Predicting Spending Patterns for Fixed Income and Family Size Using Machine Learning.

MIS-690 Capstone Project Thesis

Submitted to Grand Canyon University

Graduate Faculty of the Colangelo College of Business

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Business Analytics

By

Megha Gubbala, Sushil Sivaram, Sylvia Nanyangwe.

Phoenix, Arizona

07/21/2021

Approved by:

Isac Artzi 07/21/2021

Professor Date

Abstract

Correlation-and-regression analysis conducted by Mudrak, R. et al. (2020), found an inverse relationship between the shares of expenditure for food with GDP per capita by purchasing power parity, at constant prices. The current pandemic conditions have led to a 60% increase in food insecurity as per research published by Dubowitz, T., et al. 2021. The analysis predicts the optimal expenditure for single mothers with multiple dependents based on factors like income and number of family members. The predictions are based on the public available data Food Affordability published by California Department of Public Health.

Keywords: Predictive analysis, scikit learn, Machine Learning, Prescriptive analysis, Support vector Machine (SVM), Multilayer perceptron (MLPC).

Table of Contents

List of Tables

List of Figures

Business Problem Identification

The impact of Coronavirus has diminished the earning capacity of multiple sections of the population According to new research by Shahnasarian, M. 2021. With minimal opportunity to increase their earning capacities, households are dependent on either government benefits or substantially reducing their spending. H.R. 1319, American Rescue Plan Act of 2021 published by the Congressional Budget Office (2021) currently provides $1,400 Economic Impact Payments to households impacted by this emergency. Small changes in the planning of resources can lead to a substantial increase in allocation of spending as documented by Brandenburg, L. (n.d). One of the primary areas where the spending is reduced is food and nutrition related. This leads to a great food insecurity crisis that could potentially have long-lasting impacts. The scope of this project is to utilize data published by Olaimat, A. (n.d) to analyze and establish the optimal spending patterns of households with different family sizes and earning potential. The predicted value can then be used as a guideline to augment benefit programs to ensure that the crisis is averted.

Background

California Department of Food Affordability was established in 1989, is an organization focused on providing equal food availability and opportunities to all the sections of the society under its wings. The current project will be focused on determining the impact of Covid-19 on the marginalized sector of low-income single women with multiple dependents. Azuma, A. M., et al. (2010) conducted research based on three counties in Los Angeles and reported substantial food insecurity levels within them.

This problem has been encountered previously, however, due to the lack of data; further analysis could not be completed. The published data currently available is adequate to complete a cursory evaluation of the problem statement.

The problem regarding the earning/spending gap in lower-income segments has existed for a long time. This has been compounded by the effects of the novel coronavirus.

The non-availability of adequate nutrition has far-reaching effects on the development of society. The various factors that have compounded the effect of the problem are the current socio-economic conditions, higher unemployment, and under-employment caused due to the strains of virus present in the world. This has led to an increase in the number of households affected by the problem since March 2020.

Internal stakeholders would involve the Board of Directors, Chief Operating Officer, Resident Manager, and the employees who are in direct contact with the families. The external stakeholders include various government organizations and charities are involved in the collecting and disbursing welfare.

The business problem can be addressed by the analytical team utilizing the current data and tools. The data available is mainly in a format that can be consumed by the analytical tool to produce predictive and prescriptive recommendations. Tools currently available support natively inputting the data and further iteratively building multiple regression models that assist with further analysis.

The scope of the project will require a single input file which is available in public domain at Food Affordability - Datasets - California Health and Human Services Open Data Portal published by the California Department of Public Health.

The objective of the research is to identify a cost-effective solution during the pandemic to allocate available funds to the appropriate families with minimal overheads. This in turn will assist impacted households to budget their expenditure effectively during the crisis.

Business Problem Statement

Predict the optimal spending limits of a subset of the population with a defined income to feed and sustain a predefined family size based on the average affordability ratio published by Food Affordability Index (**California Health and Human Services Open Data Portal**).

Analytics Assumptions

For the analytical process to commence the following subcategories have to be defined and set up as listed below.

Resources – The project will primarily involve the participation of the following personnel. Scheduling and availability of resources from the start date till the end data should be outlined and set aside.

Business point of contact: 2 (Primary and a secondary)

Data Analyst: 1

Project Manager/Scrum Master: 1

Data Developer: 1

Data Validator: 1

Delivery – The delivery timelines need to be published along with the nature of the visualizations, Key performance indicators (KPI's), and other pertinent artifacts. The documentation should be included in a high-level business requirement document that can be referenced when needed. A scrum-based approach of incremental delivery as opposed to waterfall-based approach where deliverables are cumulated entity is a preferred approach as per Hidalgo, E (2019). We would be following a scrum approach for our project, due to the short timeline of the project.

Budget – Estimated cost of procuring resources including servers, database instances, cost of man hours, and other details need to be published.

Finances – Project funding details along verticals and the support groups need to be identified and the budget allocation need to be finalized.

Scope – Items that are in scope and out of scope should be identified and shell stories that can be developed pre commencement of the project should be enclosed to the data analyst. The data analyst would then further engage with the business owner and the data developer the stories into items that can be implemented in an iterative manner.

Schedule - High level project timelines need to be document into a work breakdown structure (WBS) and/or agile backlogs. This document should be used as a guide to assist with delivery timelines. Any slippage in delivery should be documented and the timelines adjusted appropriately.

Analytic Approach –

The following methods are proposed to be used for our analysis:

1. **Logistic Regression:** systems to be drained using an 80:20 split to estimate the independent variable.
2. **Support vector machine**: supervise learning using 80:20 split to estimate the independent variable.
3. **Multi**-layered perceptron: Neural network using 80:20 split to estimate the independent variable.

Analytics Problem Statement

Perform a multiple regression analysis to predict the optimal spending limits of a subset of the population with a defined income to feed and sustain a predefined family size based on the average affordability ratio published by Food Affordability - Datasets - California Health and Human Services Open Data Portal

Data Understanding, Acquisition, and Preprocessing

Data Needs and Variables

For the purpose of this project, data pertaining to the average food related expenditure per annum, the median income, and the family size were determined to be the driving factors that influence an affordability index.

Data Obtained

The data was collected from the California department of public health portal published as food affordability, 2006-2010. The data is hosted in a public domain and is available without any additional cost. The data was provided as an excel table along with the data dictionary provided in the appendix. Various variables were removed in the process of cleaning the data and the records were trimmed to account for the undue influence of outliers.

Type of Data Variables

The dependent variables are entities that are influenced by the External factors and. in this instance was identified as cost_yr, median_annualIncome, Fam_size. The independent variable is a function that depends on the dependent variable cited above. And in this instance, we have chosen the affordability_index score as the independent variable. Changes in the dependent variables showed observed changes to our decision variables.

For each data variable, identify the specific type:

The following data variables were chosen from the data set for the analysis. Their data types and definitions have been included below:

Data Variable: cost_yr

Data type: Continuous

The annual cost of food is based on the USDA's low-cost food plan, which includes a market basket of items that families would have to purchase to provide a nutritious diet for each family member. To determine the costs, the USDA conducts a monthly national market basket survey of food items. The USDA tabulates per-person costs by age for children <11 years, and age and gender for those aged 12-71+ years. For this table, family costs were the sum of costs for the female head of household and the per child-cost multiplied by the area average number of children under 18 years of age, considering their age distribution.

Data Variable: median_income

Data type: Continuous

Median income in USD of female headed family w/children <18 years.

Data Variable: affordability_ratio

Data type: Continuous

Ratio of food cost to income, female headed family w/children <18 years

Data Variable: ave_fam_size :

Data type: Continuous

Average family size for a female headed family w/children <18 yrs, specific to a geography, all races combined.

Though the data set does not have a standard output as compared to more readily available data sets that can be used as a target variable, the data was chosen as a means to implement the solution on a real-world data set as opposed to using a test sample. An assumption was made that if the food affordability_ratio value is below 0.2 the family was deemed as being under the threshold for food security. Any value above that was considered as a secure situation for feeding its members. This was done by using bin sizes appropriately while analyzing. We expect a high variable inflation factor between cost_yr and median_income as the affordability index Is a ratio of these variables. In case this was found to be true we planned to eliminate one of these variables to adjust for overfitting of the module.

Data for Specific Analytics Problem

According to Glasmeier, A. K. (2004). The living wages for California residents can be established using the published calculator. Female households are poorer than male household according to research conducted by. Rajaram, R. (2009).

This available data can be applied to the specific analytics problem. Using this data, Our Company can use the data analytics solution to perform forecasting whether an individual can afford the food for their household or not within their monthly income. The obtained data included the relevant data points required for this project. The identified dependent variables and the effects on the affordability matrix have been recorded over 4 years. The duration also includes a period that had simulated undue hardship in the form of the 2009 market collapse that is similar to the pandemic. The developed data consist of 14364 records over the course of the years. Null values were found to be present in certain variables and were removed during the cleaning process. The practice of correcting or deleting incorrect, corrupted, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning according to Tableau.com. (n.d.). Datasets with similar characteristics have been used to assess the problems of similar nature in the past. For instance, the wine quality analysis data provided at https://www.kaggle.com/c/aiml-wine-quality-dataset has been analyzed using logistic regression and MLPC which is the same proposed modules planned for analysis.

Raw Data Excel File

Here is our raw data and the corresponding data dictionary.

food_afford_cdp_co_r
egion_ca4-14-13-ada.

foodaffordabilitydd.xl
sx

Collection of Initial Data

Data reliability is an important step in assuring the sanity of an analysis. Reliability indicates how consistent the measurements were obtained and the data validity indicates the accuracy of individual measurements taken. In this instance the data was sourced from the Californian Health and Human Services (CHHS), who in turn used data obtained from the US Department of Agriculture and US Census Bureau. CHHS is a USA government initiative program that was established to provide an open data portal that increased public access to California's most valuable assets – non-confidential health and human services data (Azuma et al., 2010). We verified and concluded that our data is reliable because we obtained it from a reputable source, the CHHS. The obtained data was further cleaned by removing records with null values before processing.

Description of Data and Data relationships

Figure 1.0 shows the relationship between the count of non-white Californian population indicted by one against the count of white Californian population presented in the data post-processing. The data was segmented to include a demographic consisting of income earning single mothers.

Figure 2.0 shows the average median income between the selected demographics. As observed the median among non-white population was lower than that of the white population.

Figure 3.0 shows the independent variable-affordability index and it relationship with the sub category races. Even though the median income for whites is higher than that of the non-white population, the affordability shows an inverse relationship. Various factors that could influence this relationship may include family size, age of the population, general spending habits, inherited wealth and other external factors. All these factors stated were considered while augmenting our dataset but was not factored in our model due to the short span of our capstone.

Figure 4.0 shows the relationship of average family size between the two ethnicities. As observed, we found no significant differences in the cumulative family sizes.

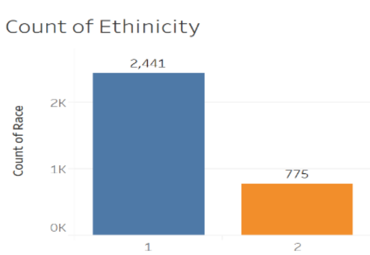


Figure 1.0

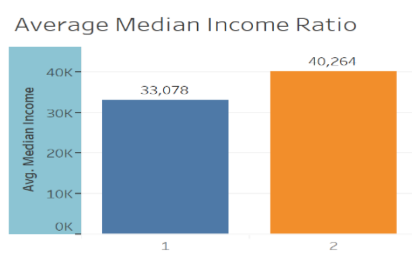


Figure 2.0

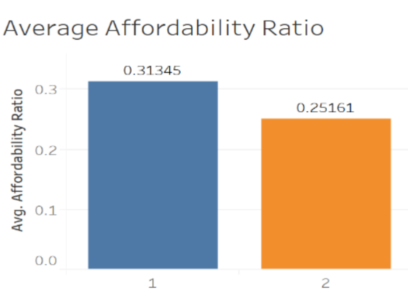


Figure 3.0

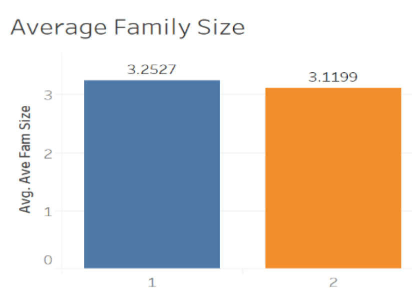


Figure 4.0

Exploration of Data

The trends of the data prior to the cleanup are as shown below in the figure. Outliers were observed that was heavily influencing and skewing the data points. These were primarily identified as a segment of the sample that had higher than the median income. The other segment identified include a sample of the population that was spending more than the median income. The data cleanup process involved removing both the segments outlined above.

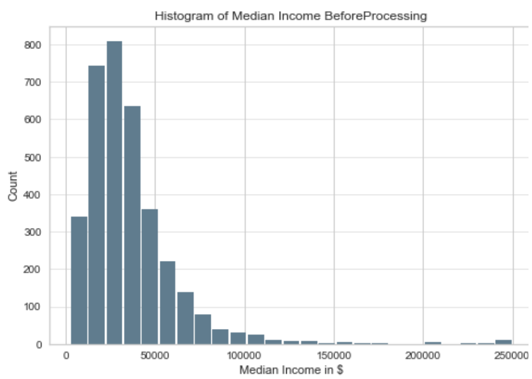


Figure 5.0

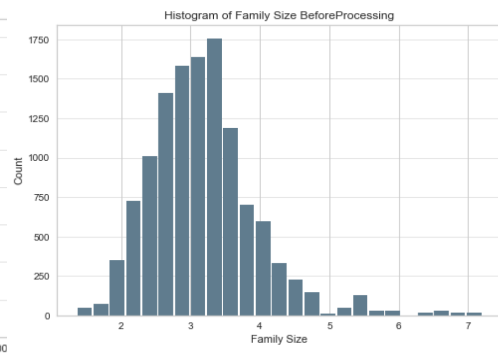


Figure 6.0

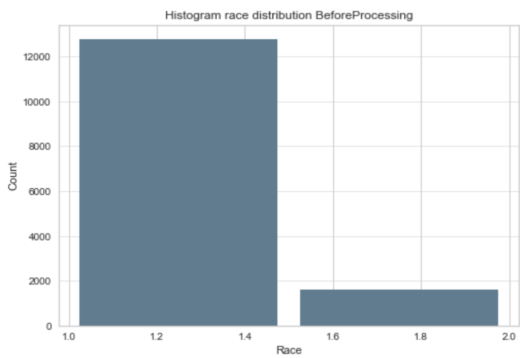


Figure 7.0

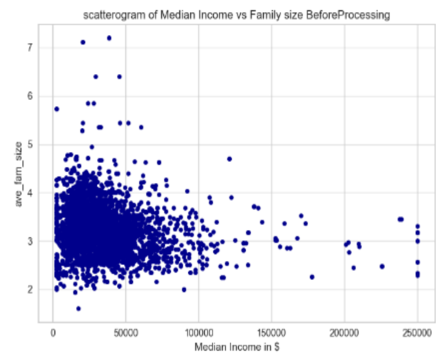


Figure 8.0

Verify Data Quality and Reliability of Data

Data reliability is an important step in assuring the sanity of an analysis. Reliability indicates how consistent the measurements were obtained and the data validity indicates the accuracy of individual measurements taken. In this instance the data was sourced from the Californian Health and Human Services (CHHS), who in turn used data obtained from the US Department of Agriculture and US Census Bureau. CHHS is a USA government initiative program that was established to provide an open data portal that increased public access to California's most valuable assets – non-confidential health and human services data (Azuma et al., 2010). We verified and concluded that our data is reliable because we obtained it from a reputable source, the CHHS. The obtained data was further cleaned by removing records with null values before processing.

We encountered a few challenges. The first challenged was establishing a business case suitable for our capstone project. Various artificial intelligence and machine learning models were explored. We decided to utilize the dataset provided by the Californian Health and Human Services based on availability and social relevance of the data.

The dataset provided by the Californian Health and Human Services included a time period where undue distress was observed by the majority of the population in a form of a global recession in 2009 similar to the unemployment crisis prevalent during the covid-19 global pandemic. Additional considerations include the factor that the data is less than a decade old, making it more relevant. The dataset includes relevant variables which impact food affordability as a measure against income and ethnicity. The inclusion of family size also adds value to the analysis.

The dataset was provided in an excel format along with a data dictionary that explains the context of each variables including units. The dataset was easily integrate-able with most currently available data processing tools such as Python-Panda-NumPy framework which was utilized in this instance. Consideration was given to augment the data by utilizing multiple data sources but due to the short span of the project this was not pursued. The final data after processing was found to be relevant for our analysis as it included major factors that influenced the food affordability of Californian residents.

Data Diagnostics and Descriptive Summary

Summary of data samples

Figure 9.0 shows the relationship between the count of non-white Californian population indicted by one against the count of white Californian population presented in the data post-processing. The data was segmented to include a demographic consisting of income earning single mothers.

Figure 10.0 shows the average median income between the selected demographics. As observed the median among non-white population was lower than that of the white population.

Figure 11.0 shows the independent variable-affordability index and it relationship with the sub category races. Even though the median income for whites is higher than that of the non-white population, the affordability shows an inverse relationship. Various factors that could influence this relationship may include family size, age of the population, general spending habits, inherited wealth and other external factors. All these factors stated were considered while augmenting our dataset but was not factored in our model due to the short span of our capstone.

Figure 12.0 shows the relationship of average family size between the two ethnicities. As observed, we found no significant differences in the cumulative family sizes.

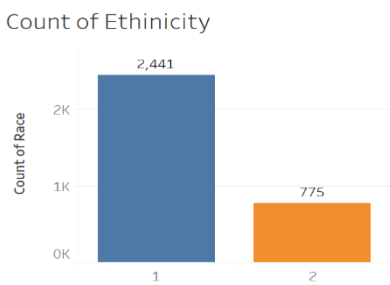


Figure 9.0

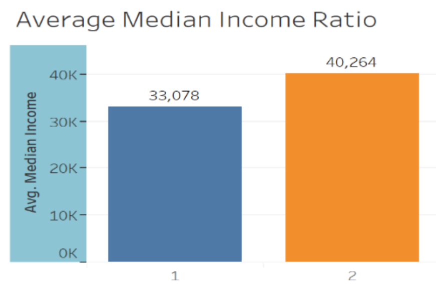


Figure 10.0

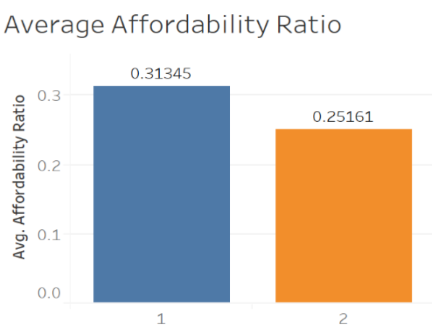


Figure 11.0

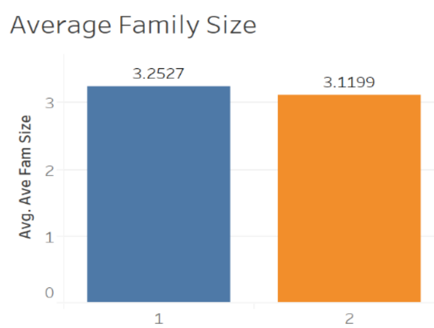


Figure 12.0

Exploratory Data Analysis

Quantitative approaches stress quantitative standards and statistical, analytical, or numeric assessment of information gathered through polls, questions, and surveys, or by altering pre-existing statistical data with computer tools (Stone et al., 2012).

Preliminary exploration of data was done on a total of 14,364 data records found in the initial data. Histograms shown below were generated along with a scatter plot. As indicated by the figures below, the dataset had a wide data range deviated from the median. We identified the outliers and eliminated them prior to the analysis to ensure the analysis was not overly influenced by the outliers.

Trends Analysis

Trend method is a measurement approach used in product design to forecast future outcomes using previous data. This is accomplished by keeping track of cost and schedule deviations. It is a product design product quality instrument in this case.

The table 1 below shows the correlation between the different variables. As observed, we see a high level of correlation between the affordability_ratio and median income, 0.762884. the rest of the variables do not display such a high level of relation.

	race_eth_code	median_income	affordability_ratio	ave_fam_size
race_eth_code	1.000000	0.154860	0.183194	-0.119783
median_income	0.154860	1.000000	0.762884	-0.208589
affordability_ratio	0.183194	0.762884	1.000000	-0.357839
ave_fam_size	-0.119783	-0.208589	-0.357839	1.000000

Table 1.0

The VIF (shown in figure 13) was analyzed for all four of our variables and was found to be within the threshold of 10. This indicates that multicollinearity is minimal in our dataset.

	feature	VIF
0	race_eth_code	8.337837
1	median_income	9.762107
2	affordability_ratio	4.203646
3	ave_fam_size	9.716444

Figure 13.0

We found a strong but negative relationship between the affordability ratio and family size as shown below. This is expected as the number of dependencies increases the cost of food should increase. Which would inevitably decrease the affordability ratio.

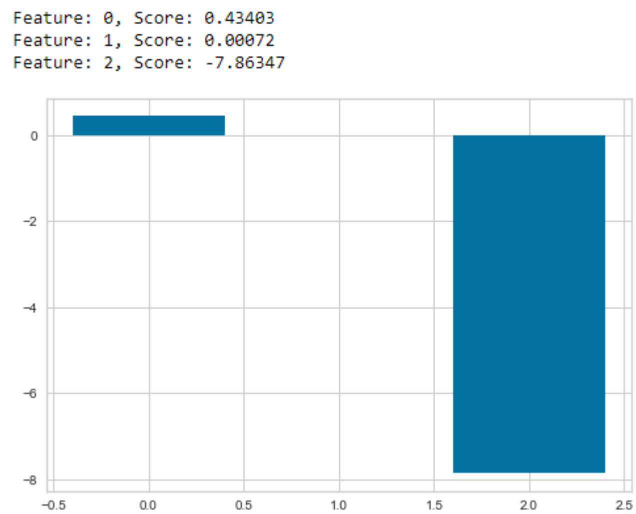


Figure 14.0

The only two variables which showed a small level of linearity for median income and ethnicity code. However, as ethnicity code is a nominal variable indicating the bin size of our demographics. This analysis does not add value to our research.

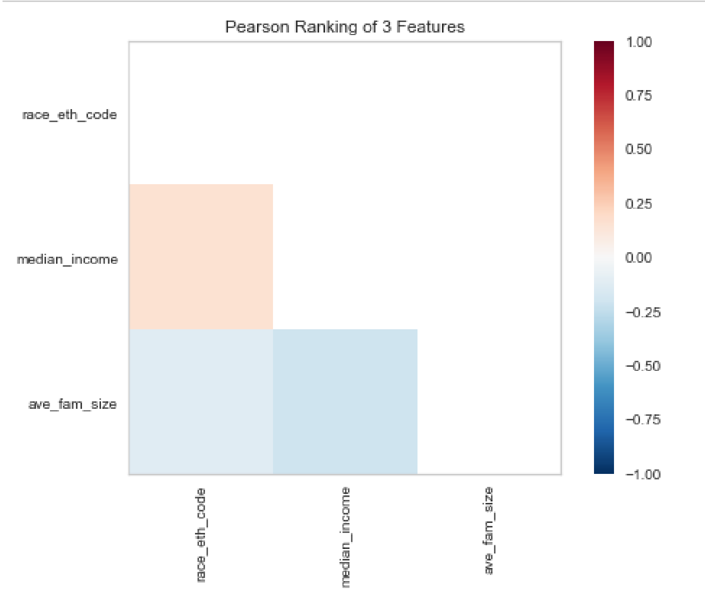


Figure 15.0

Simpson Paradox

The Simpson's Paradox is a known phenomenon which occurs in statistics, where associations between variables in a population emerges, disappears, or reverses when the population is divided into sub-populations (Sprenger et al., n.d.).

The dataset obtained from the sources consisted of multiple variables. however, during the processing stage the data was trimmed to include only three dependent variables which influenced the fourth independent variable. Preliminary correlation analysis was done including Pearson coefficient for linearity and variable inflation factor. Linearity was established between two variables, median income and affordability ratio. The variable influential factor was found to be less than ten in all cases. The observations above, strongly established that multiple collinearities do not exist within the dataset chosen.

Thus, that led us to believe the Simpson’s paradox was not applicable to our data because our study will not involve comparisons of inferences drawn across different explanatory levels. All comparisons will be done between individuals of groups and not between a subgroup and a major group to avoid any instances of Simpson’s paradox if it were to manifest.

Descriptive Analytics

In the perceptual study of products, descriptive analysis is a complex concept. It has progressed from expert analysis to a more solid and scientific method to assessing perceptions (Stone et al., 2012).

We summarized our data by using histograms, bar charts and scatter plots mainly because the two-dimension nature of the data variables. other characteristics like Pearson coefficient and confusion matrices would be represented using heat maps or appropriate graphical means. The analysis would also include descriptive summary data as well as the head of datasets represented in tabular format.

The only variable that was transformed was the categorical variable ‘ethnicity code’. This was done as a part of data segmentation non-white were assigned a value of 1 and white ethnicities were assigned a value of 2. No additional steps were performed to establish the validity of the dataset as the data was obtained from an authoritative department of the USA government.

Conclusion

The study aimed to find the food affordability for Californian residents based on median income, ethnicity and average family size. From the preliminary analysis of the data, we found a correlation of affordability index with median income. We did not find any significant differences in the average family size of the two ethnic segments analyzed. We also uncovered that in spite of the median income being greater for the white ethnicity their food affordability index remained significantly lower for the demographics. Various factors that could influence this relationship may include family size, age of the population, general spending habits, inherited wealth, and other external factors. All these factors stated were considered while augmenting our dataset but was not factored in our model due to the short span of our capstone.

Furthermore, there was a significant number of missing data points. About 14,364 unique records were present in the initial dataset. This was trimmed to about 3217 records mainly due to the removal of null values. We also removed any data row that deviates more than 3 times the standard deviation from the mean was deemed an outlier and eliminated. Approximately 80% of the data was removed to ensure that the dataset contained consistent values across each record entries.

Methodology Approach and Model Building

Multiple machine learning and artificial intelligence models are currently available. The focus of this analysis will be on the ones pertaining to regression with continuous dependent variables. The primary candidates chosen for this analysis are as listed below along with the rational for choosing each individual model

Modeling Methods

- Logistic Regression as explained by Kleinbaum, D.G., and Klein, M. (2002) is “a mathematical modeling approach that can be used to describe the relationship of several X's to a dichotomous dependent variable, such as D”. In this instance the logistic regression model was a prime candidate as it is easy to implement use and train (Tu J.V., 1996).
- Random Forest Random Forest classifier is a meta estimator for decision trees of various sub-samples and improves predictive accuracy by utilizing averaging and control overfitting. This method affords us the ability to have control over the max_sampleset (Pedregosa, F., et al., 2011).
- Support Vector Machines: Noble, W. S. (2006) states that Support vector machine (SVM) is a supervised learning algorithm which learns by example to assign labels to objects. SVM was chosen in this instance as there is a clear line of margin between the ability to afford or not afford food.
- Multilayer perceptron provides quick predictions after training with higher accuracy rates. As stated by Fandango, A. (2018), when multiple neurons are connected such that output of one-layer feeds in as the input of the next layer sequentially until the output of the final layer becomes the final output are called as feed forward neural network (FFNN). As FFNNS are made up of individual neurons joined together they are also called MultiLayer Perceptrons (MLP) or deep neural networks (DNN).

Test Design

The following test and train design was implemented. The procured data set was split into 2 major segments. A random seed was used to split the data into two parts 80 % training data and 20 % testing data. Due to the data limitation a validation segment was not implemented. Tests that were performed on the test data were analyze confusion metric for mislabeling and overall accuracy score based on the training.

The following processes were involved in ensuring the sanity of the test data.

- Initial procurement of raw data from a credible source
- Validate data consistency and accuracy: The initial feed file was analyzed for records with null values and other inconsistencies. These values were removed as a part of the cleaning process
- Outliers were eliminated as a part of the boundary analysis. Data ranges were confined to mean +/- (3 * standard deviation). Any values outside this range were removed.
- Variable selection: The appropriate variables that were essential for the analysis was selected. The criteria used in selection includes, removal of categorical variables, removal of variables that introduced high Variance Inflation Factor (VIF), correlation tests were used to establish relationship among variables, Process Control Monitoring (PCM) tests was run to establish the relevance of selected variables.
- Split data into test and train segments using random seed
- The training set was used to train the various models
- Post training the Testing set was used to analyze that that dependent variable was not misclassified

- Confusion matrix was obtained to score the level of misclassification between models
- The data from the confusion matrix was used to provide an overall accuracy score

Model Building

The default configurations were used for SVC and Logistic Regression as it did not require additional configurations to be added.

For Random Forest classifier the n_estimator was set to 200. The n_estimator signifies the maximum number of trees that would be spawned. The higher the number of trees the greater the accuracy of the model. However, as the number of trees increases the performance of the model reduces. A range from 100 to a 1000 was tested and an optimal number of trees was found to be 200 based on the docker instance specifications.

For the MLP classifier the number of hidden layers was set to {11, 11, 11} with a maximum iteration limit of 500. The hidden_layer parameter sets the number of layers and the number of nodes in the Neural Network Classifier and was set as stated above. The random seed state was set to 1 to reproduce the test results during the analysis but has been implemented as a configurable entity.

For this analysis we utilized the scikit-learn classification and neural network libraries. The libraries are available as an opensource implementation and provides robust artificial intelligence as well as machine learning resources.

The following models were imported from scikit-learn libraries for the analysis

- sklearn import svm
- sklearn.linear_model import LogisticRegression, LinearRegression

- sklearn.ensemble import RandomForestClassifier
- sklearn.neural_network import MLPClassifier

The following support libraries were also imported for analysis

- sklearn.model_selection import train_test_split
- sklearn.preprocessing import LabelEncoder, StandardScaler
- sklearn.decomposition import PCA
- sklearn.naive_bayes import GaussianNB
- statsmodels.stats.outliers_influence import variance_inflation_factor
    - sklearn.metrics import confusion_matrix, classification_report, accuracy_score

A dictionary file was created which initiated objects of the various models. An iterator was used to iterate through the various listed objects within the dictionary file. Methods within the iterator performed the training and testing functions. A separate method within the iterator also provided a graphical output of the confusion matrix along with the accuracy score.

The outputs from the iterator were analyzed manually to find the model with the least amount of misclassification.

Description of Variables included

The variables race_eth_name, median_income, ave_fam_size was chosen as the dependent variables and affordability_ratio was chosen as the target variable. ind_id, ind_definition, reportyear and version were found to not add any value to the analysis and were not included in the analysis. cost_yr was found to be direct function of affordability_ratio and median_income, this was leading to very high variable inflation factor and hence was dropped. Other string variables were dropped as they were found non compatible with the model chosen. Careful consideration was also given to other numeric variables that was found to not add value but increase the complexity of the model and hence were removed.

Table 2.0 below shows the variables used and dropped during the analysis along with the definition of the variables and their format.

Variable	Definition	Format	Usage
ind_id	Indicator ID	String	Dropped
ind_definition	Definition of indicator in plain language	String	Dropped
reportyear	Year(s) that the indicator was reported	String	Dropped
race_eth_code	numeric code for a race/ethnicity group	String	Dependent Variable
race_eth_name	Name of race/ethnic group	String	Dropped

geotype	Type of geographic unit	String	Dropped
geotypevalue	Value of geographic unit	String	Dropped
geoname	Name of geographic unit	String	Dropped
county_name	Name of county that geotype is in	Plain Text	Dropped
county_fips	FIPS code of county that geotype is in	Plain Text	Dropped
region_name	Metropolitan Planning Organization (MPO)-based region name: see MPO_County List Tab	Plain Text	Dropped
region_code	Metropolitan Planning	Plain Text	Dropped

	Organization (MPO)-based region code: see MPO_CountyList tab		
cost_yr	Annual food costs	Numeric	Dropped
median_income	Median income	Numeric	Dependent Variable
affordability_ratio	Ratio of food cost to income, female headed family w/children <18 yrs	Numeric	Target Variable
LL_95CI	Lower limit of 95% confidence interval	Numeric	Dropped
UL_95CI	Upper limit of 95% confidence interval	Numeric	Dropped

se_food_afford	Standard error of percent	Numeric	Dropped
rse_food_afford	Relative standard error (se/percent * 100) expressed as a percent	Numeric	Dropped
CA_decile	California decile	Numeric	Dropped
CA_RR	Rate ratio to California rate	Numeric	Dropped
ave_fam_size	Average family size for a female headed family w/children <18 yrs, specific to a geography, all races combined	Numeric	Dependent Variable
version	Date/time stamp of version of data	Date/Time	Dropped

Table 2.0

Screenshots of Working Model

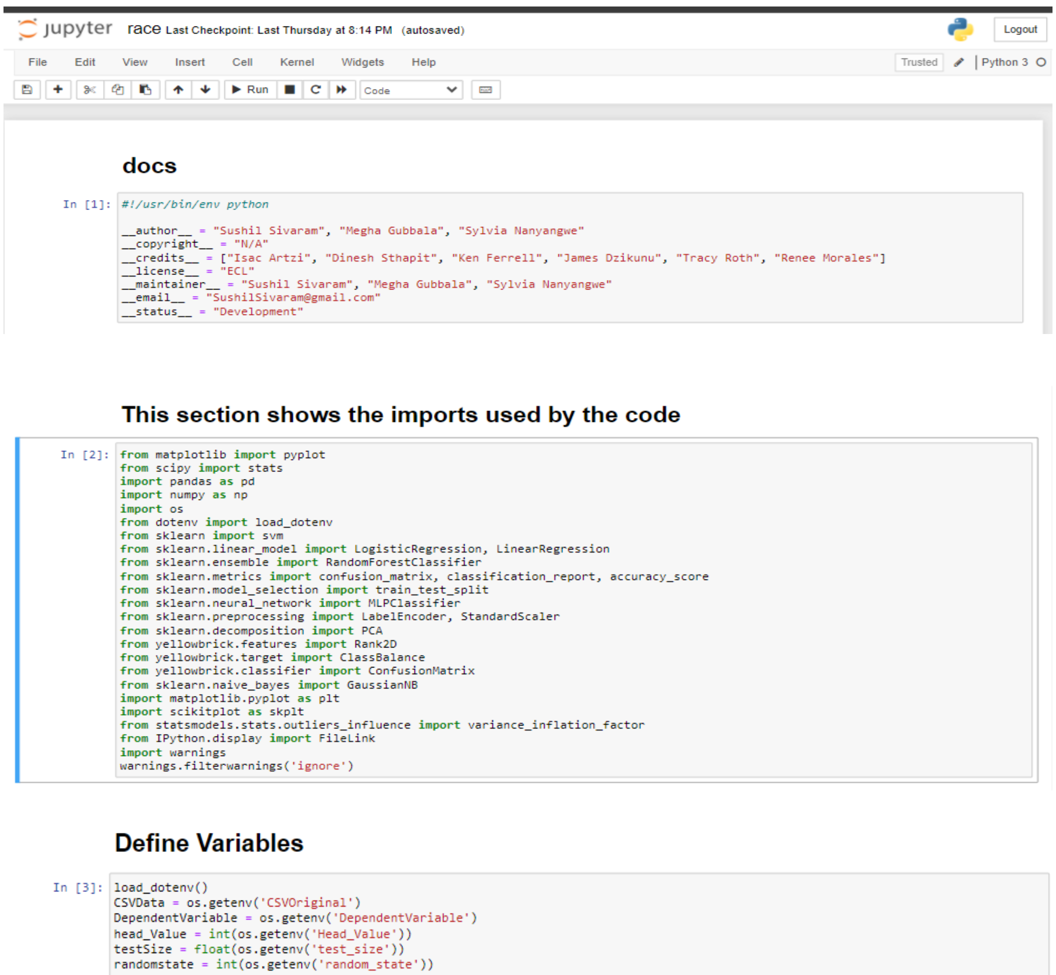


Figure 16.0

Setup Reusable Functions

In [4]:

```python
# Load Data from CSV
dataSetUp = []
keepcolumns = ['race_eth_code', 'median_income', 'affordability_ratio', 'ave_fam_size']
datasetupUnprocessed = []


def loadAndExtractData():
    def readCSV():
        global datasetupUnprocessed
        dataSetUp = pd.read_csv(CSVData)
        datasetupUnprocessed = dataSetUp
        display(datasetupUnprocessed.corr())
        dataout = datasetupUnprocessed.corr()
        dataout.to_csv('dataout.csv')
        display(FileLink('dataout.csv'))
        return dataSetUp

    def dropVariables():
        dataSetUp = readCSV()
        dataSetUp = dataSetUp.filter(keepcolumns)
        return dataSetUp

    def removeNullValues():
        dataSetUp = dropVariables()
        for keep in keepcolumns:
            dataSetUp = dataSetUp[dataSetUp[keep].notna()]
        return dataSetUp

    def removeOutliers():
        global dataSetUp
        dataSetUp = removeNullValues()
        dataSetUp = dataSetUp[(np.abs(stats.zscore(dataSetUp)) < 3).all(axis=1)]
        dataSetUp.to_csv('cleaned.csv', index=False)
        print(FileLink('cleaned.csv'))
        print(dataSetUp.shape)
        return dataSetUp
    dataSetUp = removeOutliers()

def checkforPCA():
    #dataSetUp = removeOutliers()
    pca = PCA()
    X = dataSetUp.drop(DependentVariable, axis=1)
    y = dataSetUp[DependentVariable]
    x_pca = pca.fit_transform(X)
    x_pca = pd.DataFrame(x_pca)
    datapca = x_pca.head()
    datapca.to_csv('datapca.csv')
    print(FileLink('datapca.csv'))
    return dataSetUp

# print Info
def showDataHeadAndInfo(data, headCount):
    print(f"showing head {headCount} values")
    print(data.head(headCount))
    print("**********")
    print("Showing info of dataset")
    print(data.describe(include='all'))


# preProcessing
def preProcessing():
    bins = (0, .2, 5)
    group_names = ['Cant Afford', 'Can Afford']
    dataSetUp[DependentVariable] = pd.cut(dataSetUp[DependentVariable], bins, labels=group_names)
    dataSetUp.to_csv('test.csv')
    label_quality = LabelEncoder()
    dataSetUp[DependentVariable] = label_quality.fit_transform(dataSetUp[DependentVariable])
    # showDataHeadAndInfo(head_Value)
    print(dataSetUp[DependentVariable].value_counts())
```

```python
# plotting
def plotting(dataSetUp, state):
    plt.figure()
    histmedian_income = dataSetUp['median_income'].plot.hist(bins=25, grid=True, rwidth=0.9, color='#607c8e')
    plt.title(f'Histogram of Median Income {state}')
    plt.xlabel('Median Income in $')
    plt.ylabel('Count')
    plt.grid(axis='y', alpha=0.5)
    histmedian_income.figure.savefig(f'.\outputs\histMedianIncome{state}.png')

    plt.figure()
    hist_avg_fam = dataSetUp['ave_fam_size'].plot.hist(bins=25, grid=True, rwidth=0.9, color='#607c8e')
    plt.title(f'Histogram of Family Size {state}')
    plt.xlabel('Family Size')
    plt.ylabel('Count')
    plt.grid(axis='y', alpha=0.5)
    hist_avg_fam.figure.savefig(f'.\outputs\histavgFamsize{state}.png')

    plt.figure()
    hist_race_eth_name = dataSetUp['race_eth_code'].plot.hist(bins=2, grid=True, rwidth=0.9, color='#607c8e')
    plt.title(f'Histogram race distribution {state}')
    plt.xlabel('Race')
    plt.ylabel('Count')
    plt.grid(axis='y', alpha=0.5)
    hist_race_eth_name.figure.savefig(f'.\outputs\histCost{state}.png')

    plt.figure()
    scattermedian_income = dataSetUp.plot.scatter(c='DarkBlue', x='median_income', y='ave_fam_size')
    plt.title(f'scatterogram of Median Income vs Expenditure {state}')
    plt.xlabel('Median Income in $')
    plt.ylabel('ave_fam_size')
    plt.grid(axis='y', alpha=0.5)
    scattermedian_income.figure.savefig(f'.\outputs\scatterMedianIncomeVSFamilySize{state}.png')

    plt.figure()


# scattermedian_income = dataSetUp.plot.scatter(c='DarkBlue', x='ave_fam_size', y = 'race_eth_code' )
# plt.title(f'scatterogram of Family Size vs Expenditure {state}')
# plt.xlabel('Family Size')
# plt.ylabel('race_eth_code')
# plt.grid(axis='y', alpha=0.5)
# scattermedian_income.figure.savefig(f'.\outputs\scatterFamSizeVSExpenditure{state}.png')
# plt.show()
```

Figure 17.0

Load Data from CSV

In [5]:
```
loadAndExtractData()
```

	ind_id	race_eth_code	geotypevalue	county_fips	region_code	cost_yr	median_income	affordability_ratio	LL95_affordability_
ind_id	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
race_eth_code	NaN	1.000000e+00	-9.391805e-15	-8.519700e-16	-6.967506e-16	-5.438350e-16	0.138297	-0.071037	-0.11
geotypevalue	NaN	-9.391805e-15	1.000000e+00	7.375404e-02	5.456655e-02	4.041436e-02	0.055404	0.017357	-0.04
county_fips	NaN	-8.519700e-16	7.375404e-02	1.000000e+00	1.334657e-01	-6.614141e-02	-0.027582	0.038356	-0.03
region_code	NaN	-6.967506e-16	5.456655e-02	1.334657e-01	1.000000e+00	2.819352e-01	-0.081700	0.073701	0.11
cost_yr	NaN	-5.438350e-16	4.041436e-02	-6.614141e-02	2.819352e-01	1.000000e+00	-0.091045	0.135293	0.23
median_income	NaN	1.382971e-01	5.540431e-02	-2.758242e-02	-8.170001e-02	-9.104497e-02	1.000000	-0.443224	-0.30
affordability_ratio	NaN	-7.103725e-02	1.735681e-02	3.835569e-02	7.370095e-02	1.352932e-01	-0.443224	1.000000	0.18
LL95_affordability_ratio	NaN	-1.164198e-01	-4.586301e-02	-3.020901e-02	1.151732e-01	2.363188e-01	-0.301998	0.188805	1.00
UL95_affordability_ratio	NaN	-1.762852e-02	1.331955e-02	-8.442782e-05	-1.914618e-03	-1.388046e-02	-0.207914	0.540766	-0.11
se_food_afford	NaN	-1.005539e-02	1.509296e-02	1.011063e-03	-8.380012e-03	-3.132648e-02	-0.168264	0.482920	-0.13
rse_food_afford	NaN	-5.042998e-03	3.510456e-02	9.772263e-04	-4.037071e-02	-7.738858e-02	-0.134449	0.337438	-0.27
CA_decile	NaN	NaN	5.215403e-03	-1.681248e-02	-1.553850e-01	-3.558709e-01	0.728487	-0.619112	-0.40
CA_RR	NaN	-7.103724e-02	1.735681e-02	3.835569e-02	7.370095e-02	1.352932e-01	-0.443224	1.000000	0.18
ave_fam_size	NaN	-7.896376e-16	3.332340e-02	-5.812496e-02	2.176299e-01	9.635995e-01	-0.172228	0.179759	0.26

dataout.csv

```
C:\Project\Capstone\cleaned.csv
(3216, 4)
```

```
def trainDataset():
    global X_train
    global y_train
    global X_test
    global y_test
    global sc
    # Train and test with random seed
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=testSize, random_state=randomstate)
    # Optimizing with standardScaler to minimize bias and normalize values
    sc = StandardScaler()
    X_train = sc.fit_transform(X_train)
    X_test = sc.transform(X_test)


def predictorImportance():
    X = dataSetUp.drop(DependentVariable, axis=1)
    y = dataSetUp[DependentVariable]
    model = LogisticRegression()
    # fit the model
    model.fit(X, y)
    importance = model.coef_[0]
    # summarize feature importance
    for i, v in enumerate(importance):
        print('Feature: %0d, Score: %.5f' % (i, v))
    # plot feature importance
    pyplot.bar([x for x in range(len(importance))], importance)
    pyplot.show()

    # VIF dataframe


def vifCheck(dataSetUp):
    vif_data = pd.DataFrame()
    vif_data["feature"] = dataSetUp.columns
    vif_data["VIF"] = [variance_inflation_factor(dataSetUp.values, i)
                       for i in range(len(dataSetUp.columns))]
    print(vif_data)
```

Figure 18.0

preProcessing

In [9]:
```
preProcessing()
showDataHeadAndInfo(dataSetUp,head_Value)
```

```
0    1997
1    1219
Name: affordability_ratio, dtype: int64
showing head 15 values
    race_eth_code  median_income  affordability_ratio  ave_fam_size
0               1        23777.0                    0          3.34
1               1        38508.0                    1          3.34
2               1        26192.0                    0          3.34
3               1        22858.0                    0          3.34
4               1        36737.0                    0          3.34
5               2        38641.0                    1          3.34
6               1        32866.0                    0          3.34
7               1        30439.0                    0          3.34
8               1        28184.0                    0          3.34
9               1        16063.0                    0          3.21
10              1        42048.0                    1          3.21
11              1        23858.0                    0          3.21
12              1        28917.0                    0          3.21
13              1        35238.0                    0          3.21
14              2        50497.0                    1          3.21
**********
Showing info of dataset
       race_eth_code  median_income  affordability_ratio  ave_fam_size
count    3216.000000    3216.000000          3216.000000   3216.000000
mean        1.240983   34809.645833             0.379042      3.220725
std         0.427746   19849.167081             0.485224      0.474311
min         1.000000    2500.000000             0.000000      1.940000
25%         1.000000   20740.500000             0.000000      2.900000
50%         1.000000   30801.000000             0.000000      3.220000
75%         1.000000   44216.500000             1.000000      3.500000
max         2.000000  119342.000000             1.000000      4.790000
```

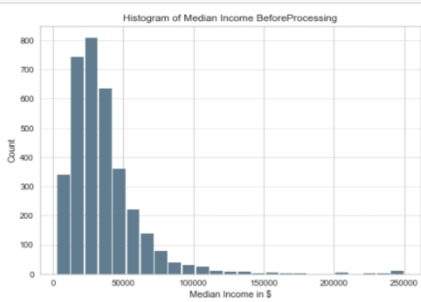
Figure 19.0

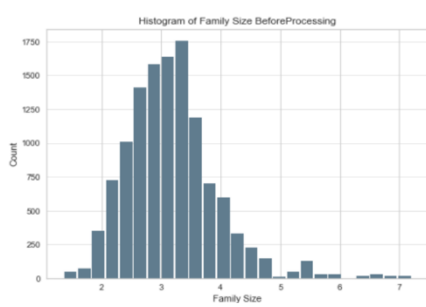
```
    LL95_affordability_ratio  UL95_affordability_ratio  se_food_afford  \
0                   0.231517                  0.400043        0.042991
1                   0.183065                  0.206895        0.006079
2                   0.279661                  0.293666        0.003573
3                   0.322637                  0.334314        0.002979
4                   0.173762                  0.234997        0.015621
5                   0.189570                  0.199048        0.002418
6                   0.210008                  0.246896        0.009410
7                   0.176559                  0.316775        0.035769
8                   0.262832                  0.269973        0.001821
9                   0.087869                  0.813846        0.185198
10                  0.121394                  0.223075        0.025939
11                  0.266850                  0.340253        0.018725
12                  0.214028                  0.286862        0.018580
13                  0.124903                  0.286138        0.041131
14                  0.134053                  0.152780        0.004777

    rse_food_afford  CA_decile     CA_RR  ave_fam_size         version
0         13.614342        NaN  1.185347          3.34  4/12/2013 4:33
1          3.117814        NaN  0.731900          3.34  4/12/2013 4:33
2          1.246349        NaN  1.076054          3.34  4/12/2013 4:33
3          0.906881        NaN  1.233004          3.34  4/12/2013 4:33
4          7.643255        NaN  0.767183          3.34  4/12/2013 4:33
5          1.244406        NaN  0.729381          3.34  4/12/2013 4:33
6          4.119149        NaN  0.857543          3.34  4/12/2013 4:33
7         14.501074        NaN  0.925917          3.34  4/12/2013 4:33
8          0.683740        NaN  1.000000          3.34  4/12/2013 4:33
9         41.076862        NaN  1.692392          3.21  4/12/2013 4:33
10        15.060224        NaN  0.646520          3.21  4/12/2013 4:33
11         6.168717        NaN  1.139445          3.21  4/12/2013 4:33
12         7.418781        NaN  0.940101          3.21  4/12/2013 4:33
13        20.013280        NaN  0.771465          3.21  4/12/2013 4:33
14         3.331023        NaN  0.538346          3.21  4/12/2013 4:33

[15 rows x 23 columns]
**********
Showing info of dataset
         ind_id                                     ind_definition reportyear  \
count   14364.0                                              14364      14364
unique      NaN                                                  1          1
top         NaN  Food affordability for female-headed household...  2006-2010
freq        NaN                                              14364      14364
mean      757.0                                                NaN        NaN
std         0.0                                                NaN        NaN
min       757.0                                                NaN        NaN
25%       757.0                                                NaN        NaN
50%       757.0                                                NaN        NaN
75%       757.0                                                NaN        NaN
max       757.0                                                NaN        NaN
```

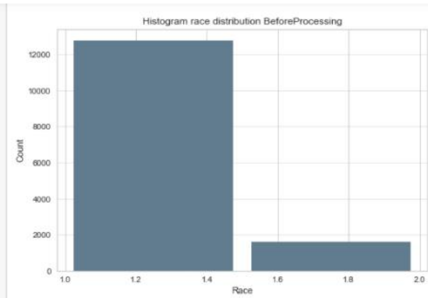
```
        race_eth_code race_eth_name geotype  geotypevalue          geoname  \
count    14364.000000         14364   14364  14364.000000            14364
unique            NaN             9       4           NaN             1581
top               NaN         NHOPI      PL           NaN  El Sobrante CDP
freq              NaN          1596   13707           NaN               18
mean         1.111111           NaN     NaN  40680.393484              NaN
std          0.314281           NaN     NaN  25834.492705              NaN
min          1.000000           NaN     NaN      1.000000              NaN
25%          1.000000           NaN     NaN  17480.500000              NaN
50%          1.000000           NaN     NaN  40382.000000              NaN
75%          1.000000           NaN     NaN  60609.500000              NaN
max          2.000000           NaN     NaN  87090.000000              NaN

        county_name   county_fips  ...  median_income  affordability_ratio  \
count         14229  14229.000000  ...    3473.000000          3473.000000
unique           58           NaN  ...            NaN                  NaN
top     Los Angeles           NaN  ...            NaN                  NaN
freq           1278           NaN  ...            NaN                  NaN
mean            NaN   6057.977862  ...   35985.685081             0.357114
std             NaN     31.048709  ...   27436.558125             0.451169
min             NaN   6001.000000  ...    2500.000000             0.021258
25%             NaN   6035.000000  ...   20219.000000             0.158028
50%             NaN   6059.000000  ...   30371.000000             0.245429
75%             NaN   6083.000000  ...   44083.000000             0.381940
max             NaN   6115.000000  ...  250000.000000             4.852371

        LL95_affordability_ratio  UL95_affordability_ratio  se_food_afford  \
count                3285.000000               3285.000000     3285.000000
unique                       NaN                       NaN             NaN
top                          NaN                       NaN             NaN
freq                         NaN                       NaN             NaN
mean                    0.105307                  0.882108        0.295230
std                     0.117439                  3.200605        1.568641
min                     0.000000                  0.041421        0.000903
25%                     0.000000                  0.239877        0.029374
50%                     0.077821                  0.381348        0.063821
75%                     0.165909                  0.658876        0.154163
max                     0.850556                108.783721       54.965397

        rse_food_afford   CA_decile        CA_RR  ave_fam_size         version
count       3285.000000  960.000000  3473.000000  12096.000000           14364
unique              NaN         NaN          NaN           NaN               1
top                 NaN         NaN          NaN           NaN  4/12/2013 4:33
freq                NaN         NaN          NaN           NaN           14364
mean          59.221472    5.500000     1.340507      3.175714             NaN
std          139.191814    2.873778     1.693561      0.762813             NaN
min            0.683740    1.000000     0.079797      1.360000             NaN
25%           14.351806    3.000000     0.593193      2.660000             NaN
50%           30.083705    5.500000     0.921273      3.130000             NaN
75%           61.209242    8.000000     1.433696      3.550000             NaN
max         5227.123987   10.000000    18.214432      7.200000             NaN

[11 rows x 23 columns]
```

Exploratory plotting

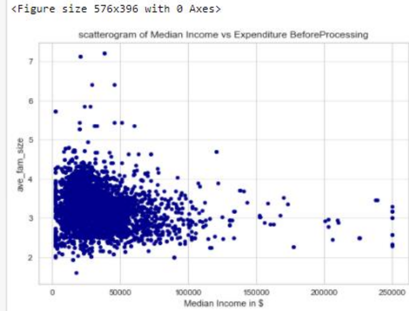
```
In [8]: plotting(datasetupUnprocessed , "BeforeProcessing")
```







<Figure size 576x396 with 0 Axes>



<Figure size 576x396 with 0 Axes>

Figure 20.0

Check Corellation

In [11]:
```
dataSetUp.corr()
```

Out[11]:

	race_eth_code	median_income	affordability_ratio	ave_fam_size
race_eth_code	1.000000	0.154860	0.183194	-0.119783
median_income	0.154860	1.000000	0.762884	-0.208589
affordability_ratio	0.183194	0.762884	1.000000	-0.357839
ave_fam_size	-0.119783	-0.208589	-0.357839	1.000000

Check VIF

In [12]:
```
vifCheck(dataSetUp)
```

```
             feature       VIF
0       race_eth_code  8.337837
1       median_income  9.762107
2  affordability_ratio  4.203646
3        ave_fam_size  9.716444
```

Predictor Importance

In [13]:
```
predictorImportance()
```

```
Feature: 0, Score: 0.43403
Feature: 1, Score: 0.00072
Feature: 2, Score: -7.86347
```

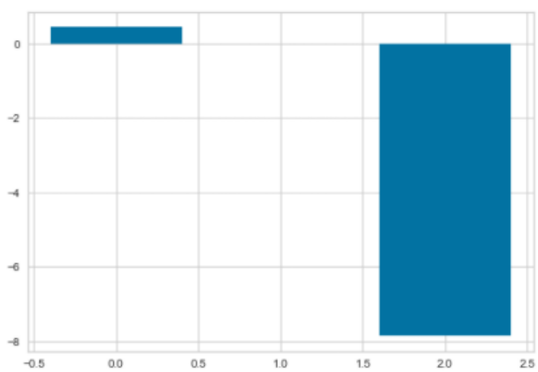


Figure 21.0

check for PCA

In [6]:
```
checkforPCA()
```

```
C:\Project\Capstone\datapca.csv
```

Out[6]:

	race_eth_code	median_income	affordability_ratio	ave_fam_size
0	1	23777.0	0.315779	3.34
1	1	38508.0	0.194980	3.34
2	1	26192.0	0.286664	3.34
3	1	22858.0	0.328475	3.34
4	1	36737.0	0.204379	3.34
...	...	...	...	...
14224	1	13233.0	0.575954	3.21
14226	2	19381.0	0.393251	3.21
14229	1	18893.0	0.403409	3.21
14235	2	67813.0	0.109646	3.07
14238	1	67813.0	0.109646	3.07

3216 rows × 4 columns

Print Info

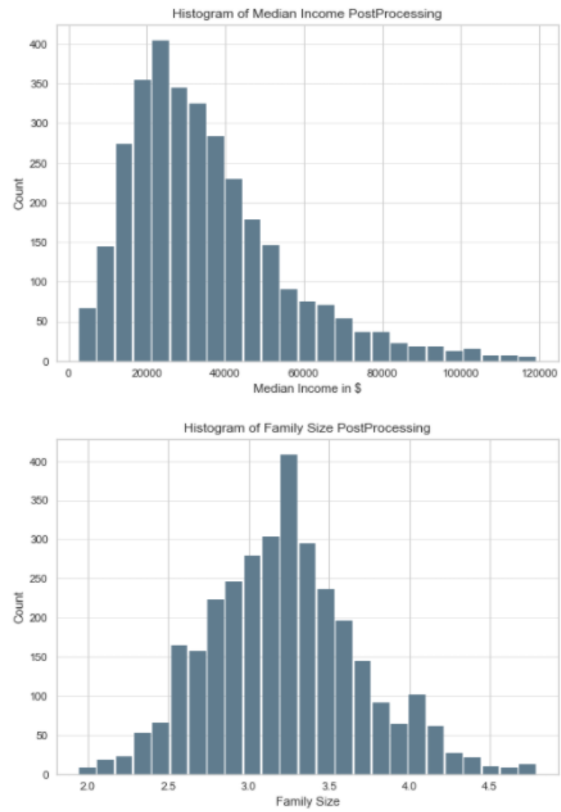
In [7]:
```
showDataHeadAndInfo(datasetupUnprocessed,head_Value)
```

```
showing head 15 values
    ind_id                                    ind_definition reportyear  \
0      757  Food affordability for female-headed household...  2006-2010
1      757  Food affordability for female-headed household...  2006-2010
2      757  Food affordability for female-headed household...  2006-2010
3      757  Food affordability for female-headed household...  2006-2010
4      757  Food affordability for female-headed household...  2006-2010
5      757  Food affordability for female-headed household...  2006-2010
6      757  Food affordability for female-headed household...  2006-2010
7      757  Food affordability for female-headed household...  2006-2010
8      757  Food affordability for female-headed household...  2006-2010
9      757  Food affordability for female-headed household...  2006-2010
10     757  Food affordability for female-headed household...  2006-2010
11     757  Food affordability for female-headed household...  2006-2010
12     757  Food affordability for female-headed household...  2006-2010
13     757  Food affordability for female-headed household...  2006-2010
14     757  Food affordability for female-headed household...  2006-2010

    race_eth_code race_eth_name geotype  geotypevalue     geoname county_name  \
0               1          AIAN      CA             6  California         NaN
1               1         Asian      CA             6  California         NaN
2               1     AfricanAm      CA             6  California         NaN
3               1        Latino      CA             6  California         NaN
4               1         NHOPI      CA             6  California         NaN
5               2         White      CA             6  California         NaN
6               1      Multiple      CA             6  California         NaN
7               1         Other      CA             6  California         NaN
8               1         Total      CA             6  California         NaN
9               1          AIAN      CO          6001     Alameda     Alameda
10              1         Asian      CO          6001     Alameda     Alameda
11              1     AfricanAm      CO          6001     Alameda     Alameda
12              1        Latino      CO          6001     Alameda     Alameda
13              1         NHOPI      CO          6001     Alameda     Alameda
14              2         White      CO          6001     Alameda     Alameda

    county_fips  ... median_income  affordability_ratio  \
0           NaN  ...       23777.0             0.315779
1           NaN  ...       38508.0             0.194980
2           NaN  ...       26192.0             0.286664
3           NaN  ...       22858.0             0.328475
4           NaN  ...       36737.0             0.204379
5           NaN  ...       38641.0             0.194309
6           NaN  ...       32866.0             0.228452
7           NaN  ...       30439.0             0.246667
8           NaN  ...       28184.0             0.266403
9        6001.0  ...       16063.0             0.450857
10       6001.0  ...       42048.0             0.172235
11       6001.0  ...       23858.0             0.303551
12       6001.0  ...       28917.0             0.250445
13       6001.0  ...       35238.0             0.205520
14       6001.0  ...       50497.0             0.143417
```

Figure 22.0

Plotting post Cleanup

```
In [10]: plotting(dataSetUp, "PostProcessing")
```



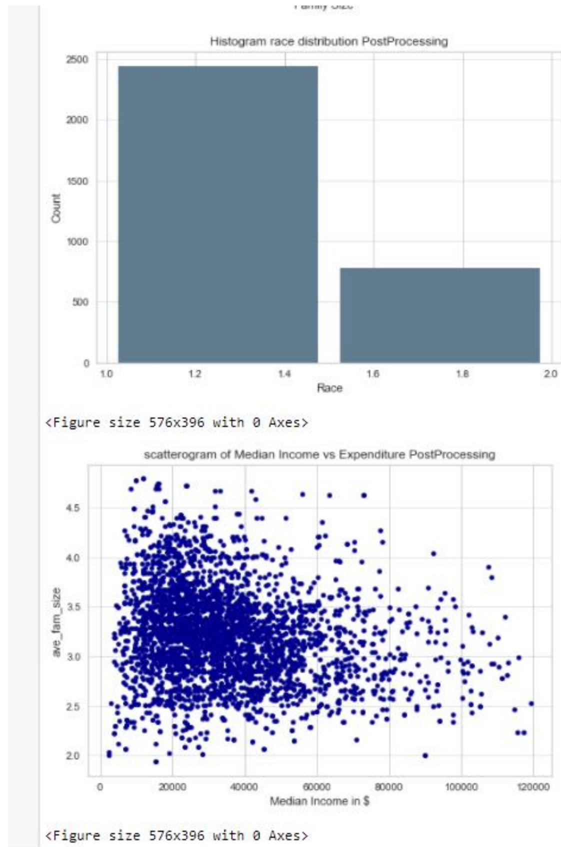


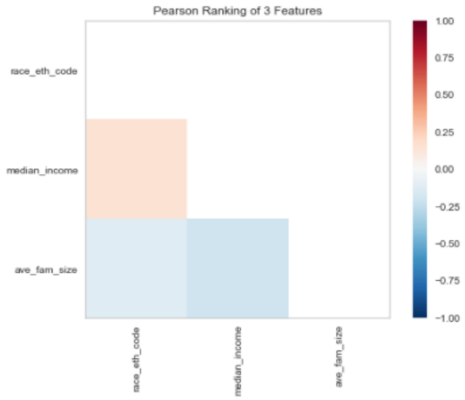
Figure 23.0

separate dependent and independent variables

In [14]:
```
X = dataSetUp.drop(DependentVariable, axis=1)
y = dataSetUp[DependentVariable]
```

Pearsons Analysis

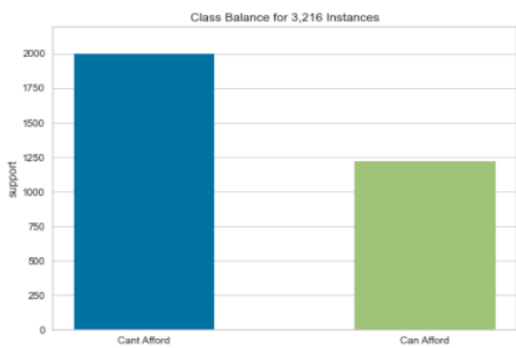
In [15]:
```
visualizer = Rank2D(algorithm='pearson')
visualizer.fit(X, y)
visualizer.transform(X)
visualizer.show()
```



Out[15]: <AxesSubplot:title={'center':'Pearson Ranking of 3 Features'}>

ClassBalance

In [16]:
```
visualizer = ClassBalance(labels=["Cant Afford", "Can Afford"])
visualizer.fit(y)        # Fit the data to the visualizer
visualizer.show()        # Finalize and render the figure
```



Out[16]: <AxesSubplot:title={'center':'Class Balance for 3,216 Instances'}, ylabel='support'>

Split Dataset into train and test dataset

In [17]:
```
trainDataset()
```

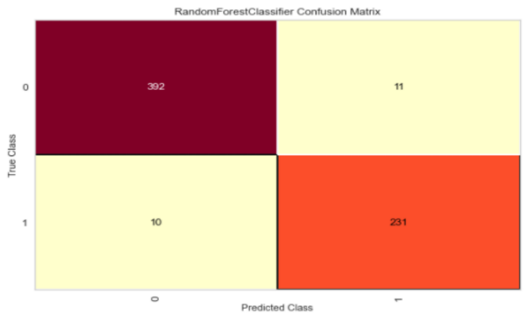
Create a dict of models to use

In [18]:
```
dict_classifiers = {
    "rfc": RandomForestClassifier(n_estimators=200),
    "clf": svm.SVC(),
    "mlpc": MLPClassifier(hidden_layer_sizes=(11, 11, 11), max_iter=500,  random_state=1),
    "lr": LogisticRegression(),
}
```

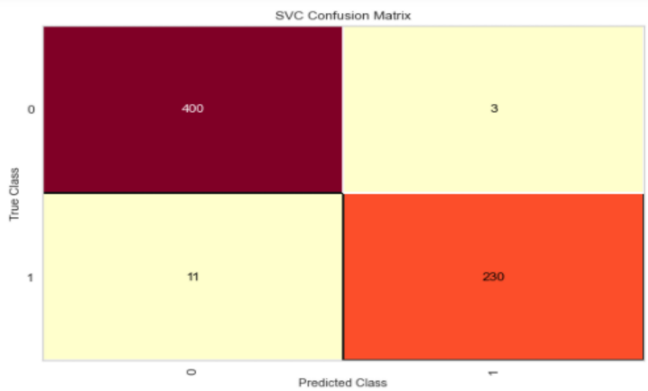
Figure 24.0

Train and Print Details

In [19]:
```
for model, model_instantiation in dict_classifiers.items():
    model = model_instantiation
    model.fit(X_train, y_train)
    y_score = model.predict(X_test)
    # yellow brick
    cm = ConfusionMatrix(model, classes=[0,1])
    cm.fit(X_train, y_train)
    cm.score(X_test, y_test)
    cm.show()
    confusion_Matrix = confusion_matrix(y_test, y_score)
    cm = accuracy_score(y_test, y_score)
    print(f"Printing Model details for : {model}\n"
          f"Printing Confusion Matrix\n{confusion_Matrix}\n"
          f"Printing Classification Report\n {classification_report(y_test, y_score)}\n"
          f"****\n"
          f"End of Model\n"
          f"****\n")
```

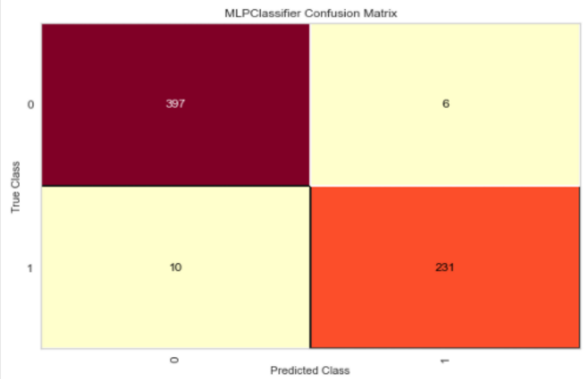



```
Printing Model details for : RandomForestClassifier(n_estimators=200)
Printing Confusion Matrix
[[392  11]
 [ 10 231]]
Printing Classification Report
               precision    recall  f1-score   support

           0       0.98      0.97      0.97       403
           1       0.95      0.96      0.96       241

    accuracy                           0.97       644
   macro avg       0.96      0.97      0.97       644
weighted avg       0.97      0.97      0.97       644

****
End of Model
****
```

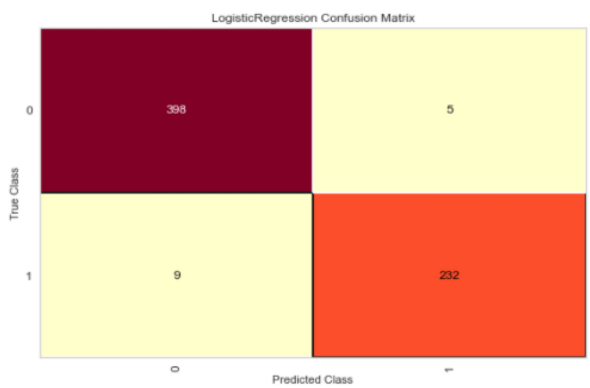



```
Printing Model details for : SVC()
Printing Confusion Matrix
[[400   3]
 [ 11 230]]
Printing Classification Report
               precision    recall  f1-score   support

           0       0.97      0.99      0.98       403
           1       0.99      0.95      0.97       241

    accuracy                           0.98       644
   macro avg       0.98      0.97      0.98       644
weighted avg       0.98      0.98      0.98       644

****
End of Model
****
```

Figure 25.0



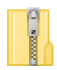

```
Printing Model details for : MLPClassifier(hidden_layer_sizes=(11, 11, 11), max_iter=500, random_state=1)
Printing Confusion Matrix
[[397   6]
 [ 10 231]]
Printing Classification Report
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       403
           1       0.97      0.96      0.97       241

    accuracy                           0.98       644
   macro avg       0.98      0.97      0.97       644
weighted avg       0.98      0.98      0.98       644

****
End of Model
****
```




```
Printing Model details for : LogisticRegression()
Printing Confusion Matrix
[[398   5]
 [  9 232]]
Printing Classification Report
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       403
           1       0.98      0.96      0.97       241

    accuracy                           0.98       644
   macro avg       0.98      0.98      0.98       644
weighted avg       0.98      0.98      0.98       644

****
End of Model
****
```

Figure 26.0

Raw Software Files



Capstone-master.zip

Model Evaluation

Utilizing the appropriate model selection, model evaluation and algorithm selection is a vital decision during the model building phase according to Raschka, S. (2018). The basis for evaluating a model is to select the optimal solution from various classification models generated in an iterated and complex model building process (Novaković, J. D. et al, 2017).

The four models that were selected for the purpose of the academic research were as follows:

1. Random forest classifier.
2. SVM (Support Vector Machine)
3. Logistic regression
4. MLPC (multilayer perceptron classifier)

The details with regards to the evaluation techniques and the matrices chosen to have been presented in the section below.

The model can be evaluated using available metrics utilizing opensource machine learning libraries like Scikit-learn, Keras and Tensorflow.

Appropriate libraries and methods were imported and utilized for the analysis to present a fully fledge solution that focusses on the validity of the model. Common matrices alike precision recalled accuracy errors were compiled and presented as numeric values. Other relevant matrices like ROC chart and precision – recalled diagrams were presented utilizing plotting libraries as graphs subsequent sections has a detail list of all the matrices that were used to validate the modeling approach.

The reason for choosing the matrices above are primary because of the results we found during our literature review where we found multiple similar models on peer review articles utilizing one or more of these matrices to validate their respective models.

Evaluation Process Justification

The holdout method is the most basic type of cross validation. The data set is divided into two parts: the training set and the testing set. As before, the errors it generates are added up to provide the mean absolute test set error, which is used to assess the model.

The purpose of this analysis the data was split into two parts namely training data, comprising of 80% of data, preselected using a set seed and 20 % used for the testing. The hold out method is configurable and can be changed on demand.

- Confusion Matrix: A confusion matrix is a table that shows how well a classifier performs on a set of test data for which the true values are known. The confusion matrix itself is straightforward, but the associated nomenclature might be perplexing.

- Classification Report: A classification report is used to assess the accuracy of a classification algorithm's predictions. How many of your guesses are correct and how many are incorrect?
- Accuracy Rate: The percentage of correct predictions for the test data is known as accuracy. It is easily calculated by dividing the number of correct predictions by the total number of predictions.
- Error Rate: The error of the method is defined as the inaccuracy of predicted output values. The error is expressed as an error rate if the goal values are categorical. This is the percentage of times the prediction is incorrect.
- Root Mean Square Error: The square root of the mean of the squared differences between actual and predicted outcomes is used to calculate RMSE. Squaring each error makes the values positive, and the square root of the mean squared error returns the error metric to its original units for comparison.
- Specificity: Specificity is the proportion of truly negative cases classified as negative; thus, it measures how well your classifier identifies negative cases. It is also referred to as the true negative rate.
- Sensitivity: Sensitivity is defined as the proportion of truly positive cases that were classified as positive; it is thus a measure of how well your classifier identifies positive cases. It is also referred to as the true positive rate.
- Balance Accuracy: To deal with imbalanced datasets, the balanced accuracy in binary and multiclass classification problems is used. It is defined as the average of recall obtained across all classes. When adjusted=False, the best value is 1 and the worst value is 0.

- Precision: Precision is the fraction of relevant instances among the retrieved instances in pattern recognition, information retrieval, and classification, whereas recall is the fraction of relevant instances that were retrieved.
- Recall: Recall literally refers to how many true positives were recalled. i.e. how many correct hits were also discovered. Precision is the percentage of returned hits that were true positives, i.e. correct hits.
- F1 Score: That is, a good F1 score indicates that you have low false positives and false negatives, indicating that you are correctly identifying real threats and are not bothered by false alarms. When an F1 score is 1, the model is considered perfect, while when it is 0, the model is considered a complete failure.
- Lift and Gain Chart: Lift is a measure of a predictive model's effectiveness calculated as the ratio of results obtained with and without the predictive model. Cumulative gains and lift charts are useful visual tools for assessing model performance. Both graphs have a lift curve and a baseline.

The reason for selecting these metrices was to ensure that we get a concordant result of misclassification of data as well as reliable numeric indicators of the performance of the individual models. This allows us to compare between the models and evaluate the performance of each individual model. Once we had the results, we use it to select the appropriate model for our predictive analysis.

The best model based on minimal misclassification and error was used in an iterative manner to predict the results.

Validation Results

Model Evaluation RandomForestClassifier(n_estimators=200)

Classification Report

	precision	recall	f1-score	support
0	0.98	0.98	0.98	403
1	0.96	0.96	0.96	241
accuracy			0.97	644

macro avg 0.97 0.97 0.97 644 weighted avg 0.97 0.97 0.97 644Accuracy Rate = 0.9720496894409938

Error Rate = 0.02795031055900621

Root Mean Square Error = 0.027950310559006212

Specificity = 0.9776674937965261

Sensitivity = 0.9626556016597511

Balance Accuracy = 0.9701615477281386

Precision = 0.9626556016597511

Recall = 0.9626556016597511

F1 Score = 0.9626556016597511

Model Evaluation SVC()

Classification Report

	precision	recall	f1-score	support
0	0.97	0.99	0.98	403
1	0.99	0.95	0.97	241
accuracy			0.98	644

macro avg 0.98 0.97 0.98 644 weighted avg 0.98 0.98 0.98 644

Accuracy Rate = 0.9782608695652174

Error Rate = 0.021739130434782594

Root Mean Square Error = 0.021739130434782608

Specificity = 0.9925558312655087

Sensitivity = 0.9543568464730291

Balance Accuracy = 0.973456338869269

Precision = 0.9871244635193133

Recall = 0.9543568464730291

F1 Score = 0.970464135021097

Model Evaluation MLPClassifier(hidden_layer_sizes=(11, 11, 11), max_iter=500, random_state=1)

Classification Report

	precision	recall	f1-score	support
0	0.98	0.99	0.98	403
1	0.97	0.96	0.97	241
accuracy			0.98	644

macro avg 0.98 0.97 0.97 644 weighted avg 0.98 0.98 0.98 644

Accuracy Rate = 0.9751552795031055

Error Rate = 0.024844720496894457

Root Mean Square Error = 0.024844720496894408

Specificity = 0.9851116625310173

Sensitivity = 0.9585062240663901

Balance Accuracy = 0.9718089432987037

Precision = 0.9746835443037974

Recall = 0.9585062240663901

F1 Score = 0.9665271966527197

Model Evaluation LogisticRegression()

Classification Report

	precision	recall	f1-score	support
0	0.98	0.99	0.98	403
1	0.98	0.96	0.97	241
accuracy			0.98	644

macro avg 0.98 0.98 0.98 644 weighted avg 0.98 0.98 0.98 644

Accuracy Rate = 0.9782608695652174

Error Rate = 0.021739130434782594

Root Mean Square Error = 0.021739130434782608

Specificity = 0.9875930521091811

Sensitivity = 0.9626556016597511

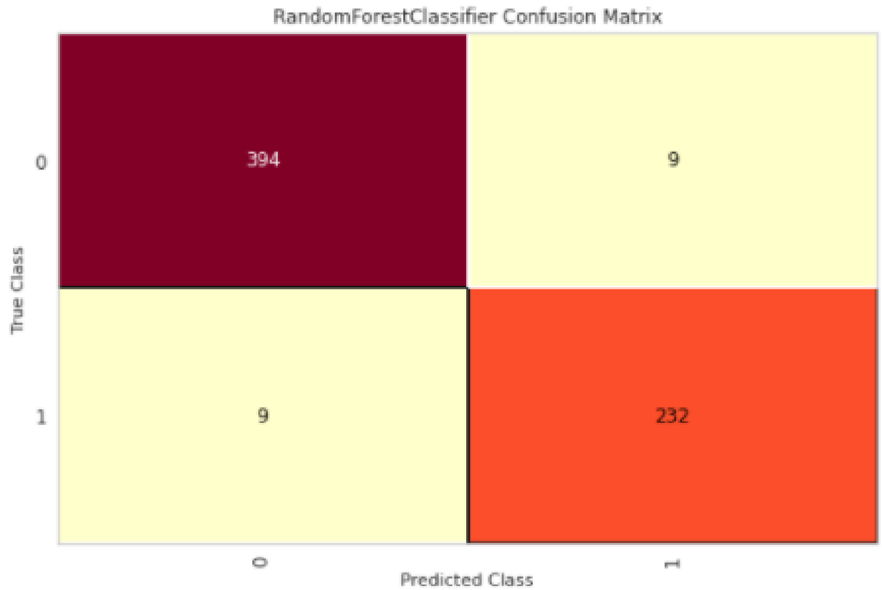
Balance Accuracy = 0.975124326884466

Precision = 0.9789029535864979

Recall = 0.9626556016597511

F1 Score = 0.9707112970711297

Model Evaluation RandomForestClassifier(n_estimators=200)



Classification Report

	precision	recall	f1-score	support
0	0.98	0.98	0.98	403
1	0.96	0.96	0.96	241
accuracy			0.97	644

macro avg 0.97 0.97 0.97 644 weighted avg 0.97 0.97 0.97 644

Accuracy Rate = 0.9720496894409938

Error Rate = 0.02795031055900621

Root Mean Square Error = 0.027950310559006212

Specificity = 0.9776674937965261

Sensitivity = 0.9626556016597511

Balance Accuracy = 0.9701615477281386

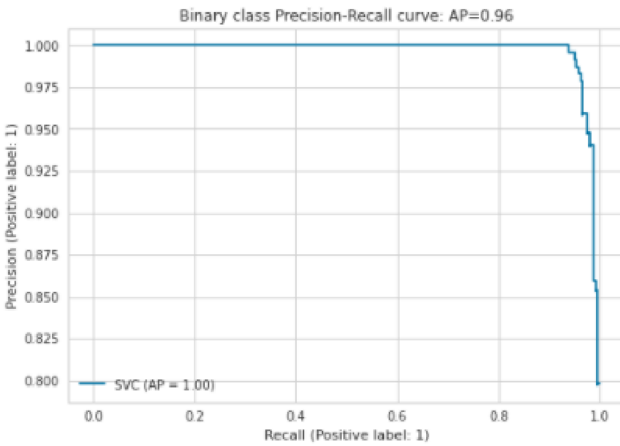
Precision = 0.9626556016597511

Recall = 0.9626556016597511

F1 Score = 0.9626556016597511

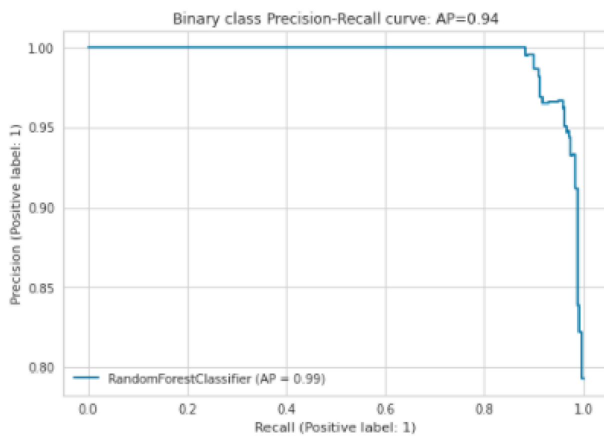
Figure 27.0

```
Average precision-recall score: 0.96
```




```
ROC unavailable for SVC()
```

```
Average precision-recall score: 0.94
```




```
findfont: Font family ['sans-serif'] not found. Falling back to DejaVu Sans.
findfont: Generic family 'sans-serif' not found because none of the following families were found: Arial, Liberation Sans, Bits
tream Vera Sans, sans-serif
```

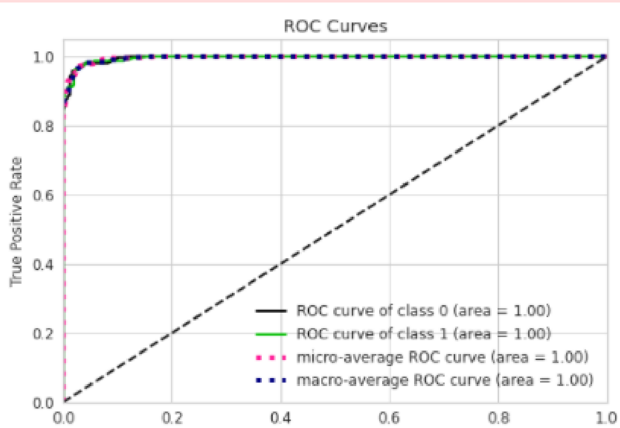
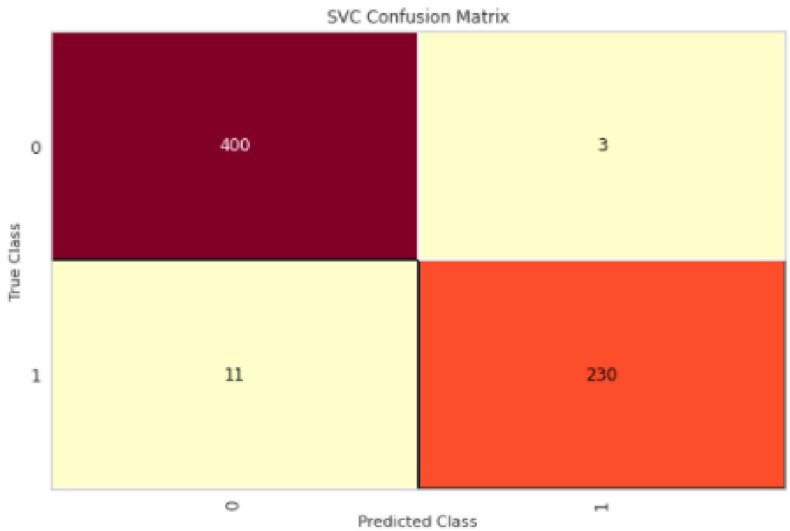


Figure 28.0

Model Evaluation SVC()



Classification Report

	precision	recall	f1-score	support
0	0.97	0.99	0.98	403
1	0.99	0.95	0.97	241
accuracy			0.98	644

macro avg 0.98 0.97 0.98 644 weighted avg 0.98 0.98 0.98 644

Accuracy Rate = 0.9782608695652174

Error Rate = 0.021739130434782594

Root Mean Square Error = 0.021739130434782608

Specificity = 0.9925558312655087

Sensitivity = 0.9543568464730291

Balance Accuracy = 0.973456338869269

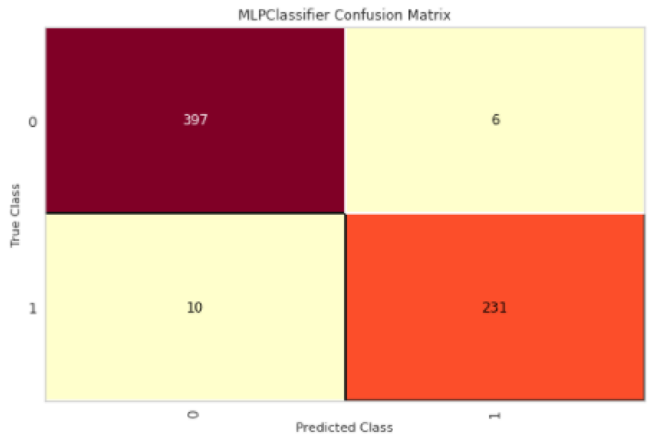
Precision = 0.9871244635193133

Recall = 0.9543568464730291

F1 Score = 0.970464135021097

Figure 29.0

Model Evaluation MLPClassifier(hidden_layer_sizes=(11, 11, 11), max_iter=500, random_state=1)



Classification Report

	precision	recall	f1-score	support
0	0.98	0.99	0.98	403
1	0.97	0.96	0.97	241
accuracy			0.98	644

macro avg 0.98 0.97 0.97 644 weighted avg 0.98 0.98 0.98 644

Accuracy Rate = 0.9751552795031055

Error Rate = 0.024844720496894457

Root Mean Square Error = 0.024844720496894408

Specificity = 0.9851116625310173

Sensitivity = 0.9585062240663901

Balance Accuracy = 0.9718089432987037

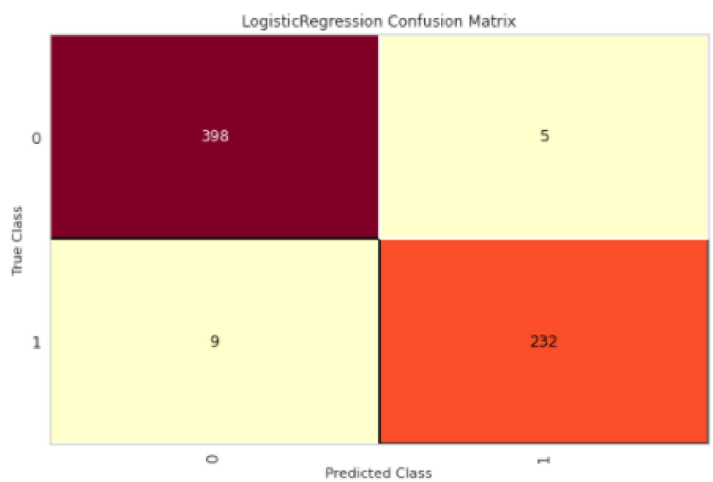
Precision = 0.9746835443037974

Recall = 0.9585062240663901

F1 Score = 0.9665271966527197

Figure 30.0

Model Evaluation LogisticRegression()



Classification Report

	precision	recall	f1-score	support
0	0.98	0.99	0.98	403
1	0.98	0.96	0.97	241
accuracy			0.98	644

macro avg 0.98 0.98 0.98 644 weighted avg 0.98 0.98 0.98 644

Accuracy Rate = 0.9782608695652174

Error Rate = 0.021739130434782594

Root Mean Square Error = 0.021739130434782608

Specificity = 0.9875930521091811

Sensitivity = 0.9626556016597511

Balance Accuracy = 0.975124326884466

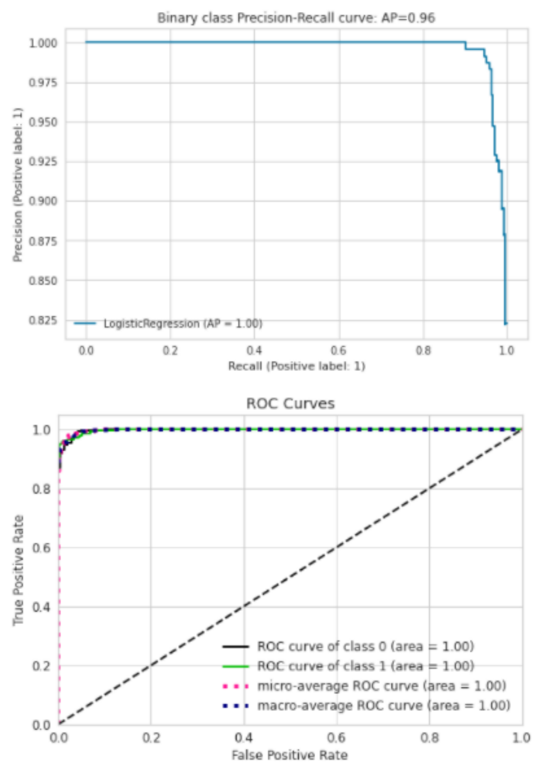
Precision = 0.9789029535864979

Recall = 0.9626556016597511

F1 Score = 0.9707112970711297

Figure 31.0

Average precision-recall score: 0.96



Average precision-recall score: 0.95

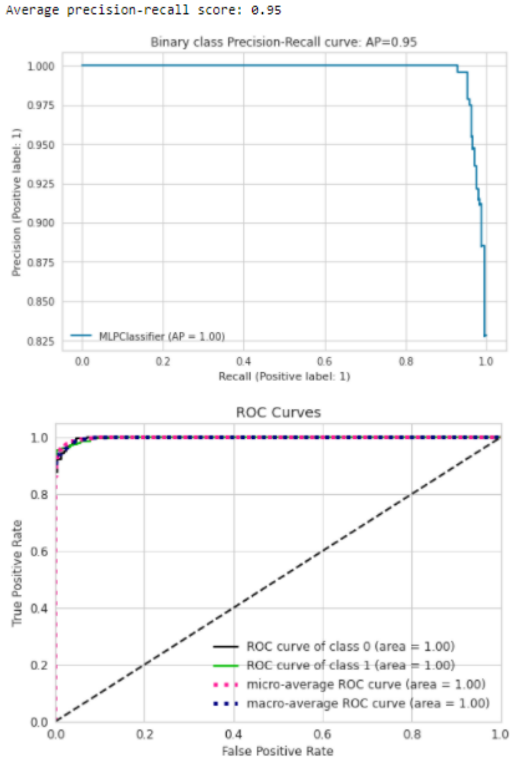


Figure 32.0

Conclusion

From the scores obtained above logistic regression md MLPC showed the highest accuracy and the least amount of misclassification. We decided to utilize MLPC for the predictive analysis section of our research. Overall, the accuracy rates of the models were generally greater than 95% of all models. This was greater than what we had expected and could be since the initial data cleanup had removed a lot of the outliers. The other factor which could have influenced the high accuracy rate could be because of the lower number of dependent variables which contribute to the forming of the independent variable. Other factors could have influenced the accuracy rates include a low multicollinearity score and lower degree of overfitting.

External Model Verification and Calibration

The dataset we utilized for this project does not have a historical record of raw data. Hence external validation using this approach is not viable. Resulting from this limitation the primary focus of this efforts will be on cross validation.

Literature Review

To cross validation, literature was reviewed focused on Artificial Intelligence Machine Learning models (AI/ML) specifically addressing food security. Much emphasis was laid to focus on any model which captured income, family size or ethnicity as a dependent variable to ensure the input data matched with our data.

Towards this end, we found three specific research papers outlined below sharing either similar data inputs, model selection or evaluation methodologies to our analysis. According to the research done by Gao, C. (2020), focus on identifying vulnerable households using machine learning food sustainability is a measure of per capita income and effective household size along with other markers specific to their research. Per capita income and effective household size were also found to be relevant variables in our own research findings. The model, random forest, that was used for the analysis is also similar to the one utilized in our research. the evaluation steps included ROC chart and the AUC charts along with the F1 score.

In the research paper Razzaq, A. (2021), utilizes the following models for their analyses SVM, KNN, random forest, neural network, naïve bayes and logistic regression. Four of these six models are also utilized in our analysis. The model evaluation techniques used in the research paper includes accuracy score, precision, recall and F1 score. These along with other evaluation techniques were also performed in our research.

Sthamer, C., (2020) utilizes income and tax, along with affordability of hobbies to measure food affordability in the United Kingdom. These data points share similarity with the data selection approach followed during our research. This paper also used random forest for their model and for evaluation precision, recall and accuracy scores methods were deployed.

Calibration

According to the literatures reviewed, all the validation methods deployed in our analysis are sufficient and comprehensive. Most peer reviewed articles provided one or more of the evaluation methodologies we adopted. Overall, our research evaluates nine matrices, accuracy rate, error rate, ROOT mean square error, specificity, sensitivity, balance accuracy, precision, recall, F1 score along with other markers like precision recall curve, lift and gain chart, class balance and predictor importance.

For instance, Gao C. (2020) utilized ROC, F1 scores and AUC charts. While Razzaq, A (2021) used accuracy score, precision, recall and F1 scores. Sthamer C., (2020) utilized precision, recall and accuracy score respectively for their models. All of these along with additional matrices have been included in our evaluation.

Future Recommendations

The next steps towards this model will be to utilize the best model algorithm to iteratively predict the median income required based on family size and ethnicity. Further development efforts if given enough time could include creating a synthetic test dataset to stress test the model. We would also like to explore the validity of the model once a newer dataset has been published by the California Department of Health. Other items which can add value to the research includes addition of dependent values like consumer habits, WIC and SNAP benefits, proximity to grocery stores, family wealth and inheritances, etc.

The model does not need to undergo any revisions because the accuracy score and error rates are within the acceptance tolerance values. Moreover, predictive evaluation outcomes are consistent with our preliminary analysis obtained using tableau.

Future recommendations would include the addition of more dependent variables as discussed in our response in question 3. If we were to redo this project, we would have selected a dataset that contains more variables and has historical data that would enable for eternal validation.

Model Deployment and Model Life Cycle

The various phases of a software deployment lifecycle for our project are as shown below with high level details of the timeline followed. The methodology followed will be close to an agile model with development performed iteratively. Figure 33.0 shows the various phases of the model development with a high-level timeline.

Figure 33.0 outlines the screenshot of the project schedule obtained from the project management website Monday.com. The various aspects including task list, timeline owner and the status of the task has been represented in the screenshot.

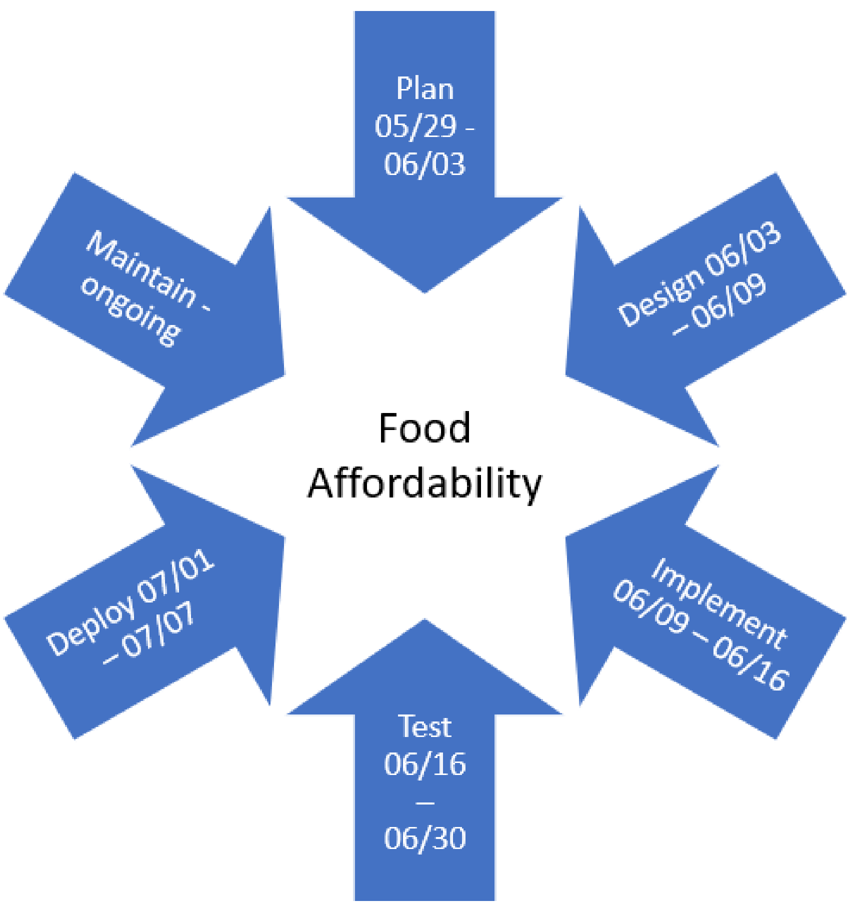


Figure 33.0

Deployment Cost

Model Deployment Costs

Deploying models on to a platform can be classified as an effort on its own. Multiple teams and resources are generally required to successfully deploy a model or any application or appliance on to an infrastructure successfully. On a high level the following steps need to be completed to stand up an application.

1. Define an architectural diagram
2. Seek approval for proposed diagram
3. Identify and size the specifications for all components over the OSI layers
4. Specify protocols used at any handoff points
5. Establish Third party authorizations, audit requirements and all necessary paperwork
6. Get licensing details if proprietary components are used
7. Procure infrastructure components
8. Define necessary service accounts, user groups, ldap / sso configurations
9. Integrate with IDP provider if external facing
10. Get necessary certificates and pem files
11. Procure configuration files for connectivity and data handling
12. Request firewall changes
13. Deploy and establish functionality on a nonproduction model
14. Create user provisioning and servicing requests
15. Define patching and maintenance, backup schedule
16. Define pipeline model for deployment

17. Deploy Production instance
18. Communicate to target audience
19. Establish training model
20. Define and follow upgrade schedule

The proposed architectural diagram for the purpose of this model is illustrated in figure 34.0 below. The various components have been defined and multiple layers that the system interacts with has been showcased in the diagram Figure 34.0.

The model can be deployed either on-prem or as a serverless instance on a cloud native environment such as public docker clouds or a lambda-dynamo db instance on an AWS VPC. The high-level aspects of the infrastructure cost break down for each of these are listed below

On Prem

Assuming a high availability (HA) model with 2 application instance and a data base the individual application instance and the database instance can be built out as indicated in Figure 34.0. A disaster recovery server has not been provisioned in this instance as the application is not classified as business critical.

To establish the cost of deploying the application we will be examining the generic quote details shared by Vmware Vra servers and Microsoft Azure servers for our calculations. The details with regards to the pricing can be found on the respective websites attached below:

Vmware : How do I estimate the price of a deployment (vmware.com)

Azure : Pricing - Machine Learning | Microsoft Azure

Vmware pricing computations are as outlined below. The assumption is based on the architectural diagram presented in Figure 34.0

Table 3.0 outlines the cost per application hardware with no additional vendor incentives attached. Typically obtaining licenses to scale will reduce the upfront cost as vendors would add some additional discounts.

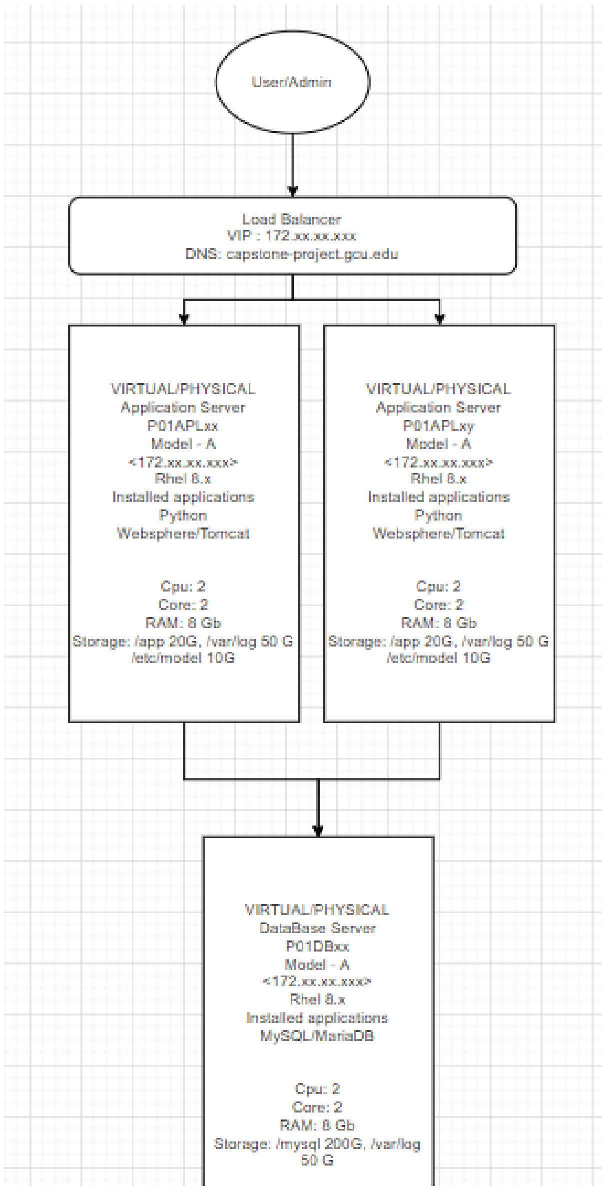


Figure 34.0

Instances	Application			Databases	
Item	**Cost/Day**	**Yearly**	**Instances (2)**	**Cost/Day**	**Yearly**
Compute	$0.40	$146.00	$292.00	$0.20	$73.00
Storage	$0.03	$10.95	$21.90	$0.03	$10.95
Additional Charge	$0.10	$36.50	$73.00	$0.10	$36.50
Total Price of service	$0.53	$193.45	$386.90	$0.33	$120.45
Grand Total	**$507.35**				

Table 3.0

Table 3.0

If assuming an Azure Linux VM the pricing quoted is as outlined below from the vendor website. As mentioned earlier scaled discounts have not been accounted for in the initial quotes. Typically, the vendor management teams, and the sourcing teams work with the external vendor to finalize a pricing that is generally lower than the price published on the vendor website.

Instance	vCPU(s)	RAM	Linux VM Price	Pay as You Go	1 year reserved	3 year reserved
				Total Price	total price	total price
F2s v2	2	4 GiB	$61.758/month	$61.758/month	$36.50/month ~41% savings	$22.638/month ~63% savings

Table 4.0

Cloud native

As per Zhucheng. TU., 2018, The cost per invocation using a serverless lambda tied to a dynomodb instance was reported to be $0.000000208. The maintenance cost was stated as zero in this instance. If we were to deploy based on a similar cloud native strategy utilizing stateless code, we could expect comparable costs.

Alternative models have been explored with minimal cost of hardware to run AI mL models utilizing raspberry pi as a host. The details and the cost break down have been presented by Truong, S. N. (2020).

The cost of load balancer VIP has been assumed to be $100 per year for initial procurement and maintenance. The overall cost of maintaining an environment of this scale is assumed to be $500 as it does not require additional overheads and minimal maintenance. The rationale behind this assumption is that the enterprise already has a partnership with the vendor and the deployment will be on an environment that can be spun up on demand versus following a complete procurement lifecycle.

Schedule, Training, and Risk

Figure 35.0 outlines the screenshot of the project schedule obtained from the project management website Monday.com. The various aspects including task list, timeline owner and the status of the task has been represented in the screenshot.

Capstone Project GCU

Powered by monday.com
Click here to start your free trial

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Cost	Item ID (auto generated)
Subitems	Name	Owner	Status	Date	Item ID (auto generated)			
	Subitem				1479582580			
Create Use Cases		ıta Availability, Sprint to Analyze Model suitability, Sprint to check for similar research articles, Final						1479584020
Subitems	Name	Owner	Status	Date	Item ID (auto generated)			
	Sprint to Analyze Data Availat		Done	2021-05-29	1479584171			
	Sprint to Analyze Model suitat		Done	2021-05-30	1479585809			
	Sprint to check for similar res		Done	2021-05-31	1479586066			
	Finalize Research article		Done	2021-06-02	1479586271			
							0	
Planning								
Name	**Owner**	**Subitems**	**Status**	**Priority**	**Timeline - Start**	**Timeline - End**	**Cost**	**Item ID (auto generated)**
Create User Stories	sushil sivaram	This is a subitem	Done	High	2021-06-01	2021-06-04	100	1479582560
Subitems	Name	Owner	Status	Date	Item ID (auto generated)			
	This is a subitem				1479582582			
Preliminary analysis of data	Megha Gubbala		Done	Medium	2021-06-02	2021-06-05	100	1479582568
Identified outliers	Sylvia Nanyangwe		Done	Medium	2021-06-04	2021-06-07		1479588704
Explored suitable libraries	sushil sivaram		Done	Low	2021-06-05	2021-06-08		1479588888
Explored Evaluation and validation methodologies	Megha Gubbala		Done	Medium	2021-06-06	2021-06-09		1479589145
Review of literature	Sylvia Nanyangwe		Done	Medium	2021-06-08	2021-06-11		1479589382
Explored Deployment technology	sushil sivaram		Done	Medium	2021-06-07	2021-06-11		1479591565
					2021-06-01	**2021-06-11**	**200**	
Execution								
Name	**Owner**	**Subitems**	**Status**	**Priority**	**Timeline - Start**	**Timeline - End**	**Cost**	**Item ID (auto generated)**
Coded Reusable functions	sushil sivaram		Done	High	2021-06-09	2021-06-10	0	1479582558
Coded Visuals and graphs	Megha Gubbala		Done	High	2021-06-09	2021-06-10	0	1479582566
Coded Evaluation techniques and results	Sylvia Nanyangwe		Done	Medium	2021-06-10	2021-06-11		1479592394
Coded Predective models	sushil sivaram		Done	High	2021-06-10	2021-06-11		1479592591
Refactored and moved to Jupyter notebook	Megha Gubbala		Done	Medium	2021-06-10	2021-06-12		1479592929
Committed and refined code revision	Sylvia Nanyangwe		Done	Medium	2021-06-12	2021-06-13		1479593190
Added Deployment superstructure	Megha Gubbala		Done	High	2021-06-12	2021-06-13		1479593736
Test	Sylvia Nanyangwe		Done	Medium	2021-06-12	2021-06-13		1479594346
					2021-06-09	**2021-06-13**	**0**	
Testing and Validation								
Name	**Owner**	**Subitems**	**Status**	**Priority**	**Timeline - Start**	**Timeline - End**	**Cost**	**Item ID (auto generated)**
Tested Model	Megha Gubbala		Done	Medium	2021-06-17	2021-06-18		1479594812
Evaluated Model for accuracy and error percentage	Sylvia Nanyangwe		Done	Medium	2021-06-18	2021-06-19		1479595189
Gathered relevant metrics and compared to peer reviewed arti	sushil sivaram		Done	Low	2021-06-18	2021-06-19		1479596509
					2021-06-17	**2021-06-19**	**0**	
Launch								
Name	**Owner**	**Subitems**	**Status**	**Priority**	**Timeline - Start**	**Timeline - End**	**Cost**	**Item ID (auto generated)**
Deployed code from github to Binder	Megha Gubbala		Done	High	2021-06-07	2021-06-08	100	1479582572
Launched code and validated	Sylvia Nanyangwe		Done	Medium	2021-07-09	2021-07-13		1479596842
					2021-06-07	**2021-07-13**	**100**	

Figure 35.0

Training, and Risk

Training is an essential part of successful model implementation. An early stage of the model ideation involves finding the stakeholders that are a part of the model. Once implemented the value of the model is driven by the utility the model brings to the enterprise. Training the end user will enhance the value of the model. To formulate the training artifacts, the following steps need to be followed

1. Analyze and identify: Analyze and identify the training needs and the targeted audience. This step should also include peripheral factors like cost, availability of audience, resources required and other factors that will affect training needs.
2. Design: The training resources should be so designed that it has all critical information pertaining to the system along with troubleshooting guides and a list of acronyms that the audience may be unfamiliar with.
3. Development: The development stage of the training resource should include activities listing and explaining core and ancillary components in a language that is engaging and easily understandable by the target audience. Any infographics, charts graphs or additional data may be added to the appendix to ensure users have the necessary tools at their disposal in any situation.
4. Implementation: Scheduled demonstrations and workshops may be organized to ensure the right audience has the required training to make the adoption of the model successful.
5. Revision: any revision to the guidelines, processes or practices need to be included in the training material in a timely manner either as an addendum or with a newer version of the training document.

Risk Mitigation

This Model Will Track Over Time Periodic checks would have to be performed to ensure that the model accuracy thresholds are within standards. Any deviation from the standards set would have to result in triage and refactoring phase where the metrics are programed to be within the preset standards. Additional data when available should be utilized to reflect the current state of the system. If any additional variables need to be introduced as a part of the analysis then the development team should be informed by means of the current ticketing system so that the work may be completed towards the enhancement.

Explanation of How This Model Can Be Used on a Repetitive Basis

The proposed model can be re-run on demand. The seed value for the training and testing data may be reset periodically to ensure model accuracies are reflective of the whole data set. The predictive analysis section had 3 user configurable entities namely Median income, Family Size and ethnicity indicator that may be changed on the fly or enhanced as a parameter if running through a CICD platform for obtaining results based on user inputs.

Benefits

The specific benefits over time of using this model for the organization would include the ability to gather invaluable data pertaining to the business problem after running the model. The predictive analysis capability ensures that the model is reusable can provide baseline metrics of income required based on various family sizes and ethnicity for the counties in California. Additional data when published can be amended to the model further enhancing its capabilities.

Recommendations

Frequent checks to the data source should be made to ensure updated data is obtained. Regular maintenance schedule should include vulnerability analysis of upgrade of libraries, Kernal patching stress and performance testing, availability monitoring and other KPIs/KRIs which will ensure 100%operational stability of the system.

Recommendations for practice.

Response to the research question revealed that food affordability is an imminent problem which needs to be addressed and potentially resolved based on the primary dependent variables as factors for funding. Frequent checks to the data source should be made to ensure updated data is obtained.

Regular maintenance schedule should include vulnerability analysis of upgrade of libraries, Kernal patching stress and performance testing, availability monitoring and other KPIs/KRIs which will ensure 100%operational stability of the system.

Recommendations for future research. Based on the findings from this study and current literature on the topic, the first recommendation for future research is to identify other factors that could affects food affordability like, average food cost, nutritional values, proximity of food stores, availability of current benefits like Supplemental Nutrition Assistance Program (SNAP) and Woman, Infants and Children (WIC) benefits. Addition of these data points would enhance the results to provide a more precise prediction. Additional recommendation include updating the data set to a newer version to reflect a more up to date model for evaluation.

Conclusions

This quantitative study addressed the problem focused on food affordability for single mothers with multiple dependents on a fixed income across different ethnic groups. The problem tries to predict the optimal income necessary to ensure that the segment of the population has the necessary means to minimize the food insecurity crisis prevalent in California. The following are the results obtained from the analysis.

Food insecurity is faced by 775 members of the sampled records of 3216 indicating a quarter of the population are in a state of financial duress. The effects of lower median income is generally leads to an elevated proportion of white demographics being effected by food insecurity. Nonwhite demographics on an average earn less than white demographics but can still find means to avoid food insecurity.

References

Azuma, A. M., Gilliland, S., Vallianatos, M., & Gottlieb, R. (2010). Food access, availability, and affordability in 3 Los Angeles communities, Project CAFE, 2004-2006. Preventing chronic disease, 7(2), A27.

Brandenburg, L. (n.d.). 3 Ways to Find Cost-Effective Solutions to Business Problems. Retrieved from https://www.bridging-the-gap.com/cost-effective-solutions/

Congressional Budget Office. (2021, March). Estimated Budgetary Effects of H.R. 1319, American Rescue Plan Act of 2021 (H.R. 1319, American Rescue Plan Act of 2021). Nonpartisan Analysis for the U.S. Congress. https://www.cbo.gov/publication/57056

Dubowitz, T., Dastidar, M. G., Troxel, W. M., Beckman, R., Nugroho, A., Siddiqi, S., Cantor, J., Baird, M., Richardson, A. S., Hunter, G. P., Mendoza-Graf, A., & Collins, R. L. (2021). Food Insecurity in a Low-Income, Predominantly African American Cohort Following the COVID-19 Pandemic. American Journal of Public Health, 111(3), 494–497

Fandango, A. (2018). Mastering tensorflow 1. x : Advanced machine learning and deep learning concepts using tensorflow 1. x and keras. ProQuest Ebook Central http://ebookcentral.proquest.com.lopes.idm.oclc.org

Gao, C., Fei, C. J., McCarl, B. A., & Leatham, D. J. (2020). Identifying Vulnerable Households Using Machine Learning. Sustainability, 12(15), 6002.

Glasmeier, A. K. (2004). Living Wage calculator. Living Wage Calculator - Living Wage Calculation for California. https://livingwage.mit.edu/states/06.

Hidalgo, E (2019). Adapting the scrum framework for agile project management in science: Case study of a distributed research initiative. doi: 10.1016/j.heliyon.2019.e01447

Irani Z., Sharif A.M, Lee H., Aktas E., Topaloğlu Z., Wout T.V., & Huda S. (2018). Managing food security through food waste and loss: Small data to big data. Computers & Operations Research, Volume 98, 2018, Pages 367-383, ISSN 0305-0548. https://doi.org/10.1016/j.cor.2017.10.007.

Kleinbaum, D.G., & Klein, M. (2002). Logistic Regression: A self-learning text (vol. 2nd ed). Springer.

Kulkarni A., Chong D., Batarseh F.A. 2020. 5 - Foundations of data imbalance and solutions for a data democracy. Data Democracy, Academic Press, Pages 83-106. https://doi.org/10.1016/B978-0-12-818366-3.00005-8.

Mudrak, R., Lagodiienko, V., Lagodiienko, N., & Rybchak, V. (2020). Food Affordability and Economic Growth. TEM Journal, 9(4), 1571–1579. https://doi-org.lopes.idm.oclc.org/10.18421/TEM94-32

Noble, W. S. (2006). What is a support vector machine? Nature Biotechnology, 24(12). https://doi-org.lopes.idm.oclc.org/10.1038/nbt1206-1565

Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž., & Milica, T. (2017). Evaluation of classification models in machine learning. Theory and Applications of Mathematics & Computer Science, 7(1), 39-46.

Olaimat, A. (n.d) Food Safety During and After the Era of COVID-19 Pandemic Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7417330/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in {P}ython. Https://Scikit-Learn.Org/Stable/about.Html#citing-Scikit-Learn. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Rajaram, R. (2009). Female-Headed Households and Poverty: Evidence from the National Family Health Survey. Department of Economics, Terry College of Business, 1. https://www.atlantafed.org/-/media/Documents/news/conferences/2009/3rd-international-economics/093rdseinternationaleconomicspaperrajaram.pdf

Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. arXiv preprint arXiv:1811.12808.

Razzaq, A., Ahmed, U. I., Hashim, S., Hussain, A., Qadri, S., Ullah, S., ... & Asghar, A. (2021). An Automatic Determining Food Security Status: Machine Learning based Analysis of Household Survey Data. International Journal

Shahnasarian, M. (2021). Implications of the Coronavirus Pandemic for Vocational Expert Assessments: A Preliminary Analysis. Rehabilitation Professional, 28(3), 135–140. https://web.b.ebscohost.com/abstract?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=23286202&AN=148023696&h=TGxQLqscwM6i3kYnzYl9ZeTd%2fctXbMuKWvtlyYJE8GSJW8nAdPz%2bxAlHGzjgp2WF9l%2bv7KmvAz8xlyj04%2bF1zQ%3d%3d&crl=c&resultNs=AdminWebAuth&resultLocal=ErrCrlNotAuth&crlhashurl=login.aspx%3fdirect%3dtrue%26profile%3dehost%26scope%3dsite%26authtype%3dcrawler%26jrnl%3d23286202%26AN%3d148023696

Sprenger, J., & Weinberger, N. (n.d.). Simpson's paradox (Stanford encyclopedia of philosophy). Stanford Encyclopedia of Philosophy. https://plato.stanford.edu/entries/paradox-simpson/

Sthamer, C., (2020, June 08). Editing of LCF (living cost and food) survey income data with machine learning. ONS (Office for National Statistics). https://unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.58/2020/mtg1/SDE2020_T1-B_UK_Sthamer_Paper.pdf

Stone, H., Sidel, J. L., & Bloomquist, J. (2012). Quantitative descriptive analysis. Descriptive Sensory Analysis in Practice, 4, 53-69. https://doi.org/10.1002/9780470385036.ch1f

Tableau.com. (n.d.). Data cleaning: The benefits and steps to creating and using clean data. Https://Www.Tableau.Com/.https://www.tableau.com/learn/articles/what-is-data-cleaning

Truong, S. N. (2020). A Low-cost Artificial Neural Network Model for Raspberry Pi. Engineering, Technology & Applied Science Research, 10(2), 5466-5469.

Tu J.V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes (Vol. 49, pp. 1225-1231). Journal of Clinical Epidemiology. https://doi.org/10.1016/S0895-4356(96)00002-9.

Tu, Z., Li, M., & Lin, J. (2018, June). Pay-per-request deployment of neural network models using serverless architectures. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations (pp. 6-10).

Appendix A: Data Set

Raw Data

food_afford_cdp_co_r
egion_ca4-14-13-ada.

Data Dictionary

foodaffordabilitydd.xl
sx

Source URI

Food Affordability - Datasets - California Health and Human Services Open Data Portal

Cleaned data used for tableau.

cleaned.csv

Tableau file

Capstone.twb