Laboratory of Data Science

Tackling COVID-19 Data Imbalance and Health Disparity using Deep Transfer Learning

- Shruti Mandaokar

Research Questions we can think about -

1. **Fairness-aware Temporal Analysis of COVID-19 Mortality Trends:**

- How do COVID-19 mortality trends evolve over time while considering fairness and equity across different racial and ethnic groups, age ranges, and regions?
- Are there temporal patterns in mortality rates that disproportionately affect certain demographic groups, and how can fairness-aware algorithms mitigate bias in trend analysis?
-

2. **Addressing Ethnic and Racial Disparities in COVID-19 Mortality with Data Imbalance:**
- Given the presence of potential data imbalance, how can we develop fairness-aware machine learning models to accurately estimate and mitigate ethnic and racial disparities in COVID-19 mortality rates?
- Can novel algorithmic approaches handle data imbalance while ensuring equitable representation and treatment of underrepresented groups?

3. **Fairness-aware Modeling of Age-specific Risk Factors for COVID-19 Mortality:**
- Investigate age-specific risk factors for COVID-19 mortality while addressing fairness and bias considerations.
- How do underlying health conditions and demographic factors contribute to mortality rates across different age groups, and how can fairness-aware modeling techniques mitigate biases in assessing age-specific risks?

Link : https://catalog.data.gov/dataset/provisional-weekly-deaths-by-region-race-age-997d6

(194040 rows × 14 columns)

- Start Date: Start date of the time period covered by the data. (Dropped)
- End Date: End date of the time period covered by the data. (Dropped)
- Group: Data grouping, including by month, by week, by total, or by year.
    - *Classes*: By Month, By Week, By Total, By Year
- Year: Year of the data, ranging from 2019 to 2023 and including combined periods like 2019/2020 and 2020-2023.
    - *Classes*: 2020, 2021, 2022, 2023, 2019/2020, 2020/2021, 2020-2023, 2021/2022
- Month: Month of the data, ranging from January to December.
    - *Classes*: (Blanks), 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
- MMWR Week: Week according to the Morbidity and Mortality Weekly Report (MMWR) system.
    - *Classes*: (Blanks), 1, 2, 3, ..., 53
- Week-Ending Date: Date when the week ended.
- HHS Region: Geographic region as defined by the United States Department of Health and Human Services.
    - *Classes*: Region 0 : United States, - Region 1: Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont; - Region 2: New Jersey, New York, New York City, Puerto Rico; - Region 3: Delaware, District of Columbia, Maryland, Pennsylvania, Virginia, West Virginia; - Region 4: Alabama, Florida, Georgia, Kentucky, Mississippi, North Carolina, South Carolina, Tennessee; - Region 5: Illinois, Indiana, Michigan, Minnesota, Ohio, Wisconsin; - Region 6: Arkansas, Louisiana, New Mexico, Oklahoma, Texas; - Region 7: Iowa, Kansas, Missouri, Nebraska; - Region 8: Colorado, Montana, North Dakota, South Dakota, Utah, Wyoming; - Region 9: Arizona, California, Hawaii, Nevada; - Region 10: Alaska, Idaho, Oregon, Washington.
- Race and Hispanic Origin Group: Ethnicity and race categories.
    - *Classes*: Hispanic, non-Hispanic American Indian or Alaska Native, non-Hispanic Asian, non-Hispanic Black, non-Hispanic more than one race, non-Hispanic Native Hawaiian or other Pacific Islander, non-Hispanic White, and unknown.
- Age Group: Age categories ranging from 0-4 years to 85 years and over.
    - *Classes*: (0-4 years), (5-17 years), (18-29 years), (30-39 years), (40-49 years), (50-64 years), (65-74 years), (75-84 years), (85 years and over)
- COVID-19 Deaths: Number of deaths involving COVID-19 reported for the specified demographic group and time period.
- Total Deaths: Total number of deaths reported for the specified demographic group and time period.
- Footnote: Indicates if data cells have counts suppressed due to NCHS confidentiality standards.

Categorical Columns:

- Data As Of
- Start Date
- End Date
- Group
- Year
- Month
- MMWR Week
- Week-Ending Date
- HHS Region
- Race and Hispanic Origin Group
- Age Group
- Footnote

Numerical Columns:

- Month
- MMWR Week
- COVID-19 Deaths
- Total Deaths

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194040 entries, 0 to 194039
Data columns (total 14 columns):
 #   Column                          Non-Null Count   Dtype
---  ------                          --------------   -----
 0   Data As Of                      194040 non-null  object
 1   Start Date                      194040 non-null  object
 2   End Date                        194040 non-null  object
 3   Group                           194040 non-null  object
 4   Year                            194040 non-null  object
 5   Month                           35640 non-null   float64
 6   MMWR Week                       154440 non-null  float64
 7   Week-Ending Date                154440 non-null  object
 8   HHS Region                      194040 non-null  object
 9   Race and Hispanic Origin Group  194040 non-null  object
 10  Age Group                       194040 non-null  object
 11  COVID-19 Deaths                 152813 non-null  float64
 12  Total Deaths                    129598 non-null  float64
 13  Footnote                        95590 non-null   object
dtypes: float64(4), object(10)
memory usage: 20.7+ MB
```

```
[136] df.dtypes

Data As Of                        object
Start Date                        object
End Date                          object
Group                             object
Year                              object
Month                            float64
MMWR Week                        float64
Week-Ending Date                  object
HHS Region                        object
Race and Hispanic Origin Group    object
Age Group                         object
COVID-19 Deaths                  float64
Total Deaths                     float64
Footnote                          object
dtype: object
```

Handling Unnecessary Columns and Missing Data

- **Removed Footnote, Month**
- **Make HHS column uniform**
- **Filled Rows with NaN to 0**
- **Used Forward and Backward Filling for Time Series/ Date Time Data**

```
print("Missing values:")
df.isnull().sum()
```

```
Missing values:
Data As Of                          0
Start Date                          0
End Date                            0
Group                               0
Year                                0
MMWR Week                       39600
Week-Ending Date                39600
HHS Region                          0
Race and Hispanic Origin Group      0
Age Group                           0
COVID-19 Deaths                 41227
Total Deaths                    64442
dtype: int64
```

```
#print("Missing values:")
df.isnull().sum()
```

```
Data As Of                        0
Start Date                        0
End Date                          0
Group                             0
Year                              0
MMWR Week                         0
Week-Ending Date                  0
HHS Region                        0
Race and Hispanic Origin Group    0
Age Group                         0
COVID-19 Deaths                   0
Total Deaths                      0
dtype: int64
```

Handling Unnecessary Columns and Missing Data

- Removed Footnote, Month
- Make HHS column uniform
- Filled Rows with NaN to 0
- Used Forward and Backward Filling for Time Series/ Date Time Data

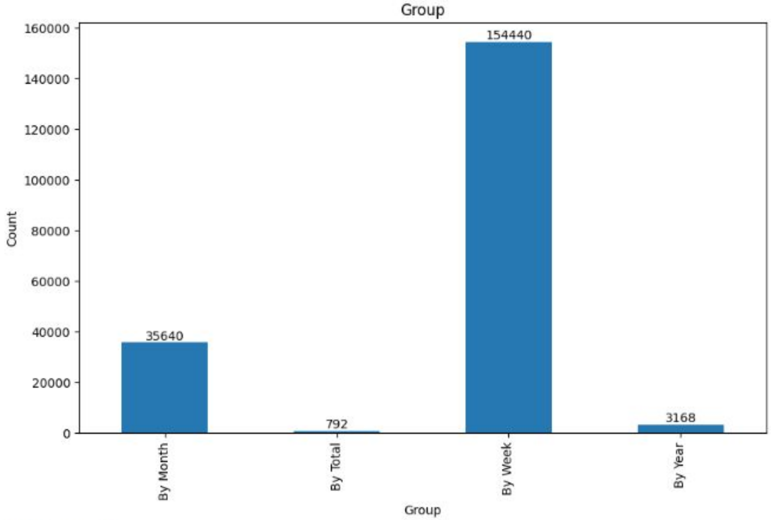
```
print("Missing values:")
df.isnull().sum()
```

```
Missing values:
Data As Of                        0
Start Date                        0
End Date                          0
Group                             0
Year                              0
MMWR Week                     39600
Week-Ending Date              39600
HHS Region                        0
Race and Hispanic Origin Group    0
Age Group                         0
COVID-19 Deaths               41227
Total Deaths                  64442
dtype: int64
```

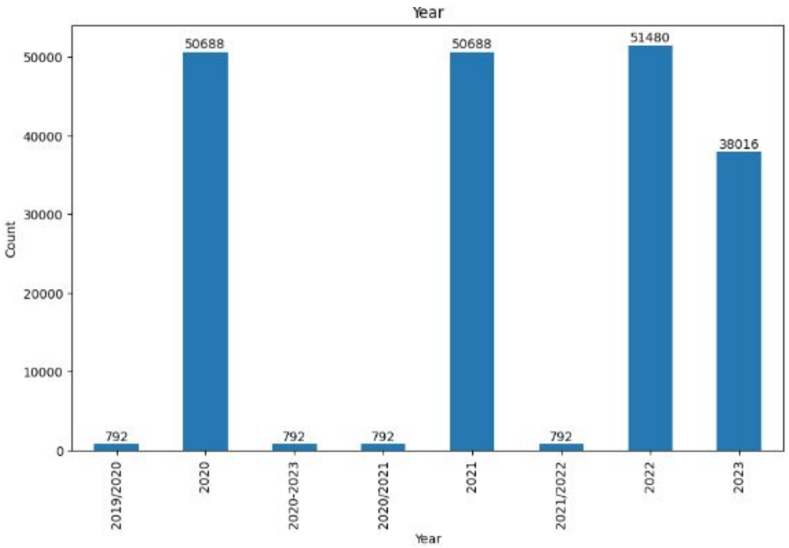
```
#print("Missing values:")
df.isnull().sum()
```

```
Data As Of                        0
Start Date                        0
End Date                          0
Group                             0
Year                              0
MMWR Week                         0
Week-Ending Date                  0
HHS Region                        0
Race and Hispanic Origin Group    0
Age Group                         0
COVID-19 Deaths                   0
Total Deaths                      0
dtype: int64
```

EDA and Plots

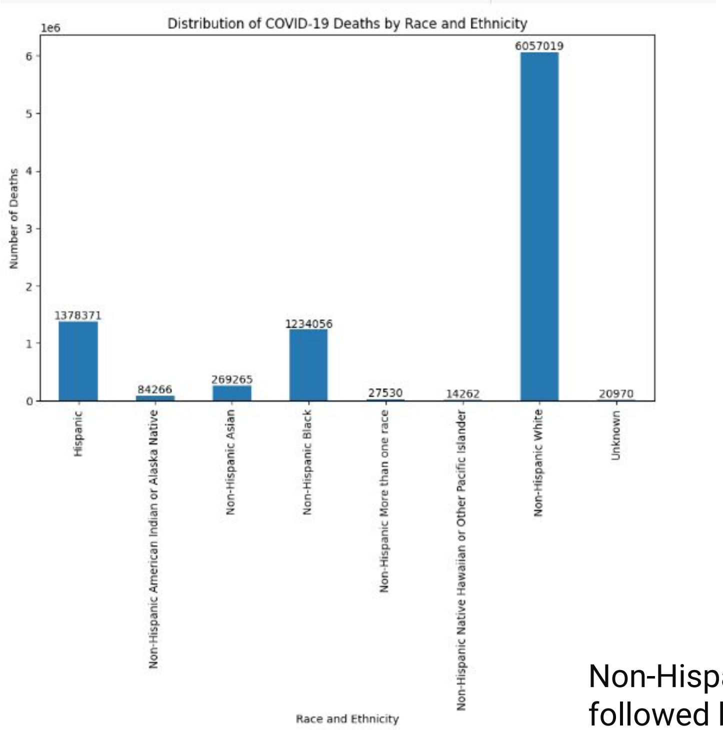



```
Counts for Group:
Group
By Month     35640
By Total       792
By Week     154440
By Year       3168
Name: count, dtype: int64
```




```
Counts for Year:
Year
2019/2020      792
2020         50688
2020-2023      792
2020/2021      792
2021         50688
2021/2022      792
2022         51480
2023         38016
Name: count, dtype: int64
```

What is the distribution of COVID-19 deaths across different demographic groups?

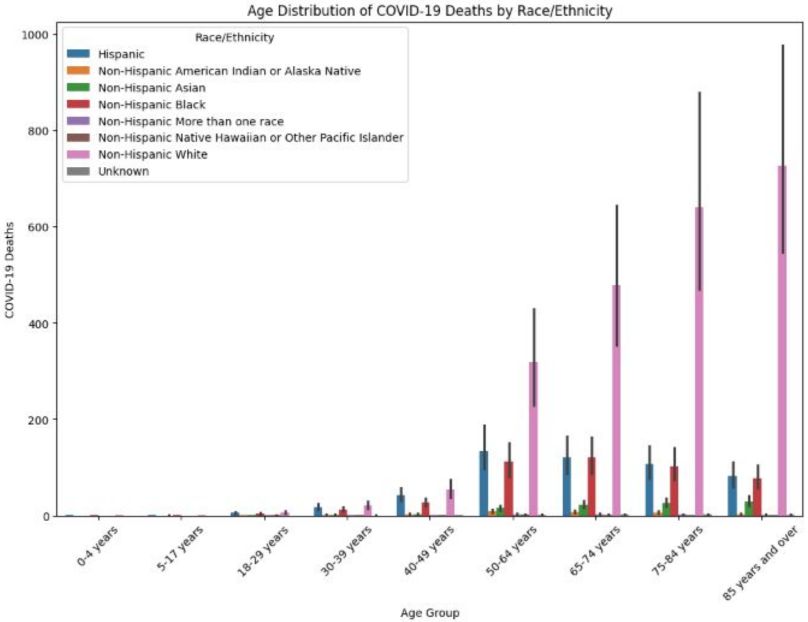


Race and Ethnicity

```
Counts of COVID-19 Deaths by Race and Ethnicity:
Race and Hispanic Origin Group
Hispanic                                                    1378371
Non-Hispanic American Indian or Alaska Native                 84266
Non-Hispanic Asian                                           269265
Non-Hispanic Black                                          1234056
Non-Hispanic More than one race                               27530
Non-Hispanic Native Hawaiian or Other Pacific Islander        14262
Non-Hispanic White                                          6057019
Unknown                                                       20970
Name: COVID-19 Deaths, dtype: int64
```

Non-Hispanic White individuals have the highest count of COVID-19 deaths, followed by Non-Hispanic Black individuals.

Non-Hispanic White individuals generally have the highest counts of COVID-19 deaths across all age groups, followed by Non-Hispanic Black individuals. Hispanic individuals also have significant counts of COVID-19 deaths, particularly in older age groups. Non-Hispanic Asian individuals tend to have lower counts of COVID-19 deaths compared to other racial and ethnic groups, but the disparity may vary across age groups




```
Counts of COVID-19 Deaths for Non-Hispanic Black:
Age Group
0-4 years               1189
18-29 years            11356
30-39 years            33680
40-49 years            73599
5-17 years              1267
50-64 years           303300
65-74 years           326321
75-84 years           276448
85 years and over     206896
Name: COVID-19 Deaths, dtype: int64
```

```
Counts of COVID-19 Deaths for Non-Hispanic Asian:
Age Group
0-4 years                108
18-29 years             1272
30-39 years             3888
40-49 years             8735
5-17 years               150
50-64 years            42243
65-74 years            61551
75-84 years            71022
85 years and over      80296
Name: COVID-19 Deaths, dtype: int64
```

```
Counts of COVID-19 Deaths for Hispanic:
Age Group
0-4 years               1088
18-29 years            15496
30-39 years            46928
40-49 years           113697
5-17 years              1385
50-64 years           363369
65-74 years           325992
75-84 years           287810
85 years and over     222606
Name: COVID-19 Deaths, dtype: int64
```

```
Counts of COVID-19 Deaths for Non-Hispanic More than one race:
Age Group
0-4 years                124
18-29 years              425
30-39 years              964
40-49 years             2042
5-17 years                96
50-64 years             6995
65-74 years             6642
75-84 years             6004
85 years and over       4238
Name: COVID-19 Deaths, dtype: int64
```

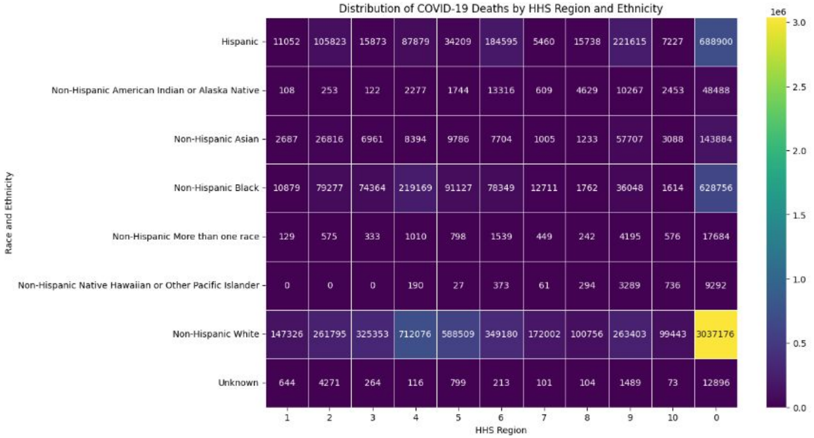
```
Counts of COVID-19 Deaths for Non-Hispanic Native Hawaiian or Other Pacific Islander:
Age Group
0-4 years              32
18-29 years           250
30-39 years           893
40-49 years          1685
5-17 years             32
50-64 years          4977
65-74 years          3514
75-84 years          2013
85 years and over     866
Name: COVID-19 Deaths, dtype: int64
```

```
Counts of COVID-19 Deaths for Non-Hispanic White:
Age Group
0-4 years                1746
18-29 years             18019
30-39 years             56097
40-49 years            147830
5-17 years               1959
50-64 years            859509
65-74 years           1288114
75-84 years           1725343
85 years and over     1958402
Name: COVID-19 Deaths, dtype: int64
```

```
Counts of COVID-19 Deaths for Unknown:
Age Group
0-4 years              24
18-29 years            72
30-39 years           188
40-49 years           510
5-17 years              8
50-64 years          4585
```

```
Counts of COVID-19 Deaths for Non-Hispanic American Indian or Alaska Native:
Age Group
0-4 years                 48
18-29 years             1203
30-39 years             3794
40-49 years             7211
5-17 years                68
50-64 years            24454
65-74 years            22399
75-84 years            16256
85 years and over       8833
Name: COVID-19 Deaths, dtype: int64
```

COVID-19 Deaths by HHS Region and Ethnicity



(Classes :

- Region 0 : United States,
- Region 1: Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont;
- Region 2: New Jersey, New York, New York City, Puerto Rico;
- Region 3: Delaware, District of Columbia, Maryland, Pennsylvania, Virginia, West Virginia;
- Region 4: Alabama, Florida, Georgia, Kentucky, Mississippi, North Carolina, South Carolina, Tennessee;
- Region 5: Illinois, Indiana, Michigan, Minnesota, Ohio, Wisconsin;
- Region 6: Arkansas, Louisiana, New Mexico, Oklahoma, Texas;
- Region 7: Iowa, Kansas, Missouri, Nebraska;
- Region 8: Colorado, Montana, North Dakota, South Dakota, Utah, Wyoming;
- Region 9: Arizona, California, Hawaii, Nevada;
- Region 10: Alaska, Idaho, Oregon, Washington.)

Regional Disparity

- HHS Region 4 has the highest counts of COVID-19 deaths across most ethnic groups, followed by HHS Region 5 and HHS Region 6.
- HHS Region 10 generally has lower counts of COVID-19 deaths compared to other regions.

Ethnic Disparities within Regions

- Non-Hispanic Black individuals have relatively high counts of COVID-19 deaths in most regions, particularly in HHS Region 4.
- Hispanic individuals also have significant counts of COVID-19 deaths, with notable numbers in HHS Region 5 and HHS Region 6.
- Non-Hispanic White individuals tend to have higher counts of COVID-19 deaths in regions like HHS Region 4 and HHS Region 5.
- Counts for Non-Hispanic Asian and Non-Hispanic American Indian or Alaska Native individuals vary across regions but generally tend to be lower compared to other ethnic groups.

Relative Magnitudes:

- Looking at the total counts, it's evident that Non-Hispanic White individuals have the highest overall counts of COVID-19 deaths across all HHS regions, followed by Hispanic individuals and Non-Hispanic Black individuals.

Encoded Categorical Variables

```
# Encode categorical variables
label_encoder = LabelEncoder()
data['Race and Hispanic Origin Group'] = label_encoder.fit_transform(data['Race and Hispanic Origin Group'])
data['Age Group'] = label_encoder.fit_transform(data['Age Group'])
data['Group'] = label_encoder.fit_transform(data['Group'])
data
```

	Start Date	End Date	Group	Year	MMWR Week	Week-Ending Date	HHS Region	Race and Hispanic Origin Group	Age Group	COVID-19 Deaths	Total Deaths
0	2019-12-29	2020-01-04	2	2019/2020	1	2020-01-04	0	0	0	0	104
1	2019-12-29	2020-01-04	2	2019/2020	1	2020-01-04	0	0	4	0	41
2	2019-12-29	2020-01-04	2	2019/2020	1	2020-01-04	0	0	1	0	190
3	2019-12-29	2020-01-04	2	2019/2020	1	2020-01-04	0	0	2	0	237
4	2019-12-29	2020-01-04	2	2019/2020	1	2020-01-04	0	0	3	0	325
...	...	...	...	...	...	...	...	...	...	...	...
194035	2020-01-01	2023-09-23	1	2020-2023	38	2023-09-23	10	7	3	0	89
194036	2020-01-01	2023-09-23	1	2020-2023	38	2023-09-23	10	7	5	19	457
194037	2020-01-01	2023-09-23	1	2020-2023	38	2023-09-23	10	7	6	15	446
194038	2020-01-01	2023-09-23	1	2020-2023	38	2023-09-23	10	7	7	17	243
194039	2020-01-01	2023-09-23	1	2020-2023	38	2023-09-23	10	7	8	10	113

194040 rows × 11 columns

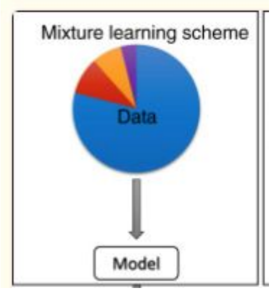
Gradient Boosting Algorithm

Gradient boosting is an ensemble learning method, meaning it combines multiple weak learners (often decision trees) to create a strong predictive model. By sequentially adding predictors, it corrects errors made by previous models, leading to improved accuracy.

Mixture Learning Scheme (RMSE : 406.071)

```
# Mixture Learning Scheme
X_mix = data.drop(columns=['COVID-19 Deaths','Start Date', 'End Date','Year', 'Week-Ending Date', ])
y_mix = data['COVID-19 Deaths']
X_mix_train, X_mix_test, y_mix_train, y_mix_test = train_test_split(X_mix, y_mix, test_size=0.2, random_state=42)

model_mix = GradientBoostingRegressor()  # Choose any model
model_mix.fit(X_mix_train, y_mix_train)
y_mix_pred = model_mix.predict(X_mix_test)
mix_rmse = mean_squared_error(y_mix_test, y_mix_pred, squared=False)
print("Mixture Learning RMSE:", mix_rmse)
```

```
Mixture Learning RMSE: 406.0716060623727
```




```
Unique values in column 'Race and Hispanic Origin Group':
['Hispanic' 'Non-Hispanic American Indian or Alaska Native'
 'Non-Hispanic Asian' 'Non-Hispanic Black'
 'Non-Hispanic More than one race'
 'Non-Hispanic Native Hawaiian or Other Pacific Islander'
 'Non-Hispanic White' 'Unknown']
```

Gradient Boosting Algorithm

Gradient boosting is an ensemble learning method, meaning it combines multiple weak learners (often decision trees) to create a strong predictive model. By sequentially adding predictors, it corrects errors made by previous models, leading to improved accuracy.

- used for predictive modeling, particularly in scenarios where the relationship between predictors and the target variable is complex.
- It's a type of ensemble learning method that combines the predictions of multiple weak learners, typically decision trees, to create a strong predictive model.
- Loss functions used in regression include mean squared error (MSE) and mean absolute error (MAE).

Objective Function: Minimize loss between predicted and actual values (e.g., MSE or MAE).

Weak Learners: Use decision trees to split data and minimize loss within subsets.

Building Ensemble: Add decision trees sequentially to improve predictions.

Gradient Descent: Calculate gradient of loss w.r.t. predictions to guide model updates.

Fitting Weak Learner: Fit new decision tree to negative gradient to correct errors.

Shrinkage (Learning Rate): Control contribution of each tree to prevent overfitting.

Combining Predictions: Weight predictions of all trees for final prediction.

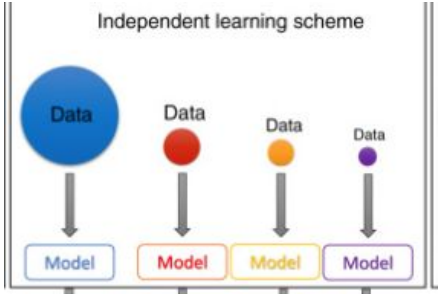
Regularization: Incorporate techniques to prevent overfitting (e.g., limiting tree depth).

Gradient Boosting Algorithm

```
Unique values in column 'Race and Hispanic Origin Group':
['Hispanic' 'Non-Hispanic American Indian or Alaska Native'
 'Non-Hispanic Asian' 'Non-Hispanic Black'
 'Non-Hispanic More than one race'
 'Non-Hispanic Native Hawaiian or Other Pacific Islander'
 'Non-Hispanic White' 'Unknown']
```

Gradient boosting is an ensemble learning method, meaning it combines multiple weak learners (often decision trees) to create a strong predictive model. By sequentially adding predictors, it corrects errors made by previous models, leading to improved accuracy.

Independent Learning Scheme




```python
# Independent Learning Scheme
ethnic_groups = data['Race and Hispanic Origin Group'].unique()
for group in ethnic_groups:
    group_data = data[data['Race and Hispanic Origin Group'] == group]
    X_group = group_data.drop(columns=['COVID-19 Deaths','Start Date', 'End Date','Year', 'Week-Ending Date'])
    y_group = group_data['COVID-19 Deaths']
    X_group_train, X_group_test, y_group_train, y_group_test = train_test_split(X_group, y_group, test_size=0.2, random_state=42)
    
    model_group = GradientBoostingRegressor()  # Choose any model
    model_group.fit(X_group_train, y_group_train)
    y_group_pred = model_group.predict(X_group_test)
    group_rmse = mean_squared_error(y_group_test, y_group_pred, squared=False)
    print(f"Independent Learning RMSE for {group}:", group_rmse)
```

```
Independent Learning RMSE for 0: 172.74848195897692
Independent Learning RMSE for 1: 15.16618098442474
Independent Learning RMSE for 2: 52.1330147005161
Independent Learning RMSE for 3: 111.7448059260064
Independent Learning RMSE for 4: 3.406019320722873
Independent Learning RMSE for 5: 4.519149986864198
Independent Learning RMSE for 6: 910.4517464802346
Independent Learning RMSE for 7: 5.087558899956792
```

Ethnic groups 1, 4, and 5 have relatively low RMSE values (15.17, 3.41, and 4.52, respectively), suggesting that the model performs well in predicting COVID-19 deaths for these groups.

Ethnic groups 2 and 3 have moderate RMSE values (52.13 and 111.74, respectively), indicating that the model's predictions for these groups have a higher level of error compared to groups 1, 4, and 5.

Ethnic groups 0, 6, and 7 have high RMSE values (172.75, 910.45, and 5.09, respectively), indicating that the model's predictions for these groups are less accurate compared to the other groups.

Transfer Learning Scheme

Finding Minority and Majority Groups: It identifies the minority and majority ethnic groups in the dataset based on the counts of each group's occurrences.

Training Model on Majority Group's Data: It selects the data corresponding to the majority ethnic group and splits it into features (X_majority) and target (y_majority). Then, it splits the data into training and testing sets and trains a GradientBoostingRegressor model on the majority group's training data.

Transfer Knowledge to Minority Groups: It iterates through each unique ethnic group in the dataset, excluding the majority group. For each minority group, it selects the corresponding data and splits it into features (X_minority) and target (y_minority). It then applies knowledge transfer from the model trained on the majority group's data by predicting COVID-19 deaths for the minority group's testing data.

Finally, it calculates the root mean squared error (RMSE) between the predicted values and the actual values for each minority group and prints the results.

RMSE is a measure of the model's accuracy, where lower values indicate better performance.

1. **Squared Error Calculation:** For each data point, the squared difference between the predicted value (ŷ) and the actual value (y) is calculated: $(y - \hat{y})^2$.
2. **Mean Squared Error (MSE):** The squared errors are averaged across all data points to compute the Mean Squared Error: $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where n is the number of data points.
3. **Root Mean Squared Error (RMSE):** RMSE is the square root of the MSE, providing a measure of the average magnitude of the errors in the same units as the target variable: $RMSE = \sqrt{MSE}$

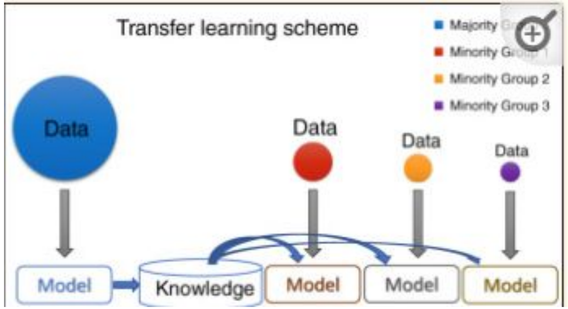
Transfer Learning Scheme

```python
majority_group = 'Non-Hispanic White'
majority_data = data[data['Race and Hispanic Origin Group'] == majority_group]
X_majority = majority_data.drop(columns=['COVID-19 Deaths'])
y_majority = majority_data['COVID-19 Deaths']
X_majority_train, X_majority_test, y_majority_train, y_majority_test = train_test_split(X_majority, y_majority, test_size=0.2, random_state=42)

model_transfer = GradientBoostingRegressor()  # Choose any model
model_transfer.fit(X_majority_train, y_majority_train)

minority_groups = ethnic_groups[ethnic_groups != majority_group]
for group in minority_groups:
    group_data = data[data['Race and Hispanic Origin Group'] == group]
    X_group = group_data.drop(columns=['COVID-19 Deaths'])
    y_group = group_data['COVID-19 Deaths']

    # Use the trained model to predict for minority groups
    y_group_pred = model_transfer.predict(X_group)
    group_rmse = mean_squared_error(y_group, y_group_pred, squared=False)
    print(f"Transfer Learning RMSE for {group}:", group_rmse)
```

```python
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import numpy as np

# Find minority and majority groups
ethnic_group_counts = data['Race and Hispanic Origin Group'].value_counts()
minority_group = ethnic_group_counts.idxmin()
majority_group = ethnic_group_counts.idxmax()

# Train model on majority group's data
majority_data = data[data['Race and Hispanic Origin Group'] == majority_group]
X_majority = majority_data.drop(columns=['COVID-19 Deaths', 'Start Date', 'End Date', 'Year', 'Week-Ending Date'])
y_majority = majority_data['COVID-19 Deaths']
X_majority_train, X_majority_test, y_majority_train, y_majority_test = train_test_split(X_majority, y_majority, test_size=0.2, random_state=42)

model_majority = GradientBoostingRegressor()
model_majority.fit(X_majority_train, y_majority_train)

# Transfer knowledge to minority groups
for group in data['Race and Hispanic Origin Group'].unique():
    if group != majority_group:
        minority_data = data[data['Race and Hispanic Origin Group'] == group]
        X_minority = minority_data.drop(columns=['COVID-19 Deaths', 'Start Date', 'End Date', 'Year', 'Week-Ending Date'])
        y_minority = minority_data['COVID-19 Deaths']
        X_minority_train, X_minority_test, y_minority_train, y_minority_test = train_test_split(X_minority, y_minority, test_size=0.2, random_state=42)

        # Use knowledge transfer from majority group
        y_majority_pred = model_majority.predict(X_minority_test)
        minority_rmse = mean_squared_error(y_minority_test, y_majority_pred, squared=False)
        print(f"Transfer Learning RMSE for {group}: {minority_rmse}")
```

```
Transfer Learning RMSE for 1: 29.29607977721638
Transfer Learning RMSE for 2: 50.48957689804763
Transfer Learning RMSE for 3: 336.803263594373
Transfer Learning RMSE for 4: 15.311465997715922
Transfer Learning RMSE for 5: 6.658942354905979
Transfer Learning RMSE for 6: 1487.461614537069
Transfer Learning RMSE for 7: 27.98353624700487
```

```
Unique values in column 'Race and Hispanic Origin Group':
['Hispanic' 'Non-Hispanic American Indian or Alaska Native'
 'Non-Hispanic Asian' 'Non-Hispanic Black'
 'Non-Hispanic More than one race'
 'Non-Hispanic Native Hawaiian or Other Pacific Islander'
 'Non-Hispanic White' 'Unknown']
```



Scenario 1: Reasonably good performance.
Scenario 2: Moderate performance, slightly higher error.
Scenario 3: Poor performance, significantly higher error.
Scenario 4: Good performance, low error.
Scenario 5: Excellent performance, very low error.
Scenario 6: Extremely poor performance, very high error.
Scenario 7: Reasonably good performance with relatively low error.