## Artificial Intelligence

## and Machine Learning

Project Report

Semester-IV (Batch-2022)

STOCK PREDICTION

A red and white sign

Description automatically generated with low confidence

**Submitted To: Submitted By:**

Ms. Shagun Sharma Akshit Nanda, 2210990093 (G9)

Gurnoor Kaur, 2210990345 (G9)

Harsh Wadhwa, 2210990382 (G9)

**Department of Computer Science and Engineering**

## Chitkara University Institute of Engineering & Technology,

## Chitkara University, Punjab

**INDEX**

|  |  |  |
| --- | --- | --- |
| Serial No. | Title | Page No. |
| 1 | Introduction | 1-6 |
| 2 | Problem Definition and Requirements | 6 |
| 3 | Materials and Methods | 8-14 |
| 4 | Data Analysis and Methodology | 14-35 |
| 5 | Evaluation Metrics | 35-37 |
| 6 | Result | 38 |

1. **Introduction:**

The initial stage in constructing a stock prediction model involves data aggregation. This encompasses compiling historical stock prices, trading volumes, financial statements, economic indicators, sentiment analysis from news sources, and other pertinent data reservoirs. The comprehensiveness of data directly influences the model's learning capacity and predictive accuracy. Post data aggregation, a preprocessing phase ensues, purging noise, handling missing data, and normalizing features. This ensures that the data is suitably formatted for algorithmic training. Feature engineering might also be implemented to distill salient information from raw data and concoct new features to augment the model's predictive efficacy.

Subsequently, the dataset is partitioned into training, validation, and test sets. The training set is utilized to educate the model, the validation set refines hyperparameters and guards against overfitting, while the test set gauges the model's performance on unseen data.

****

* 1. **Background**

Data Collection: The first step in building a stock prediction model is collecting historical and real-time data relevant to the stock market. This data may include stock prices, trading volumes, financial statements, economic indicators, news sentiment, and other factors that could impact stock prices. A background process continuously collects and updates this data from various sources such as financial databases, news feeds, and online APIs.

Data Preprocessing: Once the data is collected, it undergoes preprocessing to clean, normalize, and transform it into a format suitable for training the ML model. This preprocessing stage may involve handling missing values, removing outliers, scaling features, and encoding categorical variables. A background process handles these data preprocessing tasks autonomously, ensuring that the data is consistently prepared for model training.

Model Training and Optimization: The ML model used for stock prediction is trained on historical data to learn patterns and relationships that can help forecast future stock prices. This training process involves selecting the appropriate algorithm, tuning hyperparameters, and optimizing the model's performance. A background process iteratively trains and refines the model using updated data and feedback mechanisms to improve its predictive accuracy over time.

Model Evaluation: After training, the model is evaluated using validation data to assess its performance and identify any potential issues such as overfitting or underfitting. A background process automates the model evaluation process, running various metrics and statistical tests to quantify the model's predictive capabilities and identify areas for improvement.

Deployment and Monitoring: Once the model is deemed satisfactory, it is deployed into production for making real-time predictions. A background process manages the deployment process, ensuring seamless integration with existing systems and monitoring the model's performance in production. This includes tracking prediction accuracy, detecting drift in data distributions, and triggering retraining when necessary to maintain model relevance.



* 1. **Objectives:**

Prediction Accuracy: The foremost objective is to develop AI/ML models that can accurately predict future stock prices. This involves training models to identify patterns and trends in historical data and use them to make precise predictions about future market movements. The ultimate aim is to minimize prediction errors and maximize the reliability of forecasts.

Risk Management: Another key objective is to assist investors in managing risks associated with stock market investments. AI/ML models can analyze market volatility, correlations between different assets, and other risk factors to provide insights into potential risks and help investors make informed decisions to mitigate them.

Portfolio Optimization: AI/ML-based stock prediction models aim to optimize investment portfolios by identifying opportunities for diversification, maximizing returns, and minimizing risks. These models can suggest optimal asset allocations based on market conditions, investment goals, and risk preferences.

Real-Time Decision Making: One of the objectives is to enable real-time decision-making for investors by providing up-to-date predictions and insights. AI/ML models can analyze streaming data from various sources, such as news feeds, social media, and market events, to provide timely recommendations and alerts to investors.

Adaptability: Stock markets are dynamic and constantly evolving, so an essential objective is to develop AI/ML models that can adapt to changing market conditions. This involves building models that can learn from new data, adjust their predictions accordingly, and continuously improve their performance over time.

Scalability and Efficiency: AI/ML models for stock prediction should be scalable and efficient, capable of processing large volumes of data in a timely manner. This involves optimizing algorithms, leveraging parallel computing techniques, and deploying models on scalable infrastructure to handle increasing data loads and user demands.

**1.3 Significance:**

Risk Mitigation: By leveraging historical data and real-time market indicators, AI/ML algorithms can assess various risk factors associated with stock investments. This allows investors to better manage risks, hedge against potential losses, and construct more resilient portfolios.

Efficiency and Automation: AI/ML algorithms automate many aspects of stock analysis and prediction, significantly reducing the time and effort required by human analysts. This automation streamlines decision-making processes, increases productivity, and frees up human capital for more strategic tasks.

Innovation in Investment Strategies: The application of AI/ML in stock prediction fosters innovation in investment strategies and techniques. It opens up avenues for exploring novel approaches to portfolio optimization, risk management, and algorithmic trading, driving advancements in financial analysis and decision-making.

**Target Users:**

The survey form is distributed to various categories of the users such as General Public including students, The Doctors, and Mental Health professionals and Information Technology Professionals to gain insights for future improvements as well as obtaining new reflections and know the best value use case of the system. The primary users would be the Doctors and Mental health professionals who can use for automating their productivity. The Secondary users are the General Public including students who can gather facts about their current mental state and the third category includes the Information technology professionals who can refer to this system as for learning and optimizing purposes. The data will be analyzed, and improvements will be made based on their feedback and opinions. The Questionnaire is to prepare among 50 persons who are interested in Computational Psychiatry.

1. **Problem Definition and Requirements:**

The goal of this project is to develop a stock prediction model using machine learning that can accurately forecast the future stock prices of Microsoft based on its historical data. The model should be able to learn patterns and trends in the data and make predictions accordingly.

Requirements:

Data Collection: Collect historical stock data of Microsoft from a reliable source, such as Yahoo Finance or Quandl.

Data Pre-processing: Clean and pre-process the data by handling missing values, converting data types, and scaling the data.

Feature Engineering: Extract relevant features from the data, such as Open, High, Low, and Volume, that can be used to train the model.

Model Selection: Choose a suitable machine learning algorithm, such as LSTM, that can handle time-series data and make accurate predictions.

Model Training: Train the model using the preprocessed data and evaluate its performance using metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Model Evaluation: Evaluate the performance of the model using a test dataset and compare the predicted values with the actual values.

Model Deployment: Deploy the trained model in a production-ready environment, such as a web application or a mobile app, where it can be used to make predictions on new, unseen data.

1. **Materials and Methods:**
   1. The materials and methods for artificial intelligence (AI) and machine learning (ML) encompass a wide array of tools, techniques, algorithms, and frameworks. Here's a breakdown:

Data: The fuel for AI and ML systems. This includes structured data (like databases), unstructured data (like text, images, and videos), and semi-structured data (like JSON).

Hardware: This ranges from CPUs to GPUs to TPUs (Tensor Processing Units) for training and inference tasks. Additionally, specialized hardware like FPGA (Field-Programmable Gate Arrays) and ASICs (Application-Specific Integrated Circuits) are becoming more prevalent for accelerating AI tasks.

Software: Various software tools and frameworks are used for developing AI and ML models. This includes programming languages like Python and R, as well as libraries like TensorFlow, PyTorch, scikit-learn, and Keras.

Cloud Services: Many AI and ML projects utilize cloud computing platforms like AWS, Google Cloud Platform, and Azure for scalable compute resources and managed services.

Pre-trained Models: Pre-trained models, such as BERT for natural language processing or ResNet for image recognition, are often used as starting points for specific tasks to leverage transfer learning.

Methods:

Supervised Learning: In this approach, the algorithm learns from labeled data, where each training example is paired with an input and a corresponding output.

Unsupervised Learning: Here, the algorithm learns patterns from unlabeled data, finding hidden structures or intrinsic groupings within the data.

Semi-Supervised Learning: A combination of supervised and unsupervised learning, where the model learns from a small amount of labeled data and a large amount of unlabeled data.

Reinforcement Learning: This involves training an agent to interact with an environment to achieve a goal, receiving feedback in the form of rewards or penalties.

Transfer Learning: The technique of applying knowledge from one domain or task to another related domain or task. This often involves fine-tuning pre-trained models on new data.

Neural Networks: These are the building blocks of many AI and ML models, inspired by the structure of the human brain. Common types include feedforward neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and transformers.

Optimization Algorithms: Techniques like gradient descent and its variants are used to minimize the loss function during model training.

Evaluation Metrics: Metrics like accuracy, precision, recall, F1 score, and area under the ROC curve (AUC-ROC) are used to evaluate the performance of AI and ML models.

Deployment Techniques: Methods for deploying trained models into production environments, such as containerization (using Docker), serverless computing, and edge computing.

Ethical Considerations: With the increasing use of AI and ML in various applications, ethical considerations such as fairness, transparency, privacy, and accountability are crucial throughout the development and deployment process.

* 1. **Handling Missing Values and Relabelling Categorical Features**

Identify Missing Values: First, identify which columns or features in your dataset contain missing values.

Choose Imputation Strategy: Decide on an appropriate by dropping null values based on the nature of your data and the extent of missingness. For stock data, common strategies might include mean or median for numerical features and mode for categorical features.

**Relabelling Categorical Features:**

Identify Categorical Features: Determine which features in your dataset are categorical. In stock data, categorical features might include sectors, industries, or stock exchange symbols.

Choose Encoding Method: Select an encoding method based on the nature of your categorical features and the requirements of your predictive model. For example, one-hot encoding might be suitable for sectors or industries, while label encoding could be used for stock exchange symbols.

Apply Encoding: Implement the chosen encoding method to convert categorical features into a format that can be effectively used by machine learning algorithms.

* 1. **Removing Irrelevant or Redundant Features**

Feature Importance Analysis:

Before removing any features, conduct a feature importance analysis to understand which features contribute the most to predicting stock prices. Techniques like:

Feature Importance Scores: Use algorithms like Random Forest or Gradient Boosting to calculate feature importance scores. Features with lower importance scores can be considered for removal.

Correlation Analysis: Analyze the correlation between features and the target variable (e.g., stock prices). Features with low correlation coefficients may be candidates for removal.

2. Dimensionality Reduction Techniques:

If your dataset has a large number of features, consider applying dimensionality reduction techniques to reduce the number of features while preserving the most relevant information:

Principal Component Analysis (PCA): PCA identifies the principal components (linear combinations of features) that capture the maximum variance in the data. You can retain a subset of these components to represent the data.

Linear Discriminant Analysis (LDA): LDA finds the linear combinations of features that best separate different classes in the data. It can be useful for classification tasks in stock prediction.

3. Domain Knowledge:

Leverage domain knowledge to identify and remove irrelevant or redundant features:

Financial Ratios: Focus on financial ratios and indicators that are known to impact stock prices, such as price-to-earnings ratio (P/E ratio), earnings per share (EPS), or price-to-book ratio (P/B ratio).

Technical Indicators: Remove technical indicators that are not relevant to your prediction task or have low predictive power. Focus on indicators commonly used in stock analysis, such as moving averages, relative strength index (RSI), and Bollinger Bands.

4. Forward Selection or Backward Elimination:

Forward Selection: Start with an empty set of features and iteratively add features based on their impact on model performance until reaching a stopping criterion.

Backward Elimination: Begin with all features and iteratively remove the least important features based on a chosen criterion (e.g., p-values in linear regression or feature importance scores) until no further improvement in model performance is observed.

5. Regularization Techniques:

L1 Regularization (Lasso): Lasso regularization penalizes the absolute size of the coefficients, leading to sparse models where irrelevant features are automatically set to zero.

L2 Regularization (Ridge): Ridge regularization penalizes the squared size of the coefficients, reducing the impact of irrelevant features on model performance.

Several features were identified as irrelevant or redundant for the depression detection task and were removed from the dataset. The following features were dropped:

* Response time features (Q\*E)
* Question order features (Q\*I)
* Vocabulary check features (VCL\*)
* Other extraneous features like source, elapse, engnat, hand, orientation, voted, country, screensize, uniquenetworklocation.

This step aimed to simplify the dataset by retaining only the most informative and relevant features for predicting depression levels, potentially improving model performance and interpretability.

* 1. **Feature Engineering:**

1. Historical Prices and Returns:

Lagged Returns: Calculate lagged returns (e.g., one-day lag, weekly lag) to capture the historical price movements of stocks.

Moving Averages: Compute moving averages (e.g., simple moving average, exponential moving average) over different time periods to smooth out noise and identify trends.

Volatility Measures: Calculate measures of volatility such as standard deviation or average true range to capture the magnitude of price fluctuations.

2. Technical Indicators:

Relative Strength Index (RSI): Measure the speed and change of price movements to identify overbought or oversold conditions.

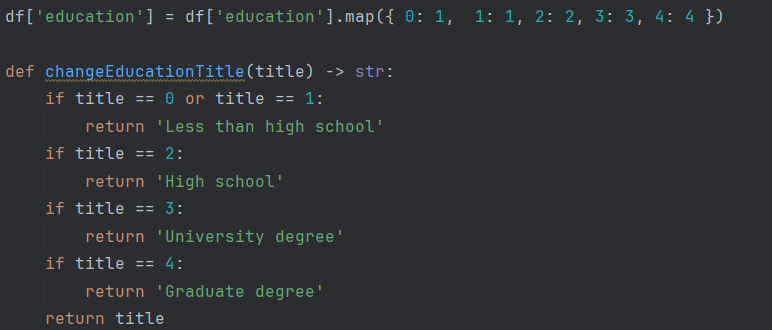
Moving Average Convergence Divergence (MACD): Detect changes in momentum by comparing short-term and long-term moving averages.

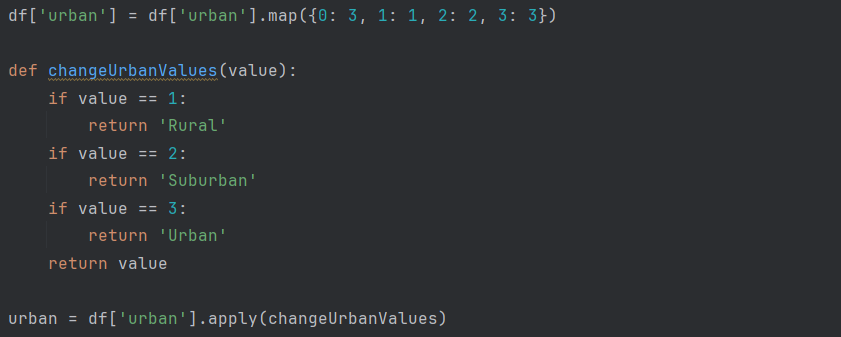
Bollinger Bands: Plot bands around moving averages to identify potential breakouts or reversals in price trends.

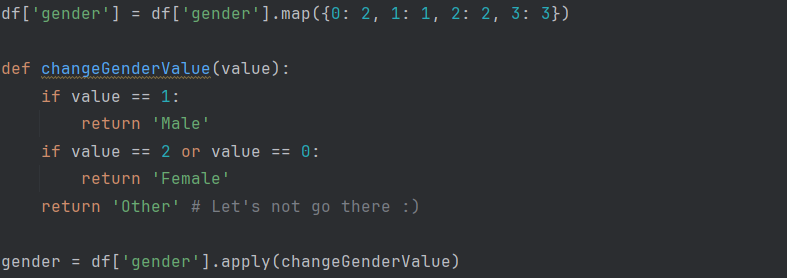
3. Fundamental Analysis:

Financial Ratios: Calculate key financial ratios such as price-to-earnings ratio (P/E ratio), earnings per share (EPS), and price-to-book ratio (P/B ratio) to assess the valuation of stocks.

Dividend Yield: Compute the dividend yield as a percentage of the stock's price to evaluate the income potential for investors.

****

****

****





1. **Data Analysis & Methodology:**

**Exploratory Data Analysis:**

It involves analyzing and visualizing data to understand its key characteristics, uncover patterns, and identify relationships between variables refers to the method of studying and exploring record sets to apprehend their predominant traits, discover patterns, locate outliers, and identify relationships between variables.

1. Data Collection and Exploration:

Data Sources: Gather historical stock data from reliable sources such as financial databases, APIs, or data providers.

Exploratory Data Analysis (EDA): Perform EDA to gain insights into the dataset, including summary statistics, data distributions, correlations, and missing values.

Visualization: Visualize key aspects of the data using plots and charts to identify patterns, trends, and anomalies.

2. Preprocessing and Feature Engineering:

Data Cleaning: Handle missing values, outliers, and inconsistencies in the dataset through imputation, filtering, or removal.

Feature Engineering: Create new features or transform existing ones to extract relevant information for predicting stock prices. This may involve technical indicators, fundamental ratios, sentiment scores, and market dynamics.

Normalization and Scaling: Normalize or scale features to ensure that they have comparable scales and distributions, which can improve the performance of machine learning models.

3. Model Selection and Evaluation:

Model Selection: Choose appropriate machine learning algorithms for stock prediction, considering factors such as interpretability, complexity, and performance. Common models include linear regression, time series models (e.g., ARIMA, LSTM), and ensemble methods (e.g., Random Forest, Gradient Boosting).

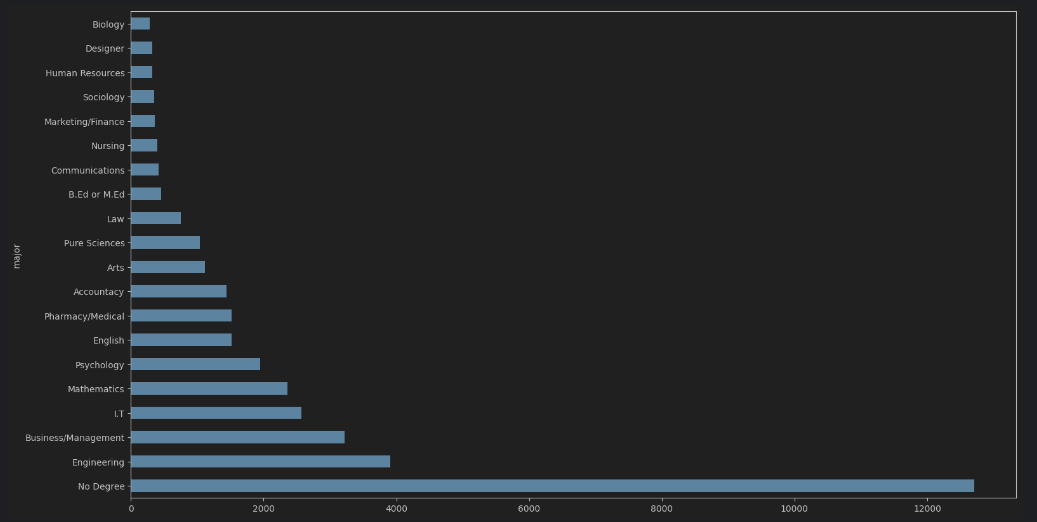
Evaluation Metrics: Define evaluation metrics to assess the performance of predictive models, such as mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), or accuracy.

Cross-Validation: Use techniques like k-fold cross-validation to assess the generalization performance of models and mitigate overfitting.

4. Risk Management and Strategy Development:

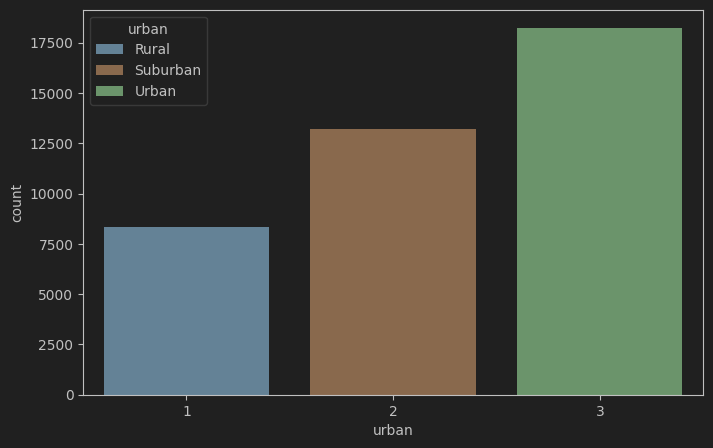
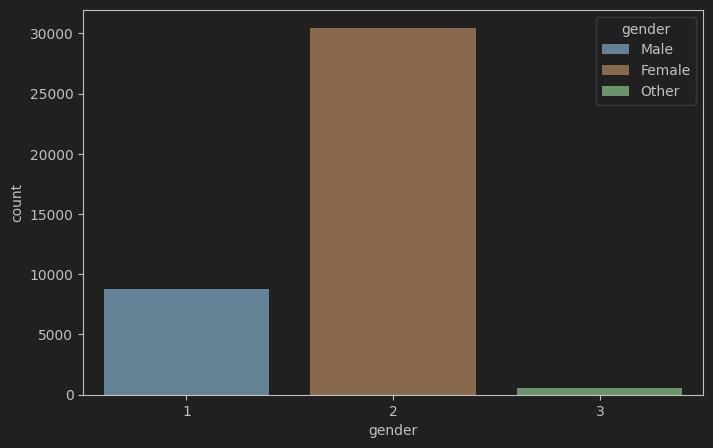
Risk Assessment: Consider risk factors such as volatility, drawdowns, and market conditions when developing trading strategies based on predictive models.

Portfolio Optimization: Incorporate portfolio optimization techniques to allocate assets based on the predictions of the models while considering risk-return trade-offs..



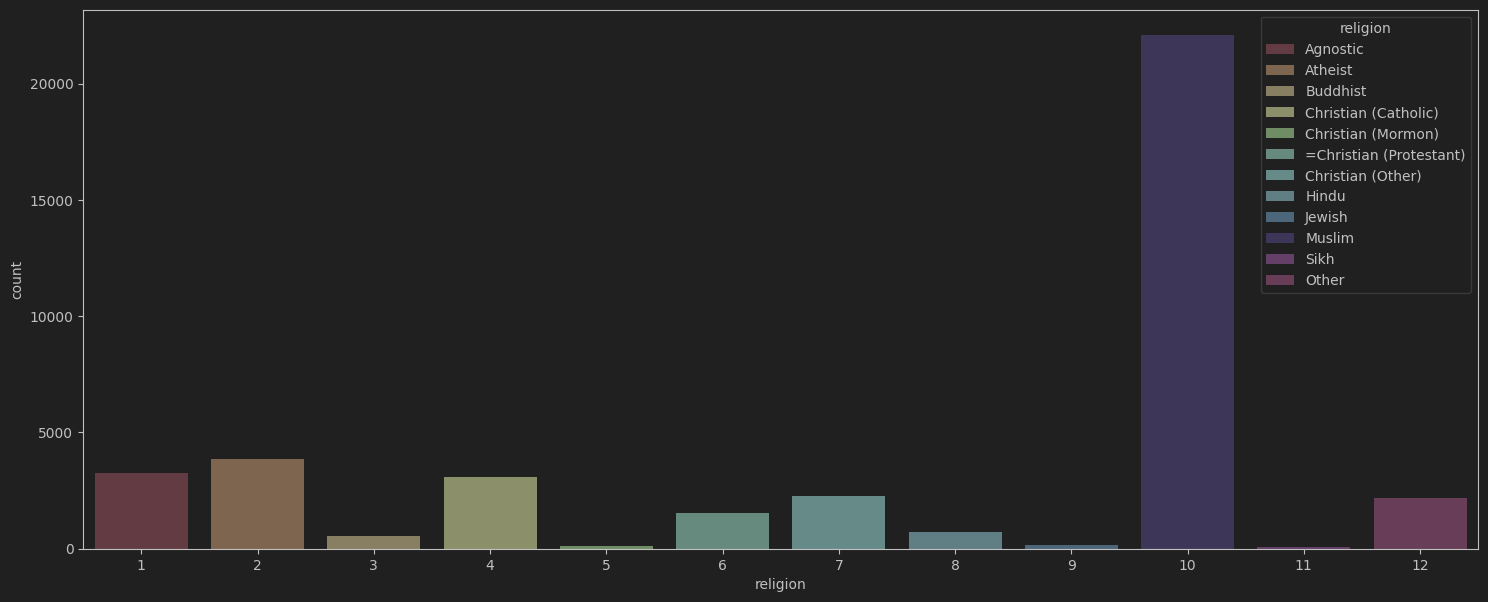
**Evaluation**

Once the model is trained, it needs to be evaluated to measure its performance. Evaluation involves calculating various metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared.

****

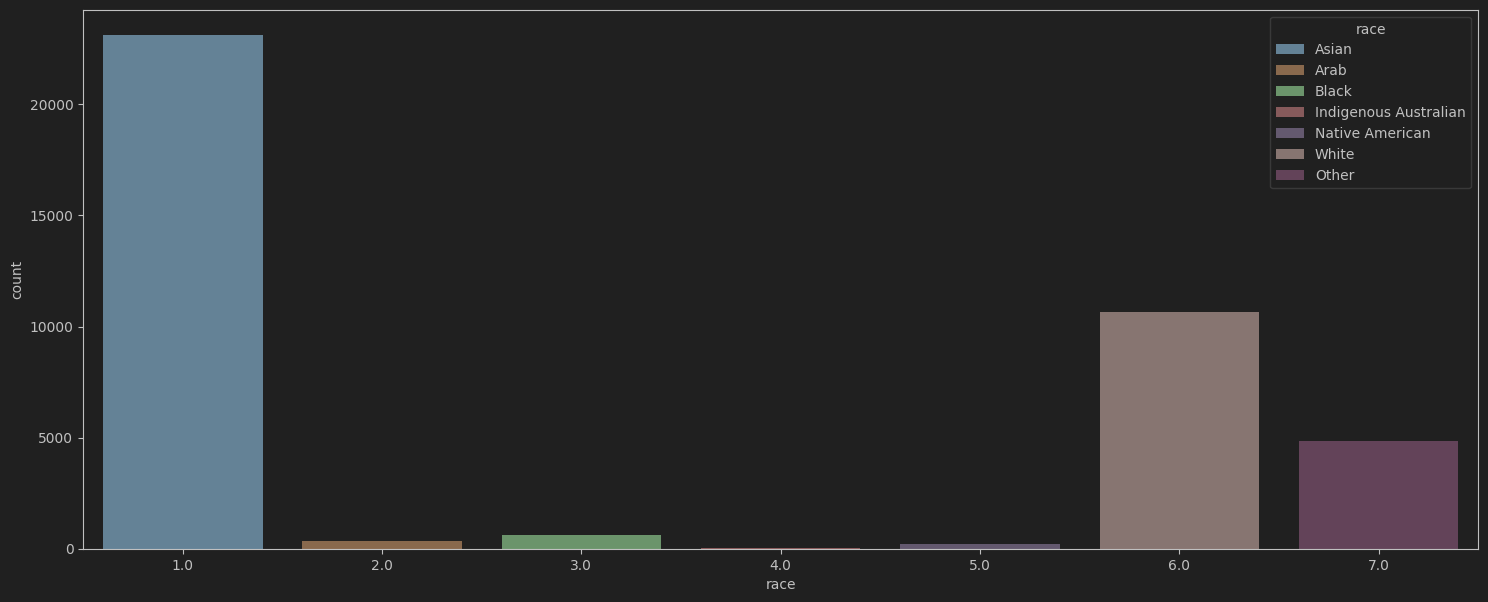
**Training**

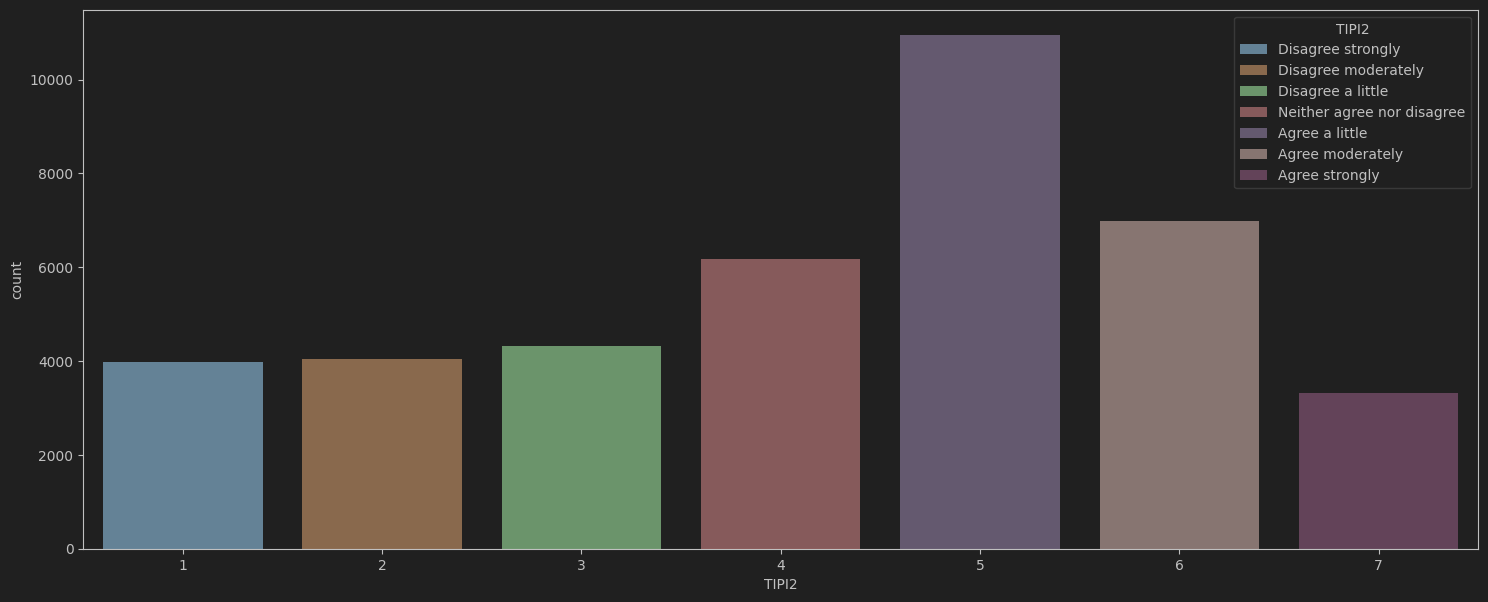
Once the model is selected, it needs to be trained on the preprocessed data. Training involves optimizing the model's parameters to minimize the error between the predicted and actual values. Training can be done using techniques such as gradient descent, stochastic gradient descent, and Adam

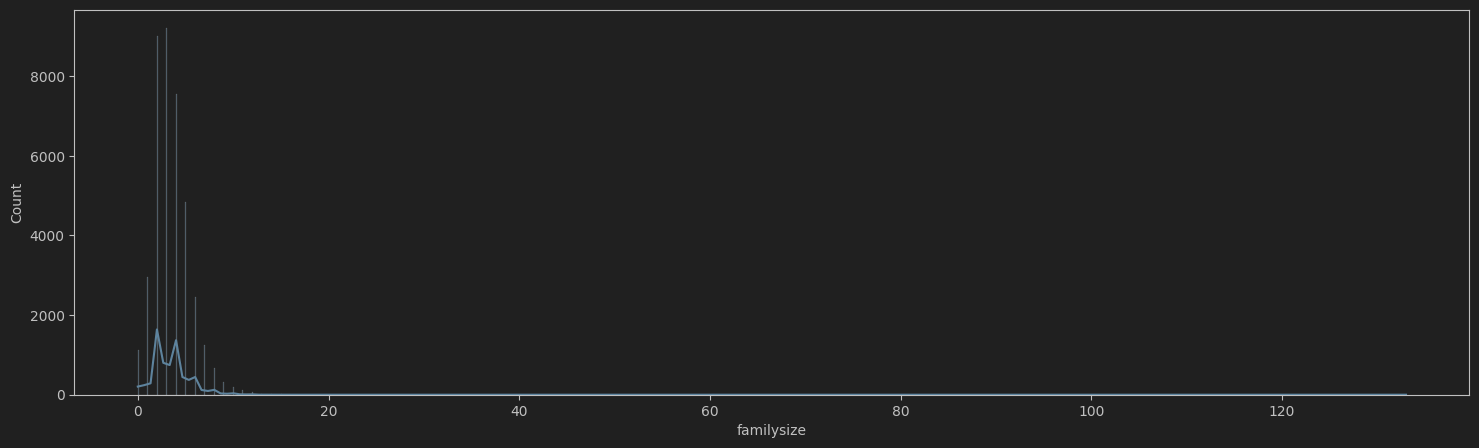
****

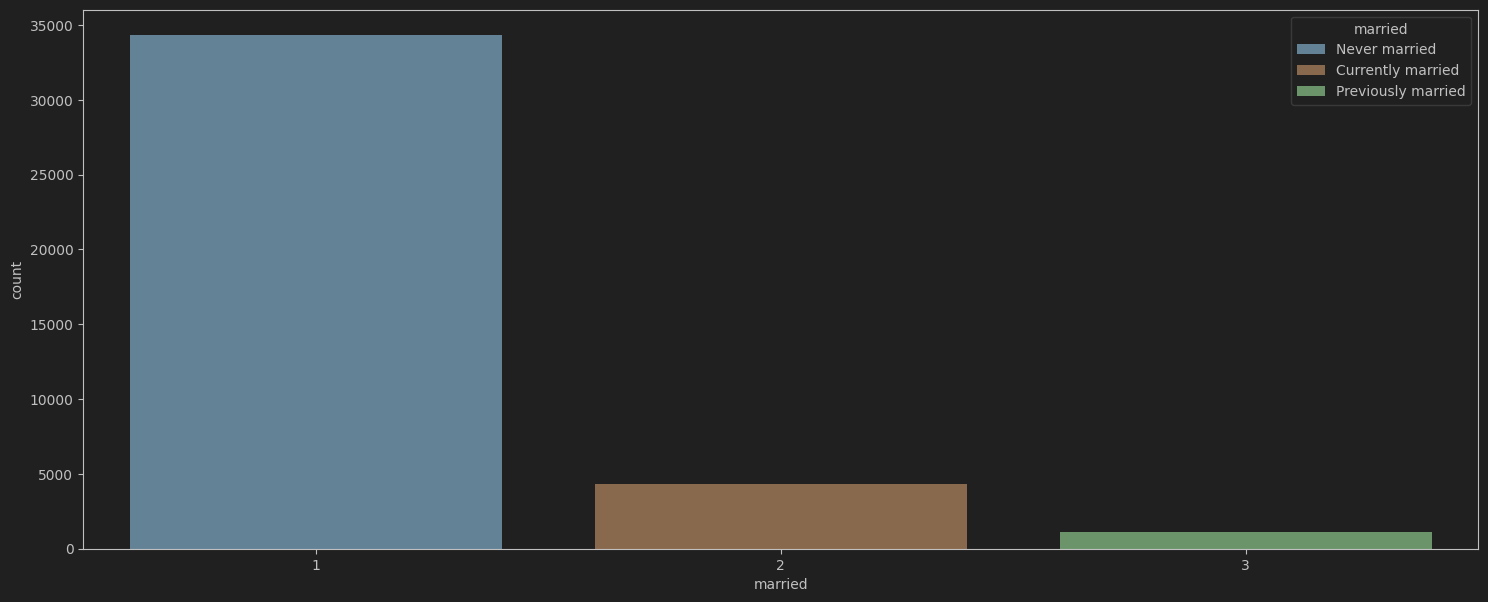
**Model Selection**

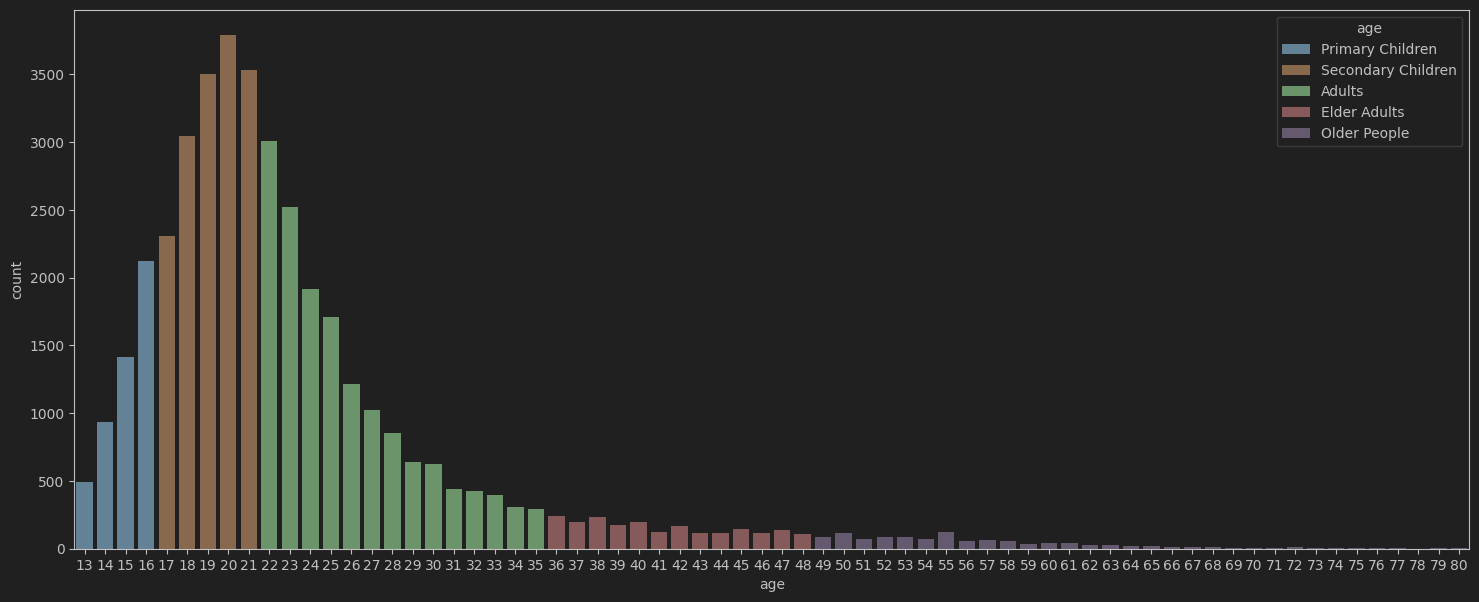
Once the data is preprocessed, the next step is to select a suitable model for stock prediction. There are various models used in stock prediction, including linear regression, decision trees, random forests, and neural networks. The choice of model depends on the data and the problem statement**.**

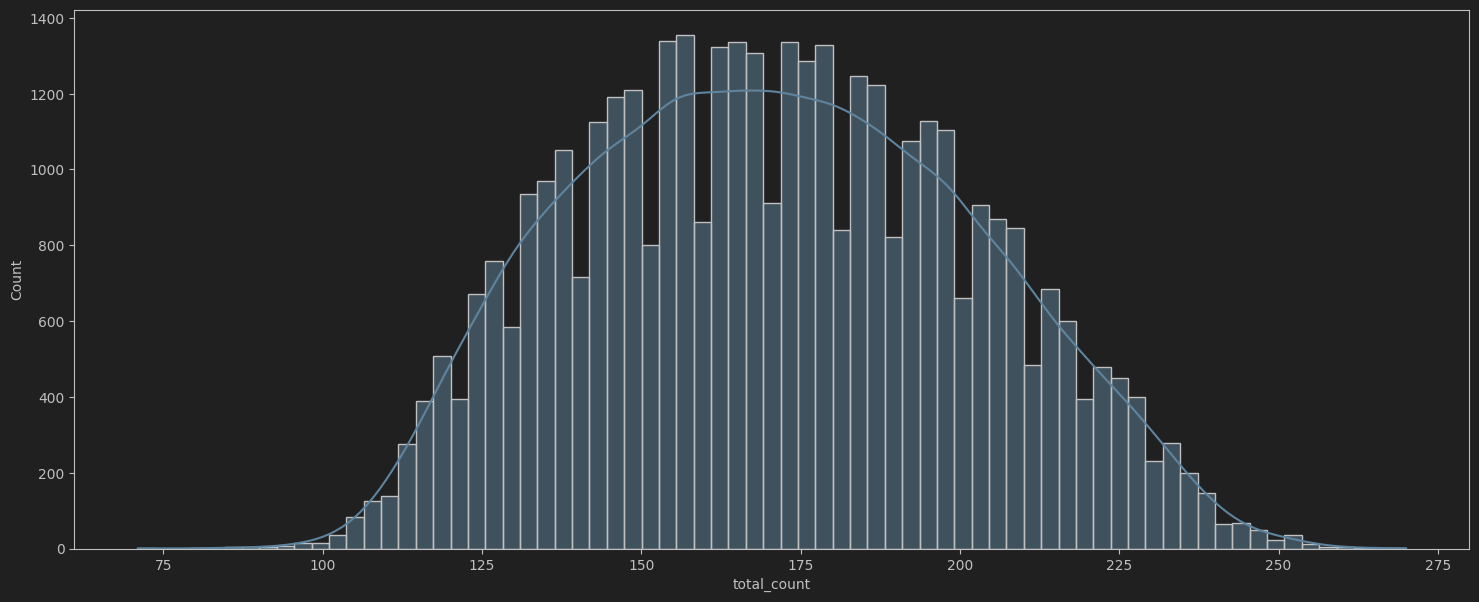


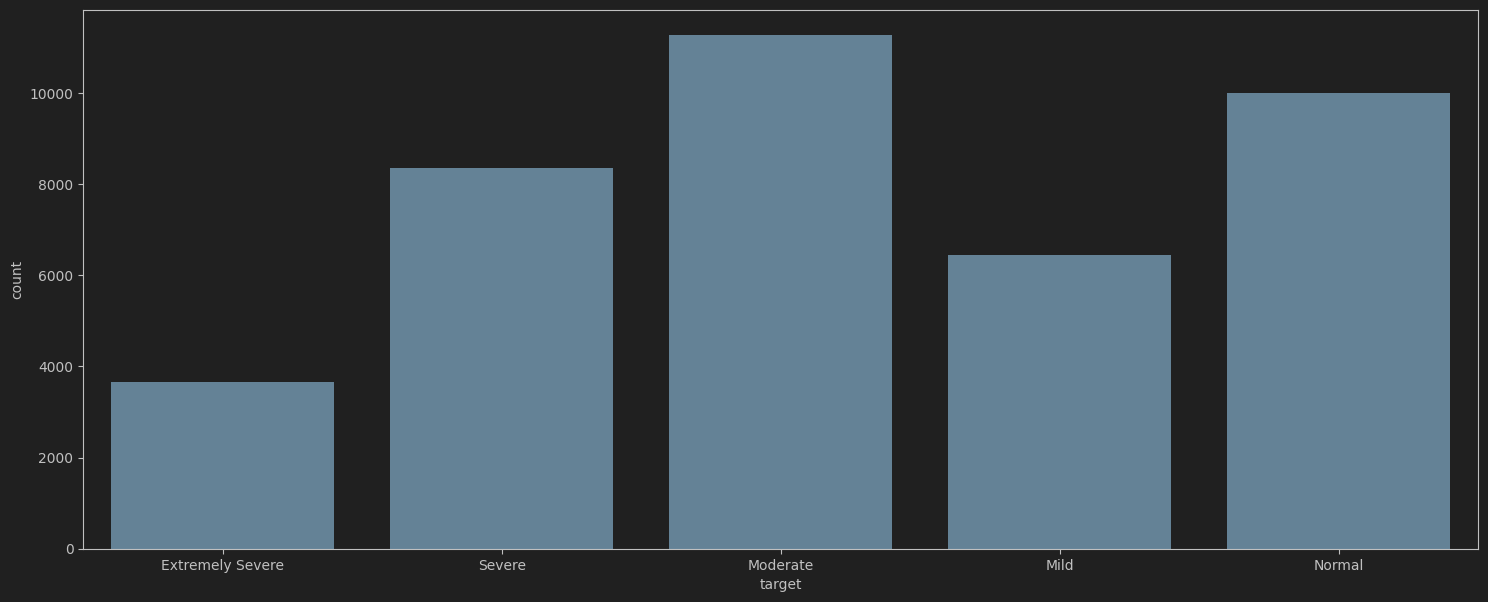


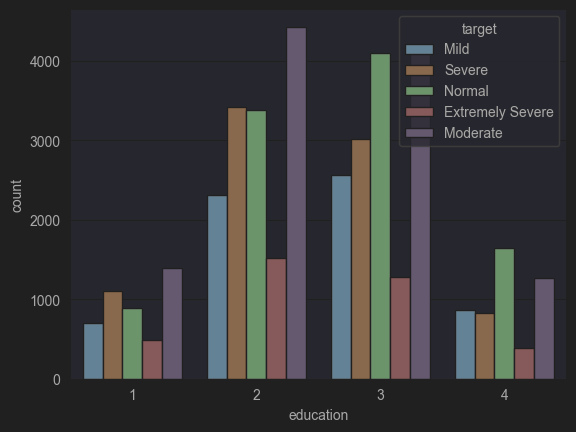


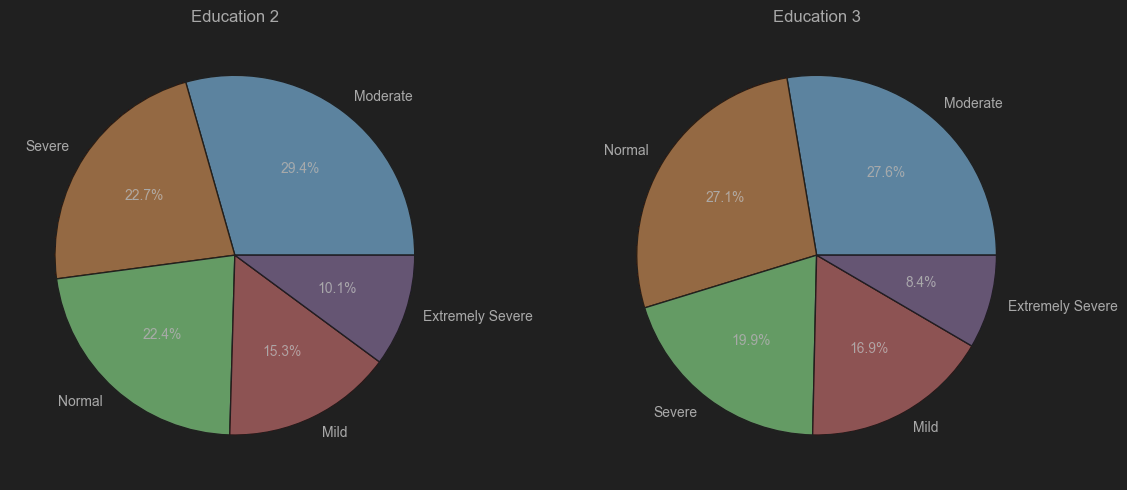


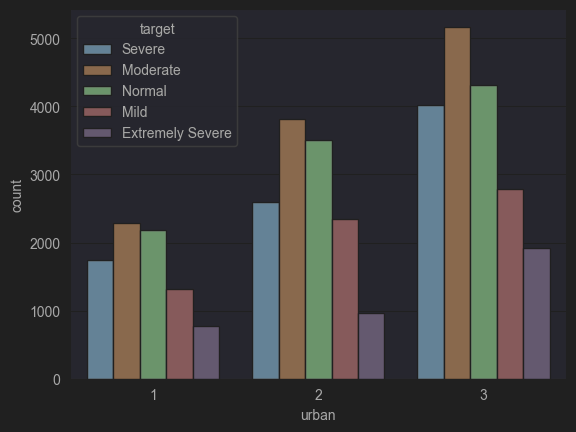


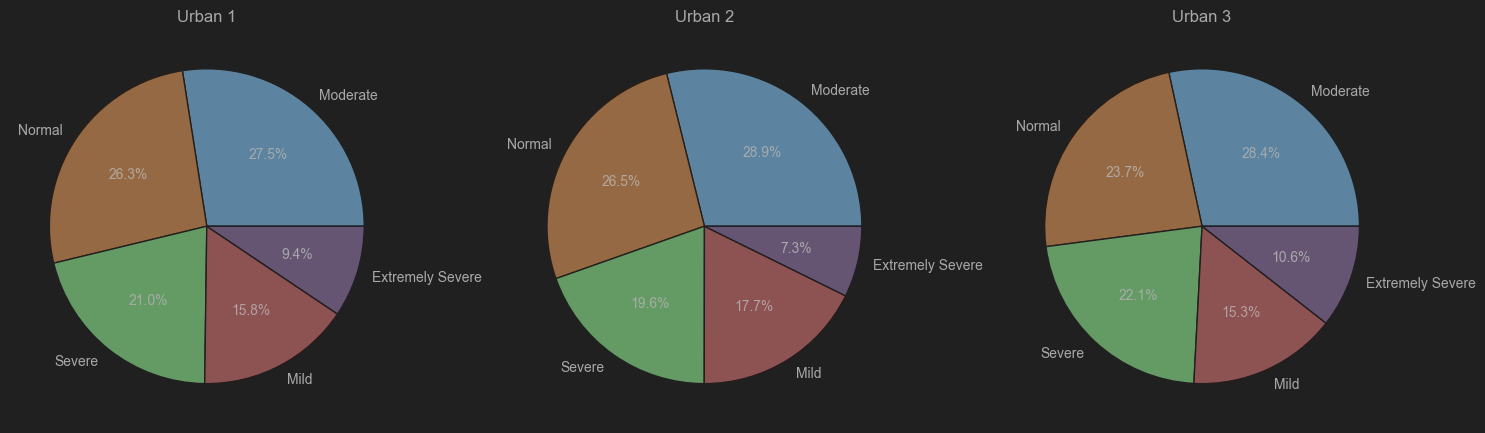


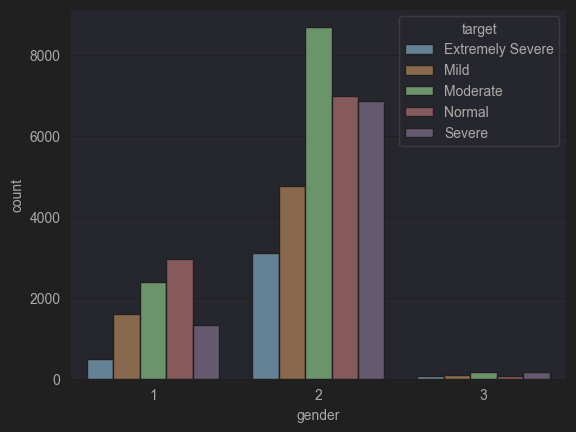


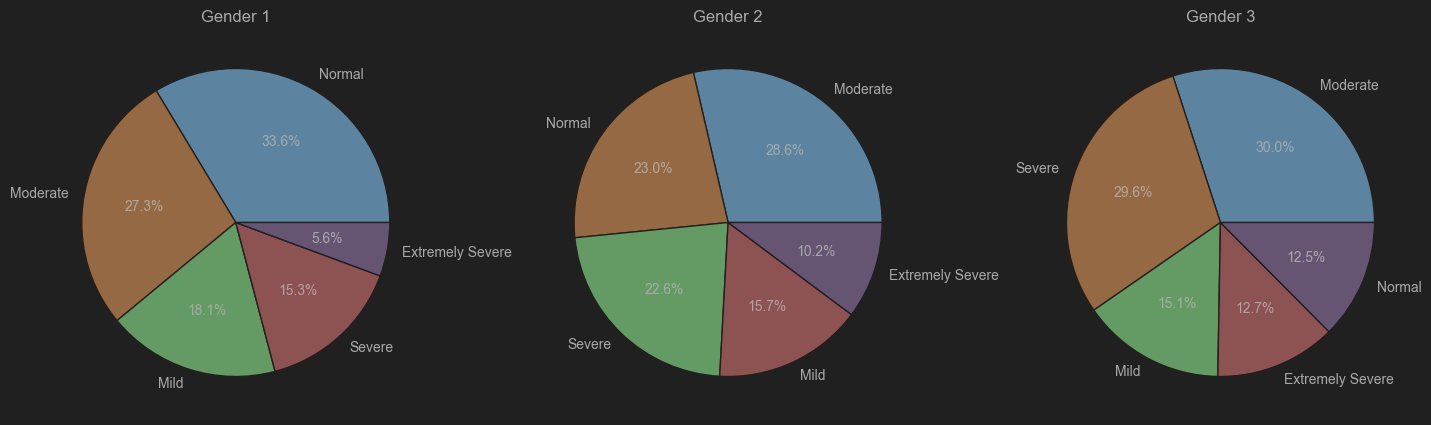
****

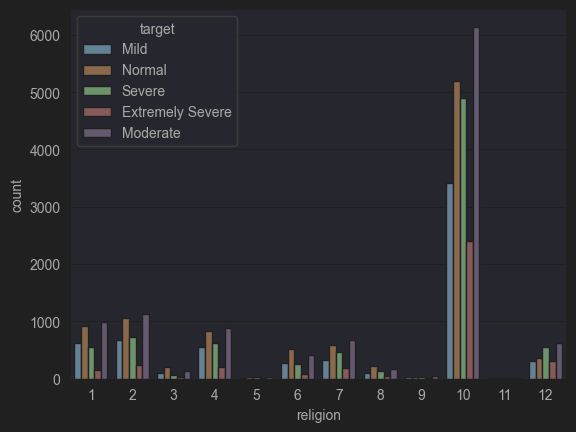
****

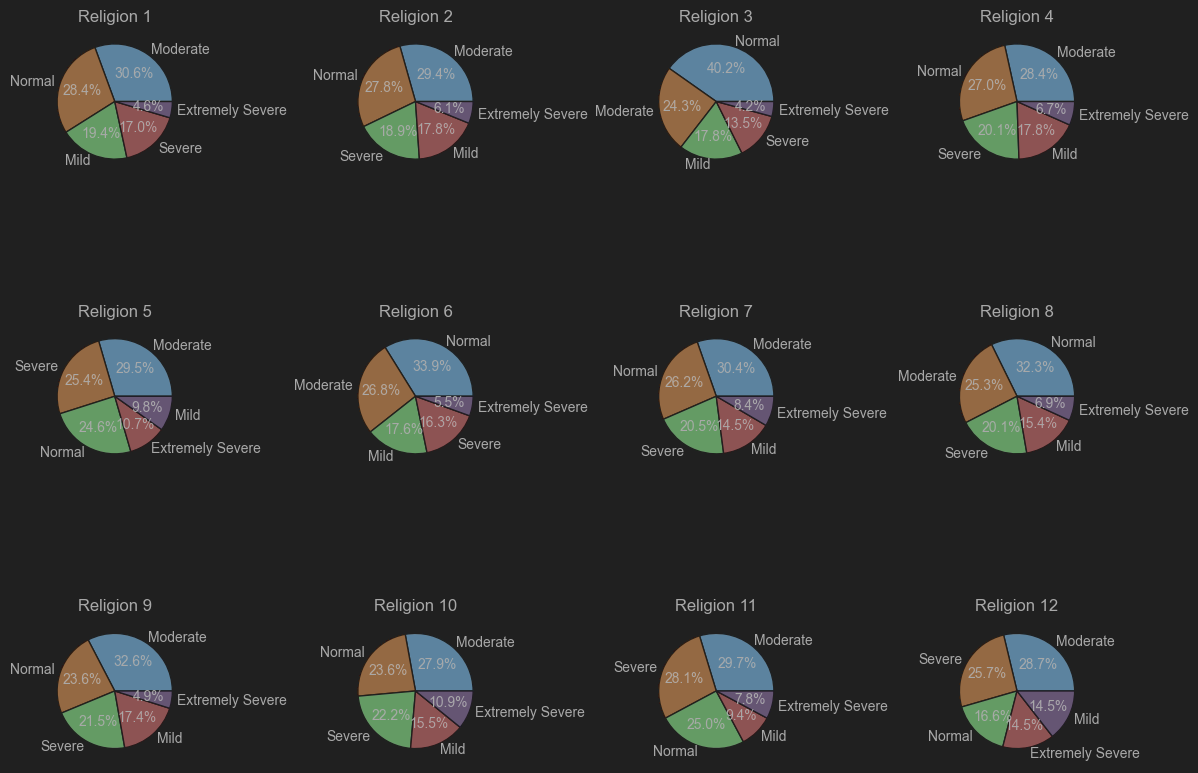
****

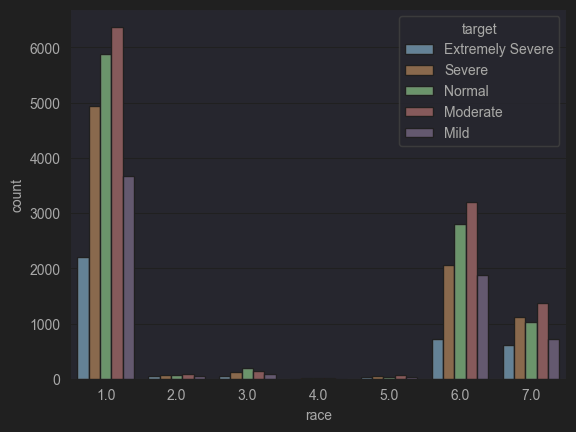
****

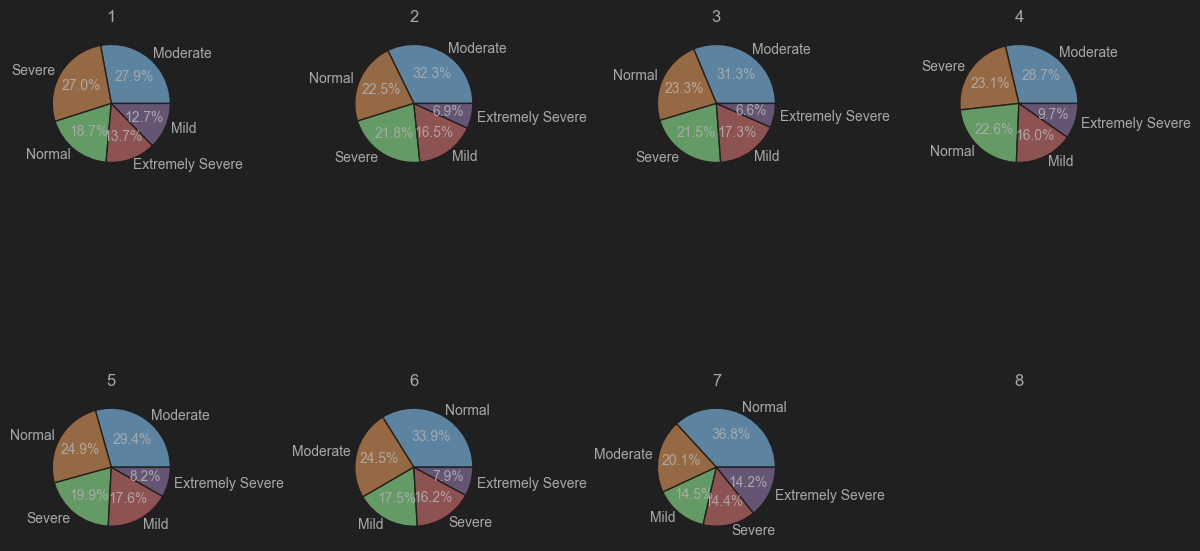
* 1. ****

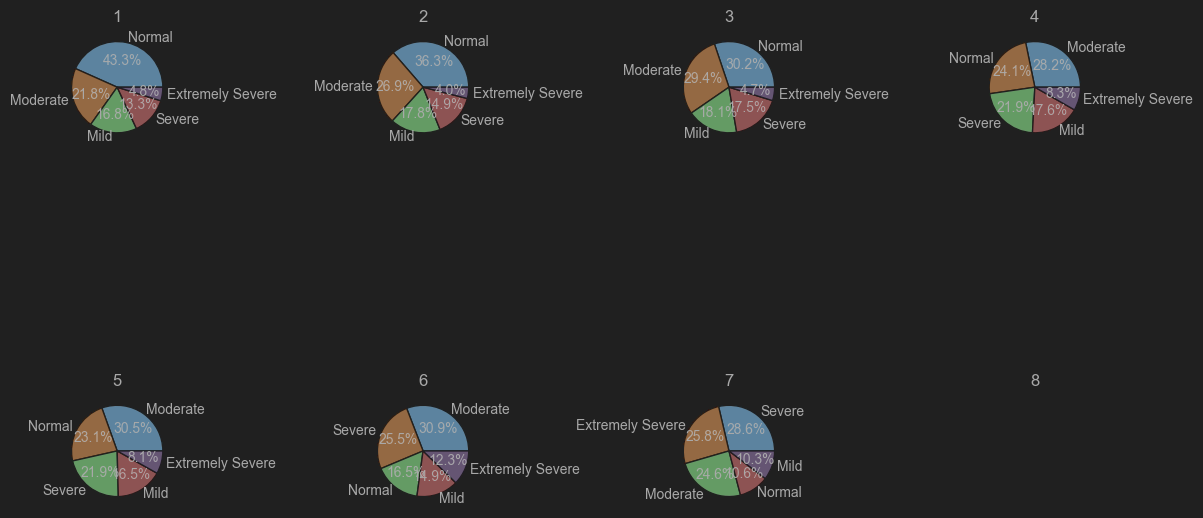
****

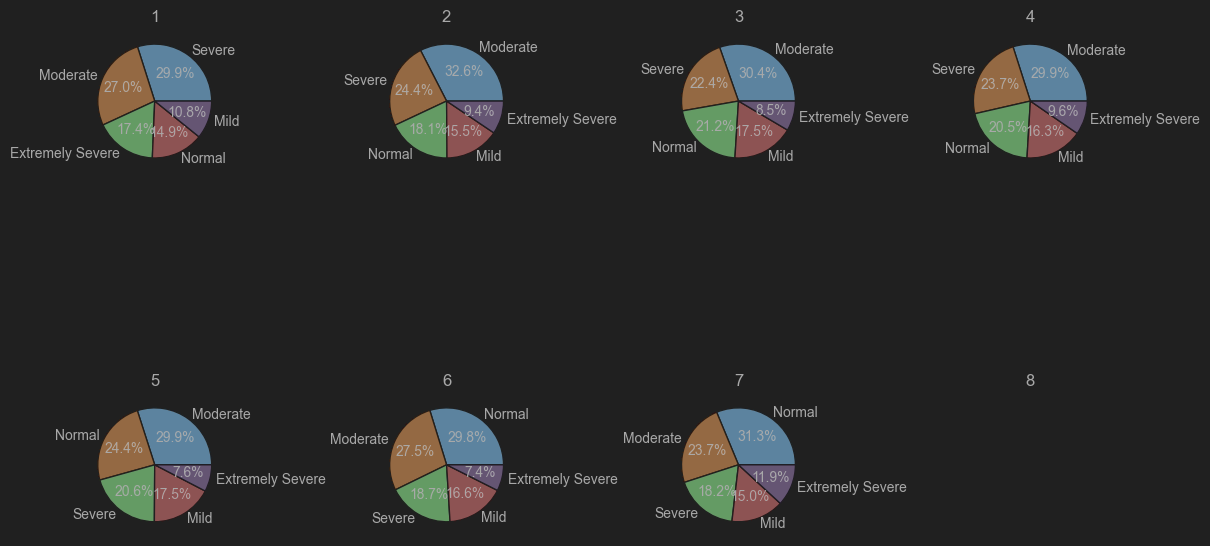
****

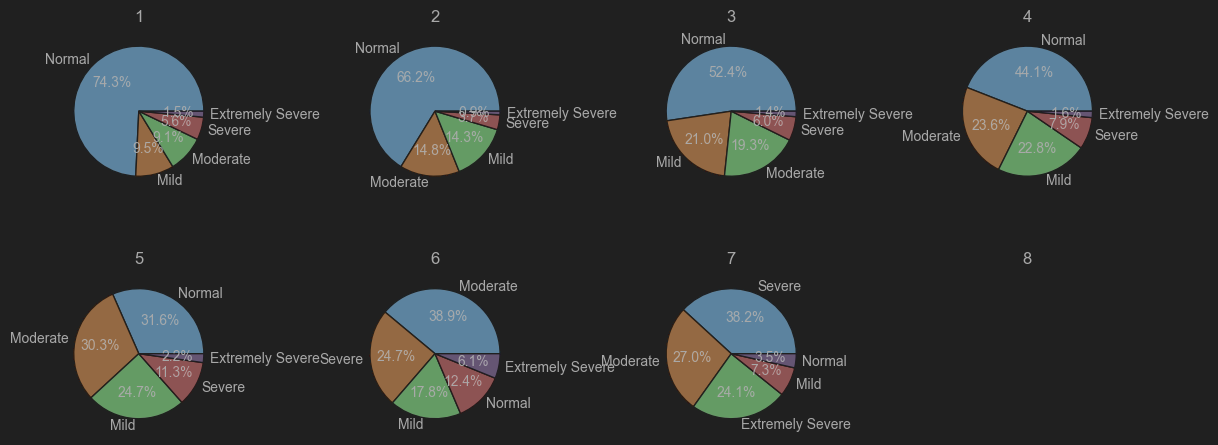
****

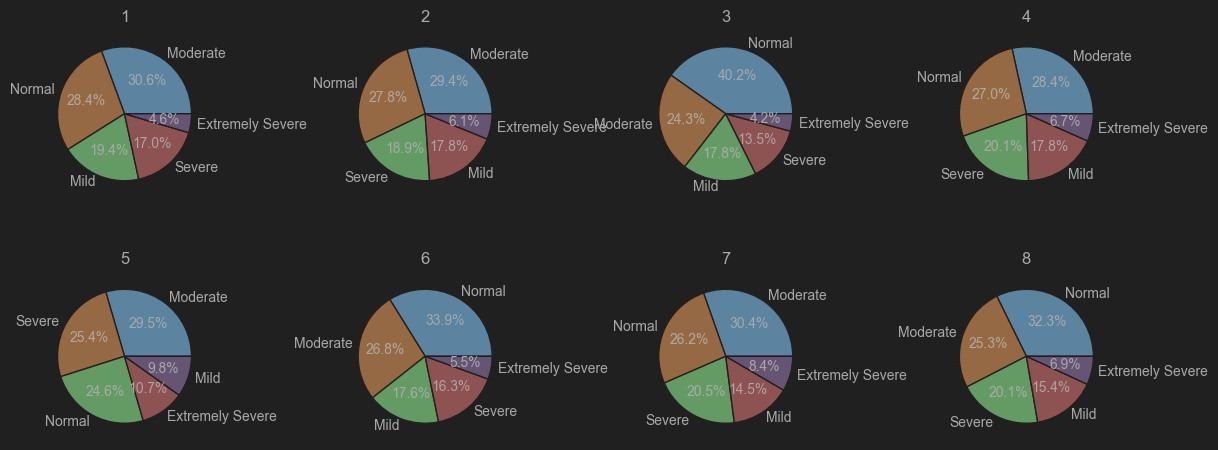


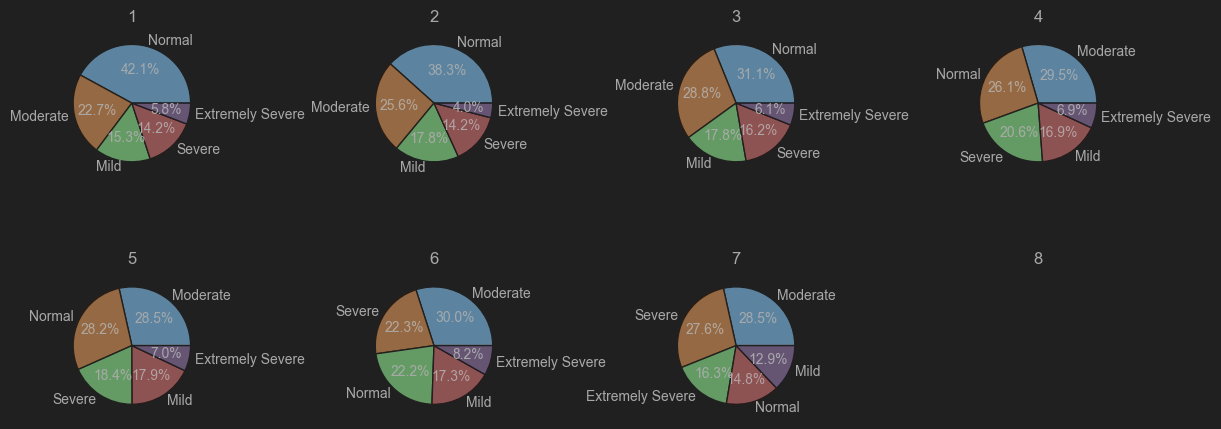


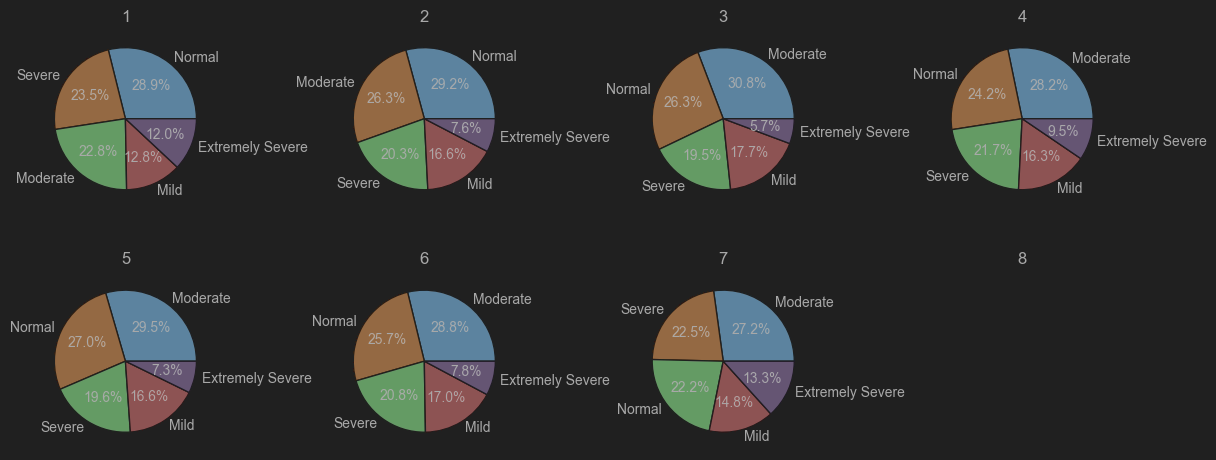


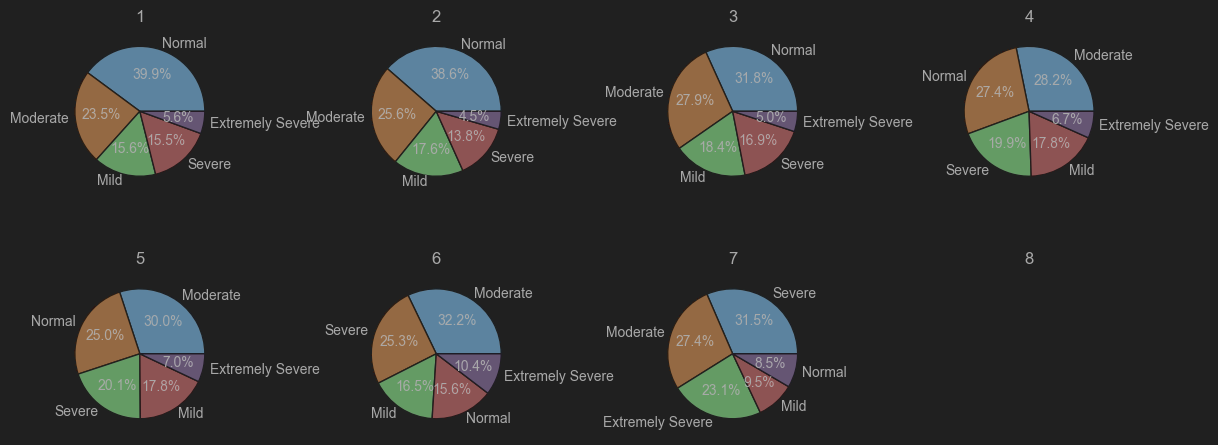


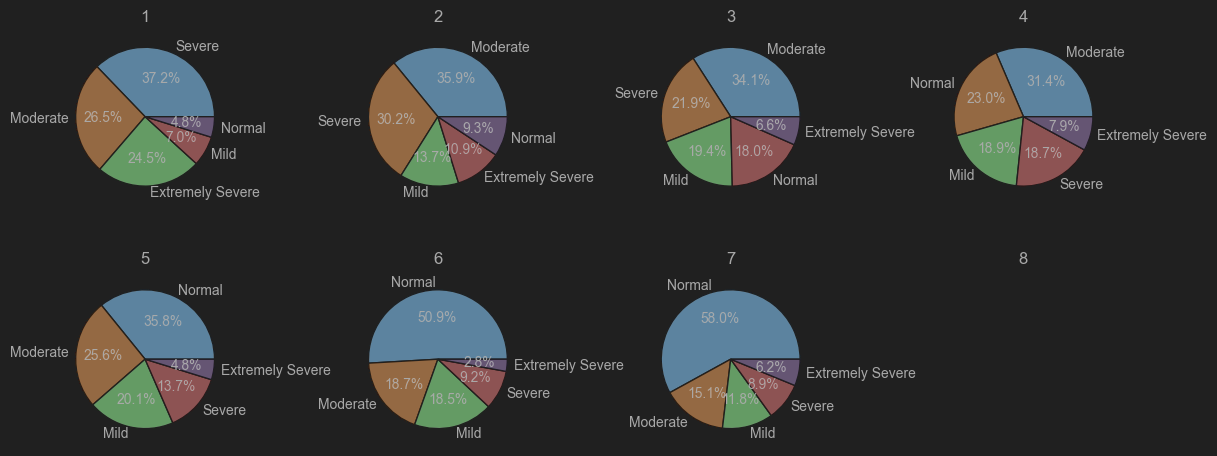


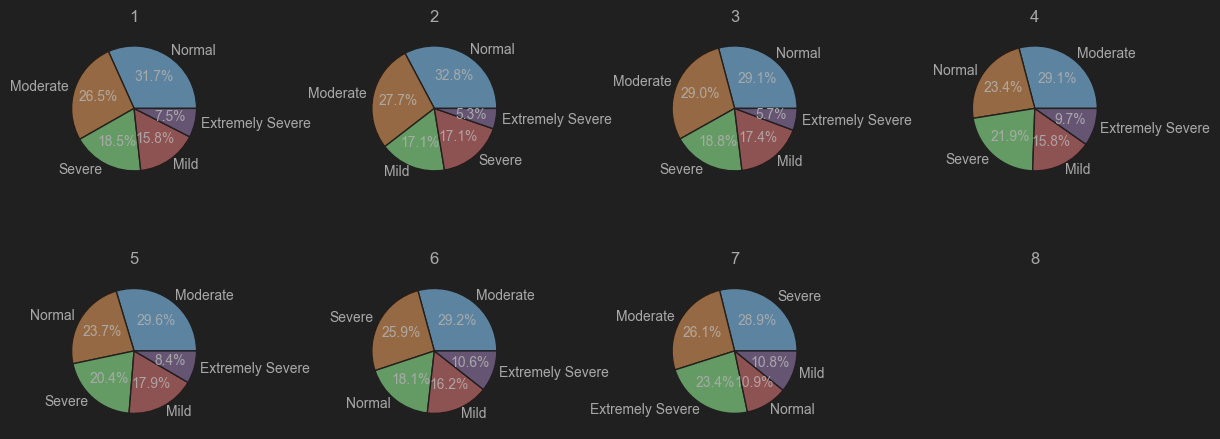


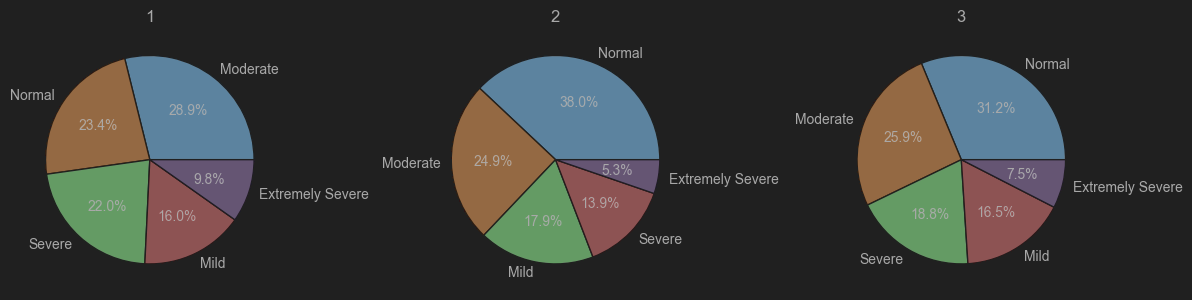












**Algorithms and Methodologies followed:**

In stock prediction AIML projects, various algorithms and methodologies are employed to achieve accurate predictions. Here are some of the key ones:

**Data Preprocessing**: The first step is to preprocess the data, which involves handling missing values, normalizing the data, and feature scaling. This is done using techniques such as Min-Max Scaler, Standard Scaler, and Log Scaler.

**Feature Selection**: Feature selection is crucial in stock prediction as it helps in selecting the most relevant features that affect the stock prices. Techniques such as Correlation Analysis, Mutual Information, and Recursive Feature Elimination are used for feature selection.

**Model Training**: The preprocessed data is then used to train machine learning models such as Linear Regression, Decision Tree Regressor, Random Forest Regressor, and Long Short Term Memory (LSTM) networks.

**Model Evaluation**: The trained models are evaluated using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Percentage Error (RMSPE), and Coefficient of Determination (R2).

**Hyperparameter Tuning**: Hyperparameter tuning is done using techniques such as Grid Search, Random Search, and Bayesian Optimization to optimize the performance of the models.

**Model Selection**: The best-performing model is selected based on the evaluation metrics, and it is used for making predictions on the test data.

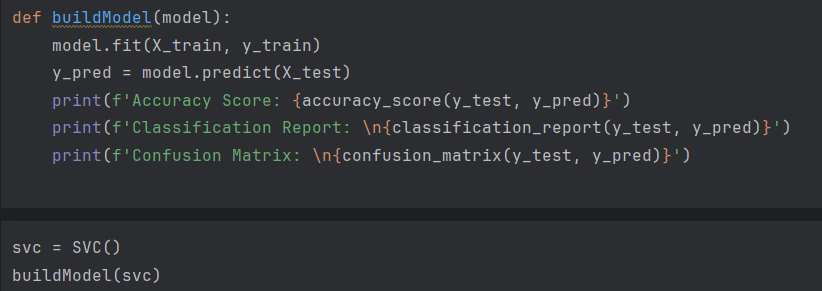
Here is some sample code for building a Decision Tree Regressor model:

Similarly, other machine learning models such as Linear Regression, Random Forest Regressor, and LSTM networks can be built and evaluated using similar code.

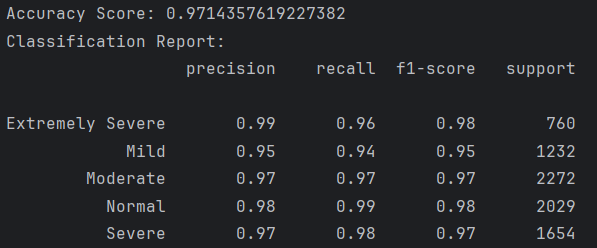
**Stochastic Gradient Descent**: Stochastic Gradient Descent is an optimization algorithm used to find the model parameters that correspond to the best fit between predicted and actual outputs. It is often used in machine learning applications.

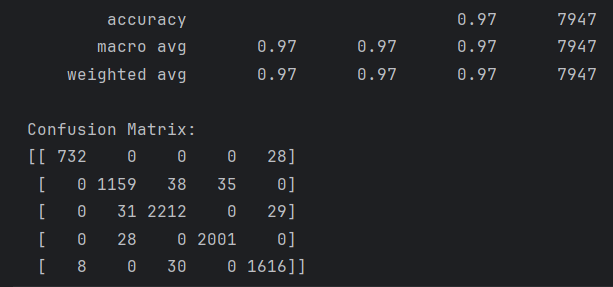
**Ridge Regression**: Ridge Regression is a regularization technique used to reduce overfitting in machine learning models. It adds a penalty term to the loss function to discourage large weights.

**Lasso Regression**: Lasso Regression is another regularization technique used to reduce overfitting in machine learning models. It adds a penalty term to the loss function to discourage large weights.



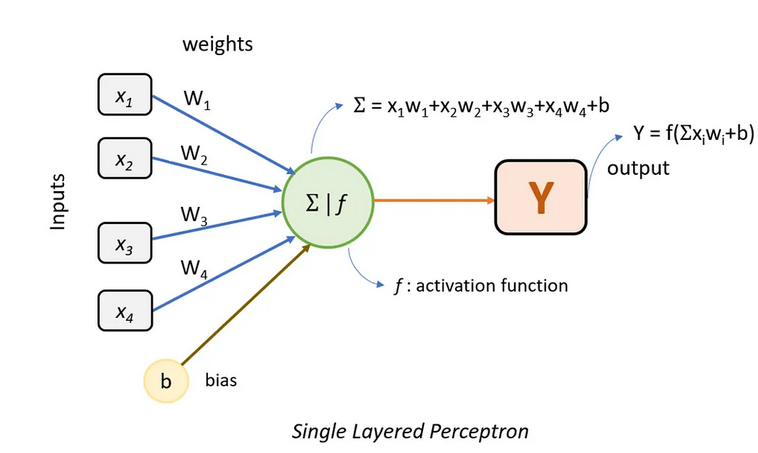
**Results:**





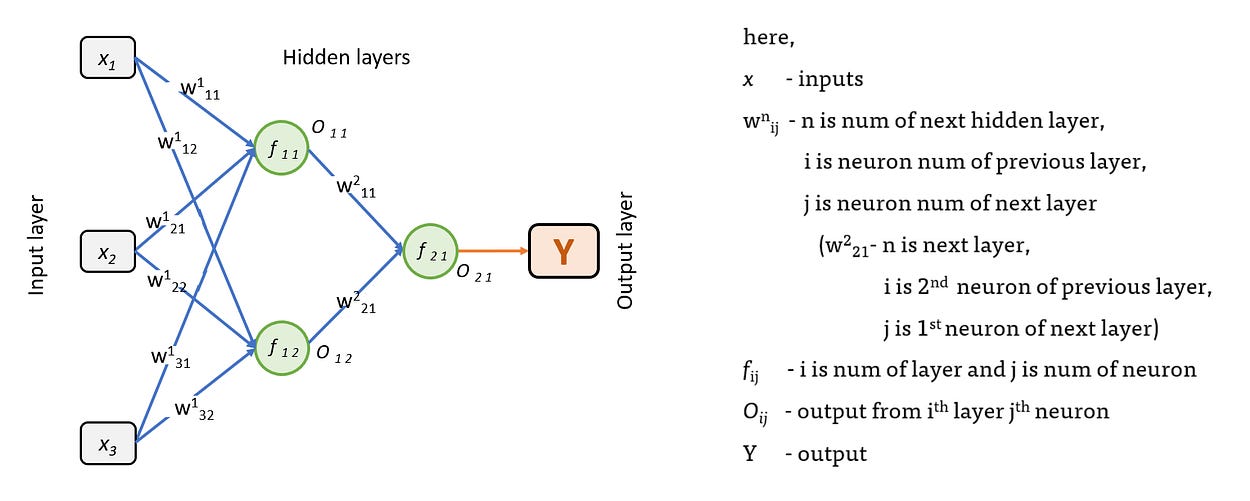
**Multi-Layer Perceptron:**

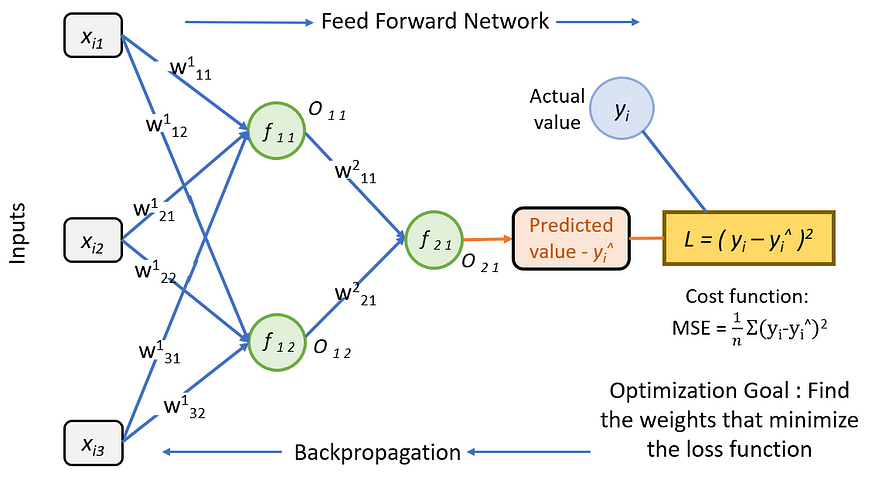
A perceptron is a **single-layer** neural network inspired from biological neurons. The so-called dendrites in biological neuron are responsible for getting incoming signals and cell body is responsible for the processing of input signals and if it fires, the nerve impulse is sent through the axon.

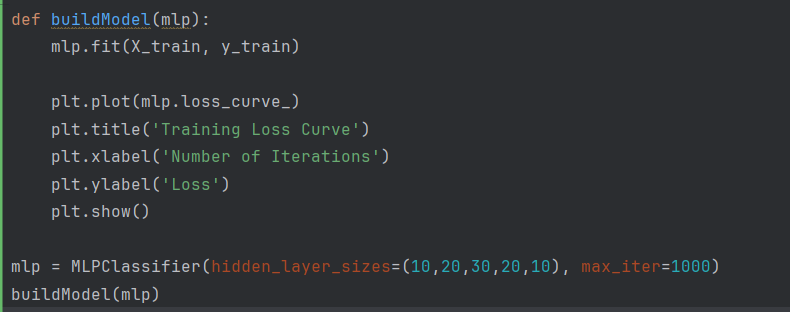
****

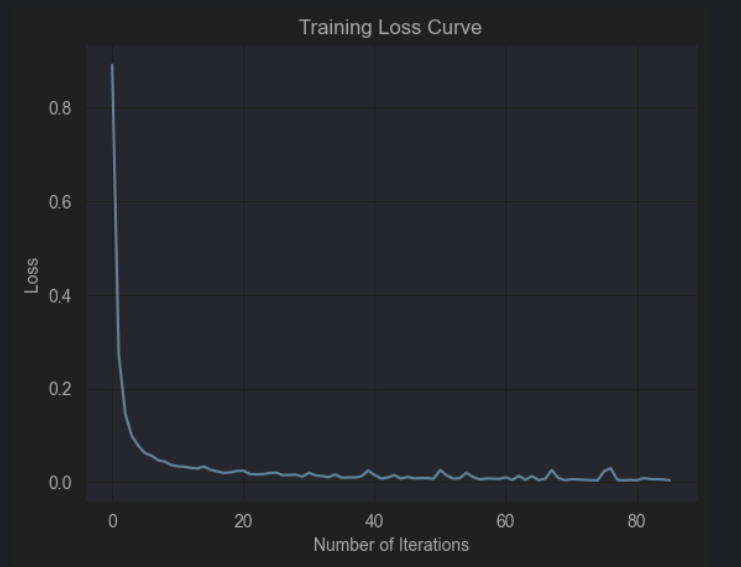
a perceptron gives us only linear relationship between inputs and output. When we need to solve more complex problems, a more complex model is needed which can give us [non-linear relationship](https://en.wikipedia.org/wiki/Nonlinear_system#:~:text=In%20mathematics%20and%20science%2C%20a,the%20change%20of%20the%20input.&text=Systems%20can%20be%20defined%20as,functions%20appear%20in%20the%20equations.). This is solved by using Multi-Layered Perceptron NN.

A Multi-Layered Perceptron NN can have n-number of hidden layers between input and output layer. These hidden layers can have n-number of neurons, in which the first hidden layer takes input from input layer and process them using activation function and pass them to next hidden layers until output layer. Every neuron in a hidden layer uses a non-linear activation function. MLP uses a supervised learning technique called backpropagation for training.

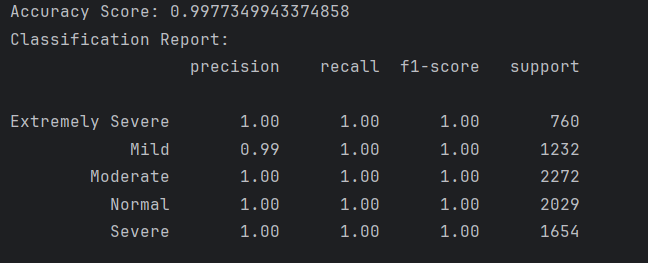


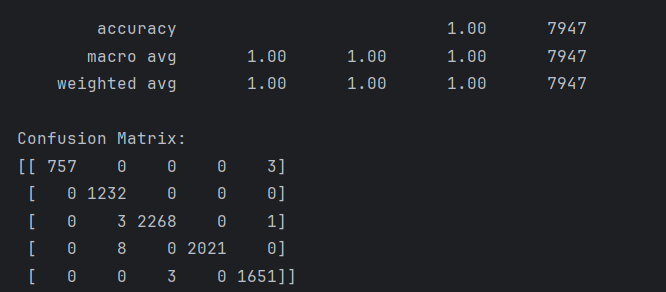






**Results:**

****

****

1. **Evaluation Metrics:**

. Mean Squared Error (MSE**)**: This metric measures the average squared difference between the predicted and actual values. A lower MSE indicates better performance.

Root Mean Squared Error (RMSE): This metric is the square root of the MSE, and it measures the standard deviation of the prediction errors. A lower RMSE indicates better performance.

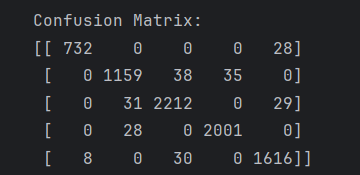
Mean Absolute Error (MAE): This metric measures the average absolute difference between the predicted and actual values. A lower MAE indicates better performance.

Coefficient of Determination (R2**)**: This metric measures the proportion of the variance in the dependent variable that is predictable from the independent variable(s). A higher R2 indicates better performance.

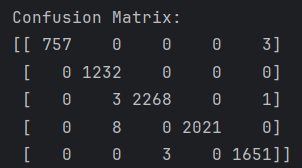
Mean Absolute Percentage Error (MAPE**)**: This metric measures the average absolute difference between the predicted and actual values as a percentage of the actual values. A lower MAPE indicates better performance.

Explained Variance Score (EVS): This metric measures the proportion of the variance in the dependent variable that is explained by the independent variable(s). A higher EVS indicates better performance.

These evaluation metrics are used to compare the performance of different models and to select the best-performing model for making predictions on new data. It is important to note that the choice of evaluation metric depends on the specific problem and the characteristics of the data.

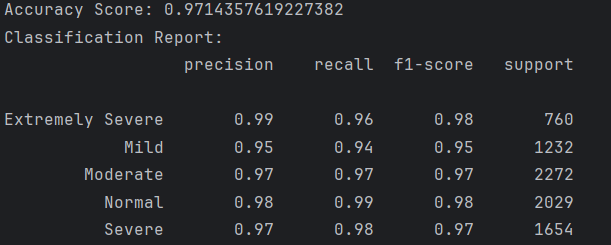
* + - 1. ****

MlP:

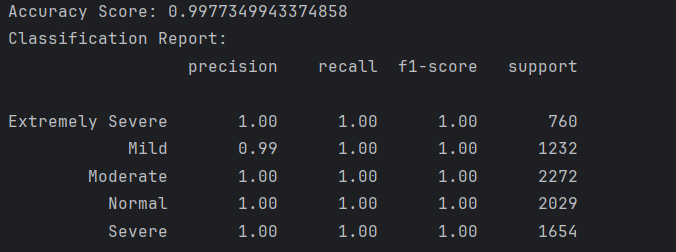
****

**Classification Report:**

**SVC:**

****

**MLP:**

****