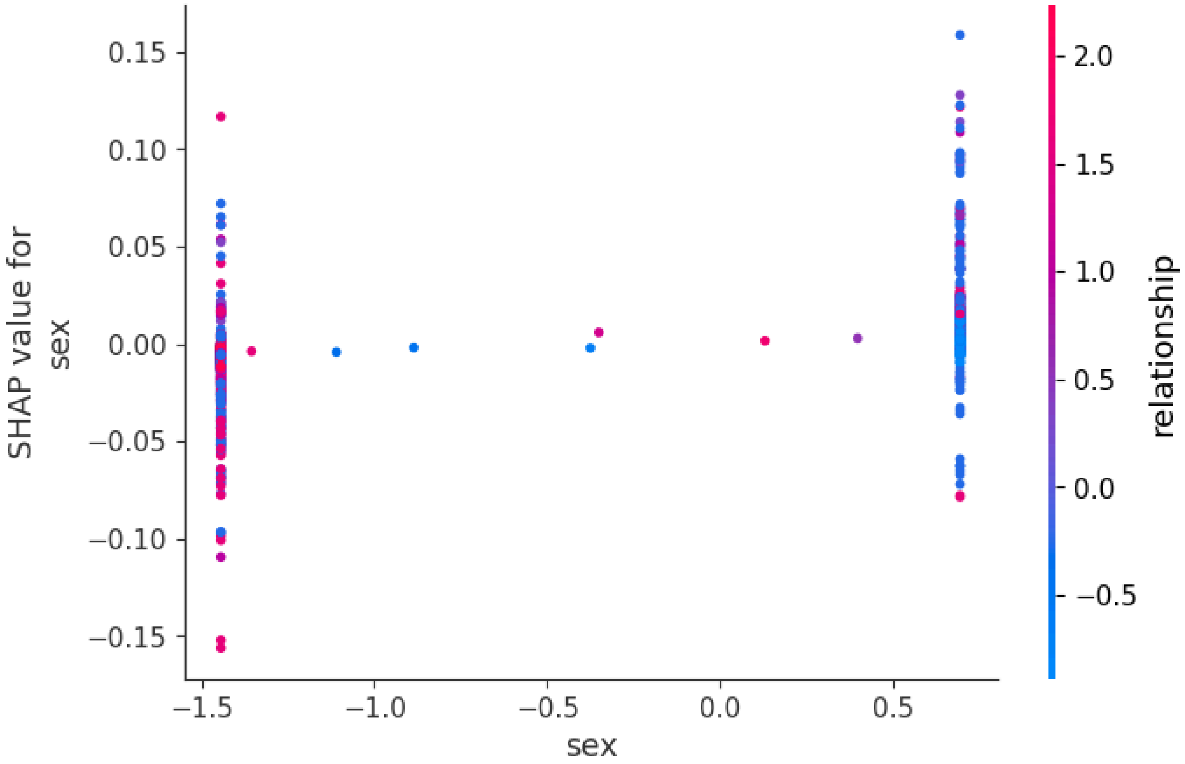
Relationship:



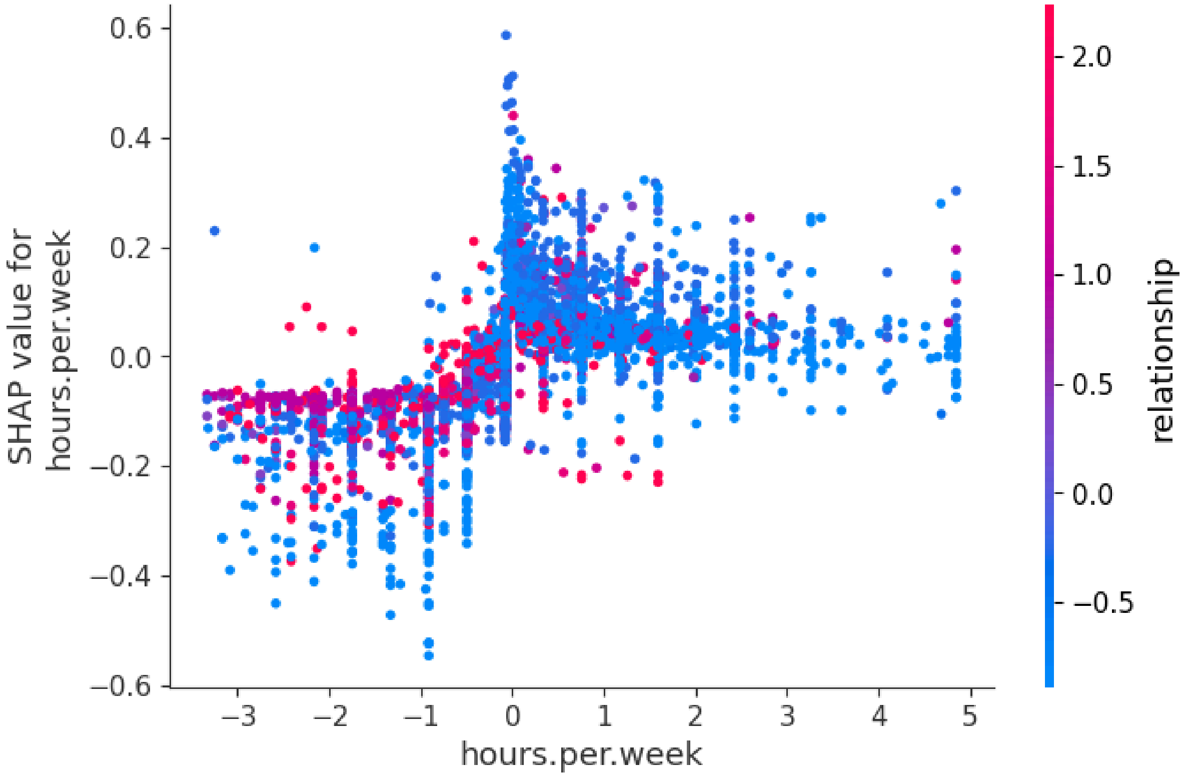
Race:



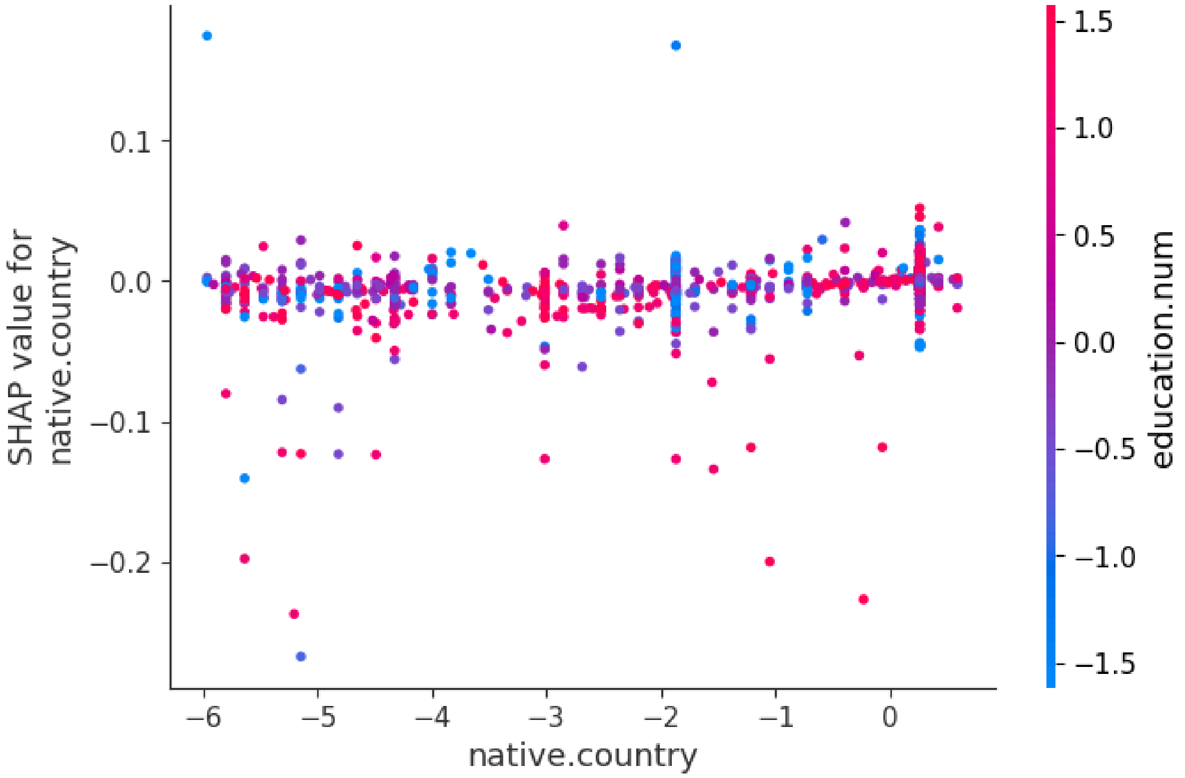
Sex:



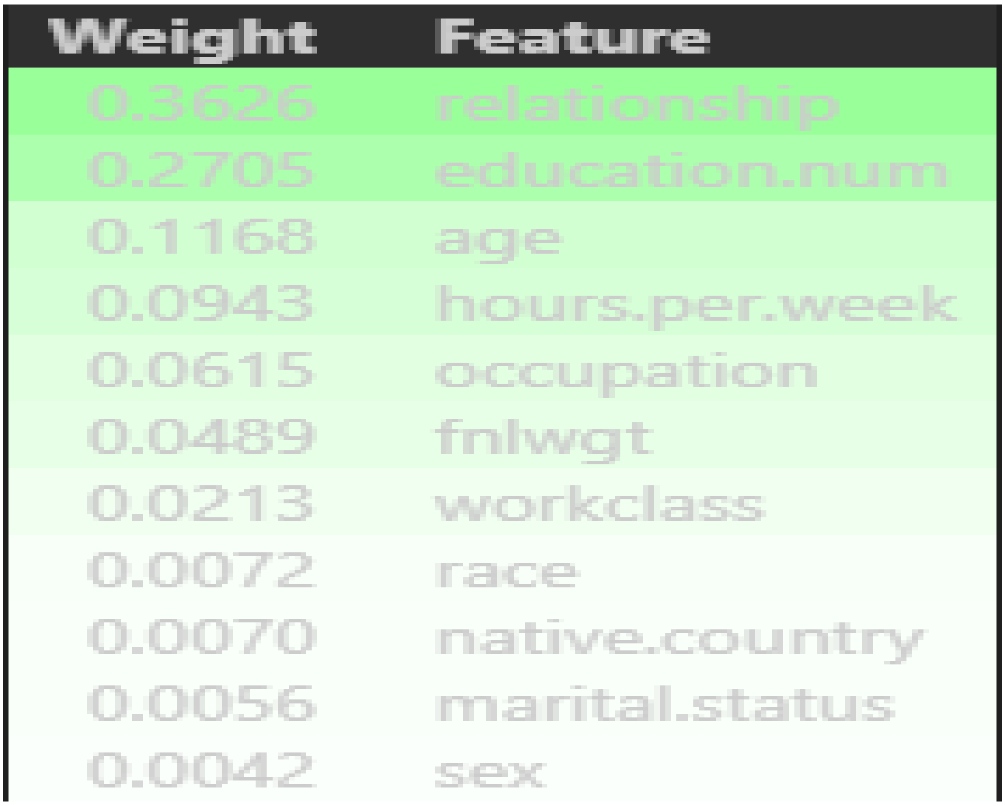
Hour per week:



Native country:

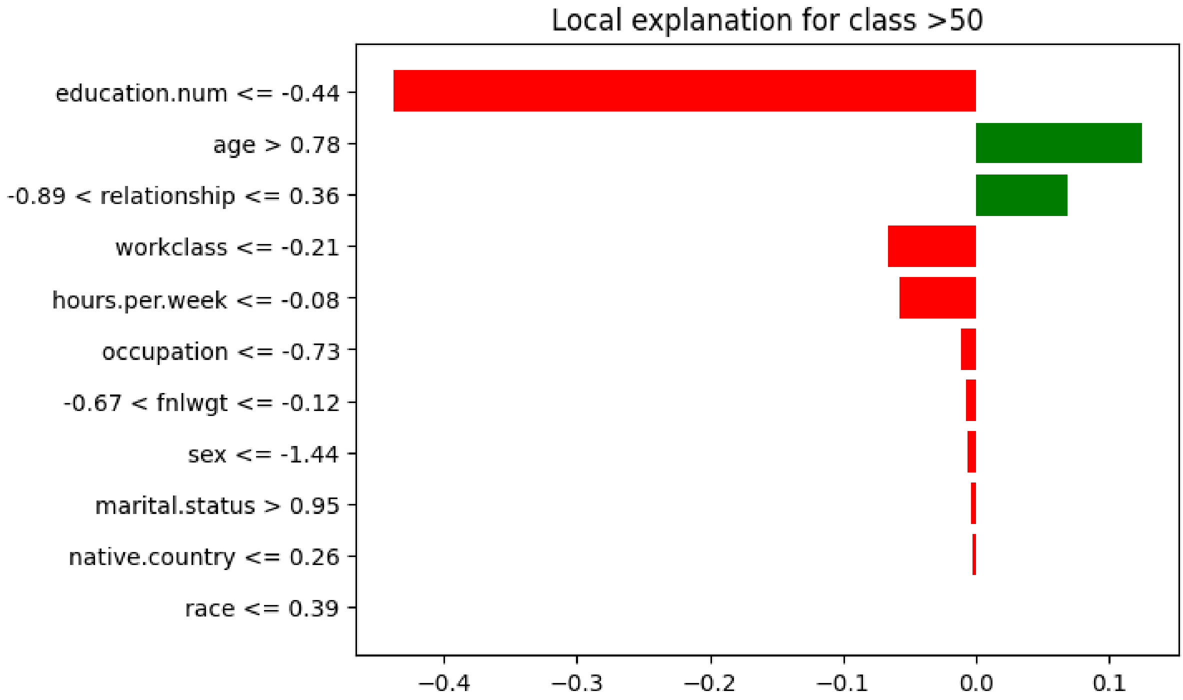


Eli5 plot:

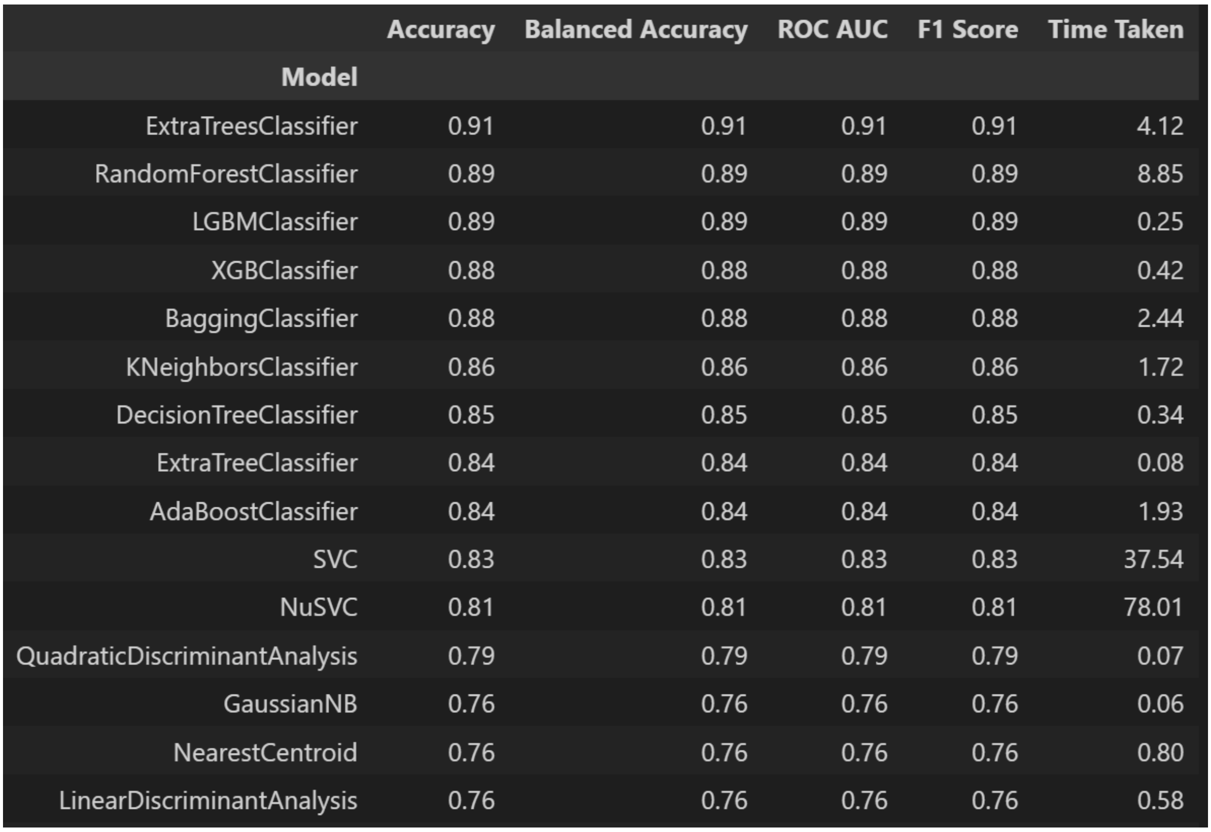


Weight	Feature
0.3626	relationship
0.2705	education.num
0.1168	age
0.0943	hours.per.week
0.0615	occupation
0.0489	fnlwgt
0.0213	workclass
0.0072	race
0.0070	native.country
0.0056	marital.status
0.0042	sex

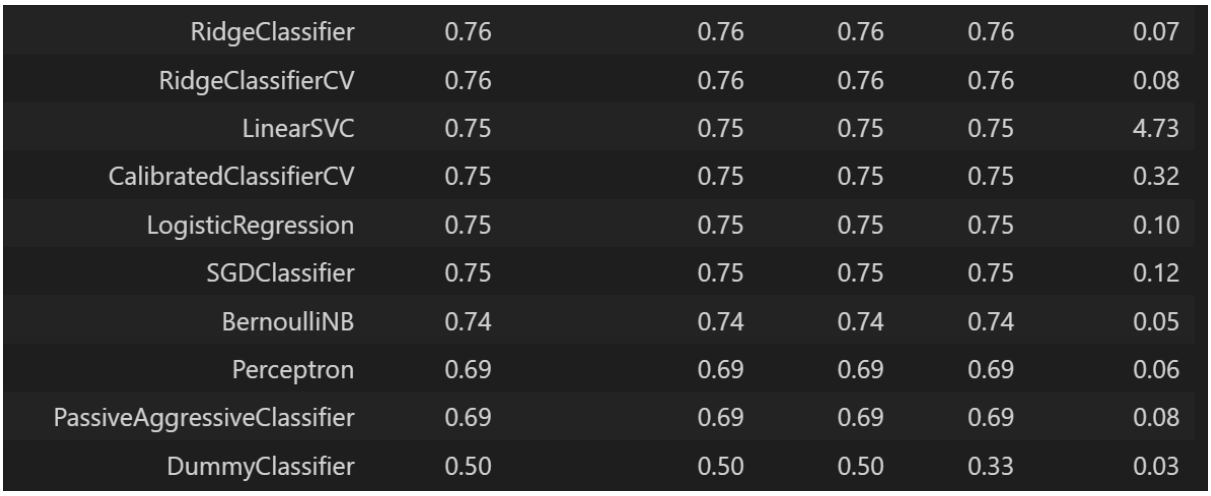
Lime plot:



Lazy classifier:

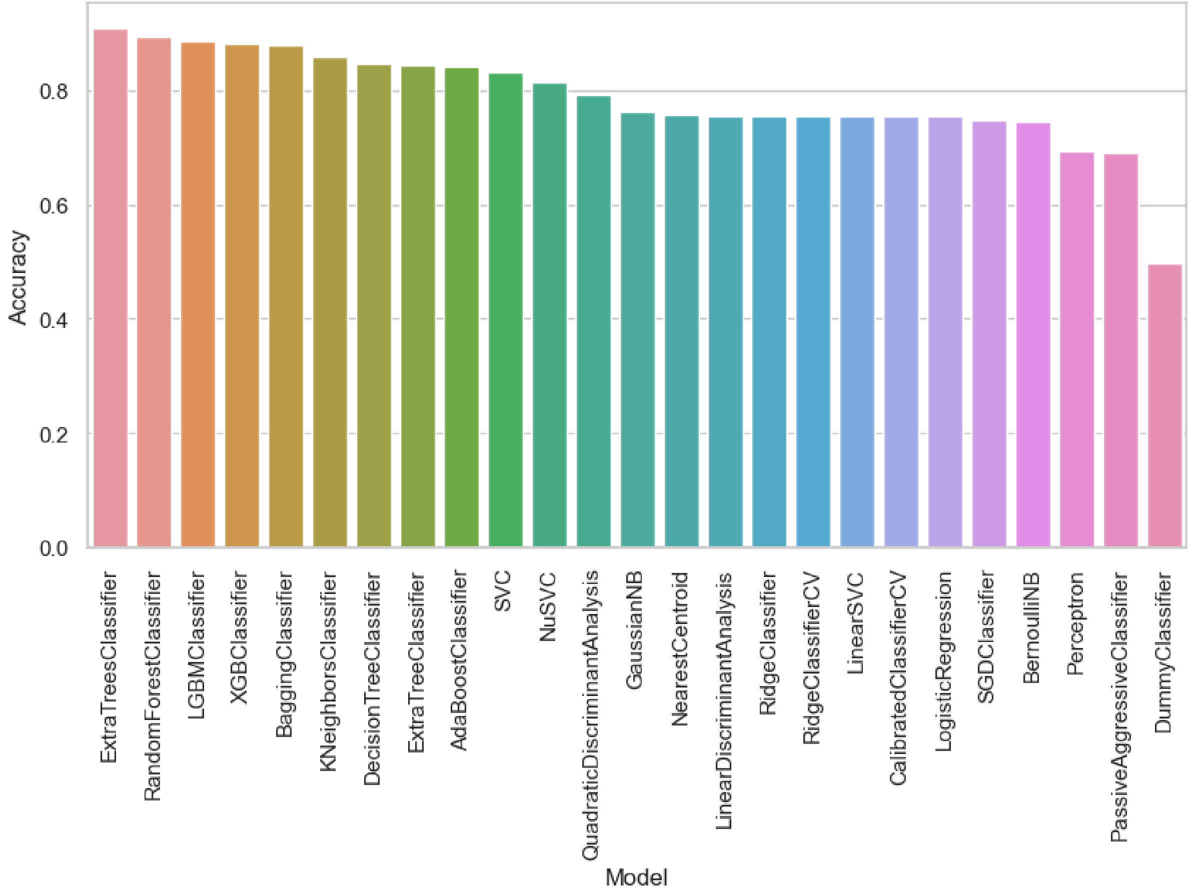


Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
ExtraTreesClassifier	0.91	0.91	0.91	0.91	4.12
RandomForestClassifier	0.89	0.89	0.89	0.89	8.85
LGBMClassifier	0.89	0.89	0.89	0.89	0.25
XGBClassifier	0.88	0.88	0.88	0.88	0.42
BaggingClassifier	0.88	0.88	0.88	0.88	2.44
KNeighborsClassifier	0.86	0.86	0.86	0.86	1.72
DecisionTreeClassifier	0.85	0.85	0.85	0.85	0.34
ExtraTreeClassifier	0.84	0.84	0.84	0.84	0.08
AdaBoostClassifier	0.84	0.84	0.84	0.84	1.93
SVC	0.83	0.83	0.83	0.83	37.54
NuSVC	0.81	0.81	0.81	0.81	78.01
QuadraticDiscriminantAnalysis	0.79	0.79	0.79	0.79	0.07
GaussianNB	0.76	0.76	0.76	0.76	0.06
NearestCentroid	0.76	0.76	0.76	0.76	0.80
LinearDiscriminantAnalysis	0.76	0.76	0.76	0.76	0.58



RidgeClassifier	0.76	0.76	0.76	0.76	0.07
RidgeClassifierCV	0.76	0.76	0.76	0.76	0.08
LinearSVC	0.75	0.75	0.75	0.75	4.73
CalibratedClassifierCV	0.75	0.75	0.75	0.75	0.32
LogisticRegression	0.75	0.75	0.75	0.75	0.10
SGDClassifier	0.75	0.75	0.75	0.75	0.12
BernoulliNB	0.74	0.74	0.74	0.74	0.05
Perceptron	0.69	0.69	0.69	0.69	0.06
PassiveAggressiveClassifier	0.69	0.69	0.69	0.69	0.08
DummyClassifier	0.50	0.50	0.50	0.33	0.03

Plot:



6.Discussion and Challenges:

6.1.Preprocessing:

Handling Missing Values: Strategies for imputation or removal, considering data patterns and model sensitivity.

Feature Scaling: Normalization or standardization for algorithms that require features on similar scales.

Feature Engineering: Creating new features from existing ones to enhance model performance (e.g., interaction terms, polynomial terms).

6.2.Model selection:

Considerations: Balancing accuracy, interpretability, computational efficiency, and resource constraints.

Interpretable Models: Decision trees, linear models, rule-based systems for understanding feature importance and decision logic.

Black-Box Models: Neural networks, ensemble methods for higher accuracy when interpretability is less critical.

Ensemble Methods: Combining multiple models for improved robustness and performance.

6.3.Specific Techniques:

LIME: Explains individual predictions for any model, revealing feature importance for specific instances.

SHAP: Quantifies feature importance globally and for individual predictions, identifying feature contributions and interactions.

SMOTETomek: Addresses class imbalance by oversampling minority class and undersampling majority class to enhance model fairness.

LazyPredict: Streamlines model building and comparison for quick prototyping and exploration of different algorithms.

7.CONCLUSION

This project unraveled the factors impacting income exceeding $50,000, leveraging a diverse dataset and powerful XAI techniques. Using a decision tree model, we peered into its decision-making, revealing key insights:

Age, education, occupation, hours worked, and marital status emerged as top influencers. LIME, SHAP, and eli5 pinpointed these, while SHAP further unveiled hidden interactions (e.g., education influencing income differently based on occupation).

LIME provided case-specific explanations, making the model transparent for individual predictions. LazyClassifier compared it to other algorithms, hinting at potential performance improvements.

This project showcased the power of XAI in deciphering and refining income prediction models. By illuminating key factors and potential biases, we pave the way for responsible AI development that empowers individuals and informs policies.

8.REFERENCE

Explainable Ai (XAI): Concepts and Techniques:
https://arxiv.org/abs/2107.07045 - A comprehensive paper defining XAI, exploring different techniques, and discussing research challenges.

DARPA XAI Program:
https://www.darpa.mil/program/explainable-artificial-intelligence - Learn about the DARPA XAI program's initiatives and funded projects, showcasing cutting-edge research in the field.

MIT XAI Website:
https://www.ll.mit.edu/r-d/projects/explainable-artificial-intelligence-decision-support - Explore resources from MIT's XAI initiative, including tutorials, workshops, and research papers.

LIME Documentation:
https://christophm.github.io/interpretable-ml-book/lime.html - Official documentation for LIME, showcasing its functionalities and providing tutorials for different use cases.

SHAP Website: https://shap.readthedocs.io/en/latest/api.html - Learn about SHAP and its various implementations, with interactive examples and explanations.

Eli5 Documentation: https://eli5.readthedocs.io/en/latest/overview.html - Explore Eli5's documentation for explaining model predictions in plain English, including code examples and best practices.