

PREDICTING SOCIAL HEALTH INSURANCE FUND CONTRIBUTIONS FROM KENYA'S INFORMAL SECTOR WORKERS

INTRODUCTION TO THE PROJECT

This project focuses on the informal sector which is a major contributor to the Kenyan economy yet informal workers often face challenges accessing social security benefits and due to the undocumented and variable nature of their income and hence this presents a critical barrier to achieving universal health through the social health insurance fund(SHIF)

GROUP 11 MEMBERS:

1. CYNTHIA DALMAS
2. BRIAN OCHIENG
3. NICOLE BOSIBORI
4. RUTH NYAKIO
5. EDWIN MUTENDWA

DATE OF SUBMISSION: 09/08/2024

GITHUBLINK:

<https://github.com/itsyourgirlnicky/Prediciting-Contributions-of-Informal-Workers-to-the-SHIF>

Table of Contents

1. Introduction:	3
1.1 Stakeholders	3
2. Business Understanding	3
3. Data Understanding	4
3.1 Data elements	4
4. Problem Statement	4
4.1 Justification of the problem	4
5. OBJECTIVES	5
5.1 Main Objective	5
5.2 Specific Objective	6
6. Metrics of Success	6
7. Data Preparation and Cleaning	6
7.1 Keys step include:	7
8. Exploratory Data Analysis	7
8.1 Income distribution:	7
8.2 Occupation Distribution:	7
9. Statistical Analysis	7
10. Modeling	8
10.1 Models Used	8
11. Evaluation	8
11.1 Best Performing Model	9
12. Deployment	9

1. Introduction:

In 2023, the government of Kenya enacted the Social Health Insurance Act of 2023, marking a significant step towards Universal Health Coverage. This legislation ensures all citizens can access quality healthcare services without catastrophic health expenses.

An accurate income prediction model can significantly improve the SHIF program. It can streamline contribution collection by determining appropriate amounts for informal workers, ensuring fairness and accuracy. The model can also identify low-income households within the informal sector, enabling targeted social programs that support the most vulnerable. Furthermore, data insights from the model can inform policy decisions directly impacting informal workers and the overall SHIF program, making the system more responsive and efficient

1.1 Stakeholders

1. Informal Sector Workers: Beneficiaries who will receive fair and accurate contribution assessments and targeted social programs.
2. Government of Kenya: Implementers of the SHIF program who will benefit from streamlined contribution collection and informed policy decisions

2. Business Understanding

A critical challenge in implementing this Act lies in determining appropriate contributions from informal sector workers, who make up a substantial portion of the Kenyan workforce. Unlike formal employment with documented salaries, income in the informal sector is often variable and undocumented, complicating the process of contribution assessment. About

80% of Kenya's population is engaged in the informal sector, and it is difficult to determine the monthly income, and by extension the SHIF contributions, for this sector

3. Data Understanding

The project aims to use publicly available data from the Kenya National Bureau of Statistics (KNBS), specifically the Kenya - Kenya Demographic and Health Survey 2022. KNBS collects data on household indicators from various regions in Kenya. Some of the variables include

3.1 Data elements

- House Structure: Type and quality of housing.
- Sources of Water: Accessibility and types of water sources.
- Incomes: Documented and estimated income levels.
- Urban or Rural: Classification of the area as urban or rural.

4. Problem Statement

Determining appropriate contributions for informal sector workers is challenging due to the variable and undocumented nature of their income. This project aims to develop a machine learning model to predict the income of informal sector workers, ensuring accurate and fair contributions to the Social Health Insurance Fund (SHIF)

4.1 Justification of the problem

The informal sector is a significant contributor to the Kenyan economy, yet informal workers often face challenges accessing social security benefits due to the undocumented and variable nature of their income. This presents a

critical barrier to achieving Universal Health Coverage (UHC) through the Social Health Insurance Fund (SHIF). Here's why predicting income for informal workers is a crucial problem to address:

- **Fair and Equitable Contributions:** An accurate income prediction model ensures fair contribution assessments for informal workers to the SHIF. Without such a model, some workers might be under- or over-charged, hindering the program's financial sustainability and fairness.
- **Improved Program Efficiency:** Streamlining contribution collection through a reliable model reduces administrative burdens, allowing the SHIF to focus on core functions like healthcare provision.
- **Targeted Social Programs:** Identifying low-income households within the informal sector enables the development of targeted social programs that directly address their needs. This improves the overall impact of social safety nets.
- **Data-Driven Policy Decisions:** Insights from the model can inform policy changes that better support informal workers and optimize the SHIF program. This promotes a more responsive and effective social security system.

5. OBJECTIVES

5.1 Main Objective

1. Develop a model to predict the contribution of the informal sector workers to the Social Health Insurance Fund based on data from household demographics, location, income group and type of work.

5.2 Specific Objective

1. Conduct Exploratory Data Analysis (EDA): Analyze the dataset to understand the distribution and relationships of various features and identify patterns associated with Income per month.
2. Create an easy-to-use Chatbot that allows informal sector workers to determine their required contributions to the SHIF based on the prediction model.
3. Utilize insights from the model to inform and support policy decisions, ensuring that SHIF contributions are fair and equitable for all informal sector workers.

6. Metrics of Success

Accuracy & F1 Score metrics will be used to evaluate the balance between precision and recall in classifying low-income households within the informal sector, ensuring targeted social programs are accurately directed.

1. ROC curves will be used to identify the best classification model
2. User Satisfaction Score: This metric will gauge the ease of use of the Chatbot to determine their SHIF contribution
3. Policy Impact Assessment: This metric will evaluate how effectively our model's insights support policy decisions

7. Data Preparation and Cleaning

Our data which is in form of csv involves handling missing values,normalizing features,and encoding categorical variables to ensure the dataset is suitable for training purposes

7.1 Keys step include:

Dropping irrelevant columns

Imputing missing values

Encoding categorical variables

8. Exploratory Data Analysis

8.1 Income distribution:

The income groups are divided into ranges: 0-10k, 10k-20k, 20k-30k, 30k-40k, and 40k-50k. The chart reveals that the 0-10k and 10k-20k income groups have the highest frequencies, indicating a large portion of the population falls within these lower income brackets. This distribution suggests a concentration of individuals in the lower income ranges.

8.2 Occupation Distribution:

'Agriculture - employee' and 'skilled manual' categories have the highest frequencies, indicating a significant portion of the population is employed in agricultural work and skilled manual labor

9. Statistical Analysis

Statistical methods, such as the Kruskal-Wallis test are employed to analyze the relationships between different variables and their impact on income predictions. Results indicate differences across various demographic and socio-economic factors, highlighting the multifaceted nature of income determinants

10. Modeling

10.1 Models Used

- 1.Naive Bayes
- 2.Logistic Regression
- 3.Decision Tree
- 4.Support Vector Machine
- 5.Random Forest
- 6.Gradient Boosting
- 7.K-Nearest Neighbors
- 8.Deep Learning

11. Evaluation

From the ROC curve, it is evident that all classifiers except Naive Bayes achieve perfect performance with an Area Under the Curve (AUC) of 1.00. This indicates flawless discrimination between classes for KNN, Decision Tree, Random Forest, Gradient Boosting, SVM, and Logistic Regression models. The Naive Bayes classifier, while still performing well, has a slightly lower AUC of 0.96, indicating some degree of misclassification compared to the other models. The near-perfect and identical ROC curves for KNN, Decision Tree, Random Forest, Gradient Boosting, SVM, and Logistic Regression suggest these models are exceptionally effective for this classification task, reflecting their high accuracy and reliability. The Naive Bayes classifier, although slightly

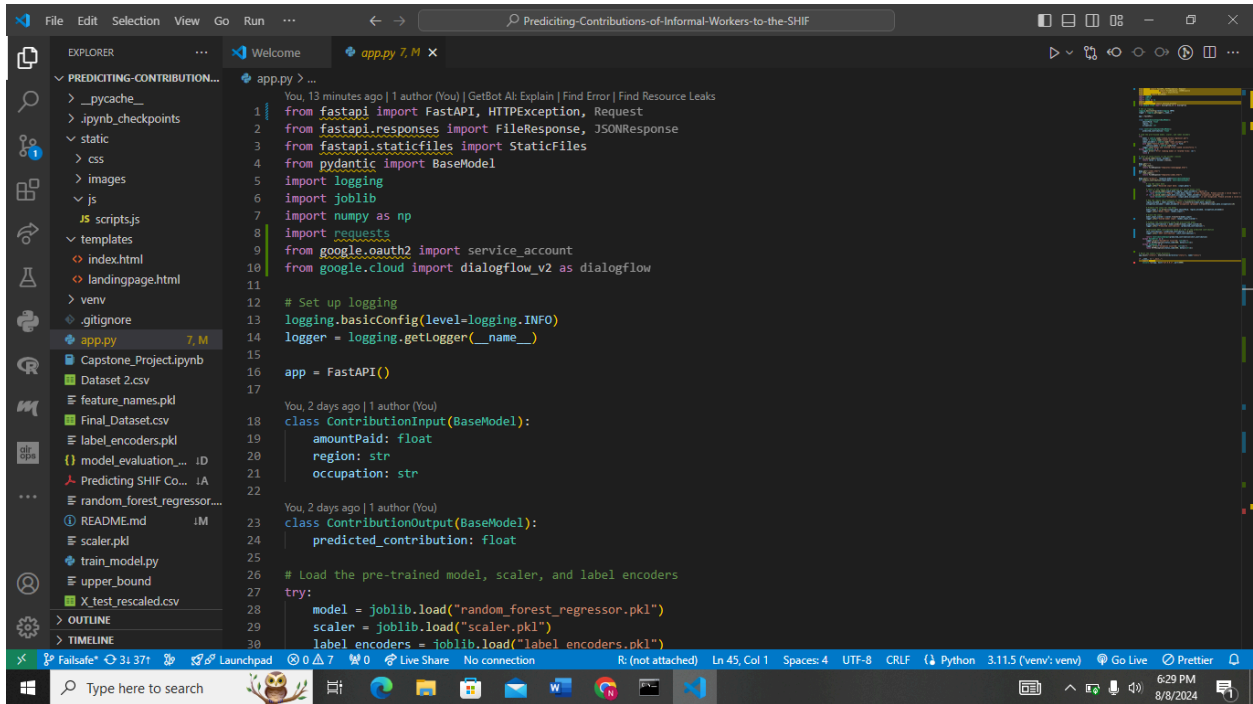
less accurate, still demonstrates strong performance, making it a viable option depending on the specific context and requirements of the task.

11.1 Best Performing Model

Models are evaluated using metrics such as accuracy, precision, recall and F1 score. ROC curves and confusion matrices provide a visual representation of model performance and from the confusion matrix and classification report the k-nearest neighbors model displayed an outstanding performance. The model achieved perfect classification across all classes resulting in an overall accuracy of 1.00(100%).

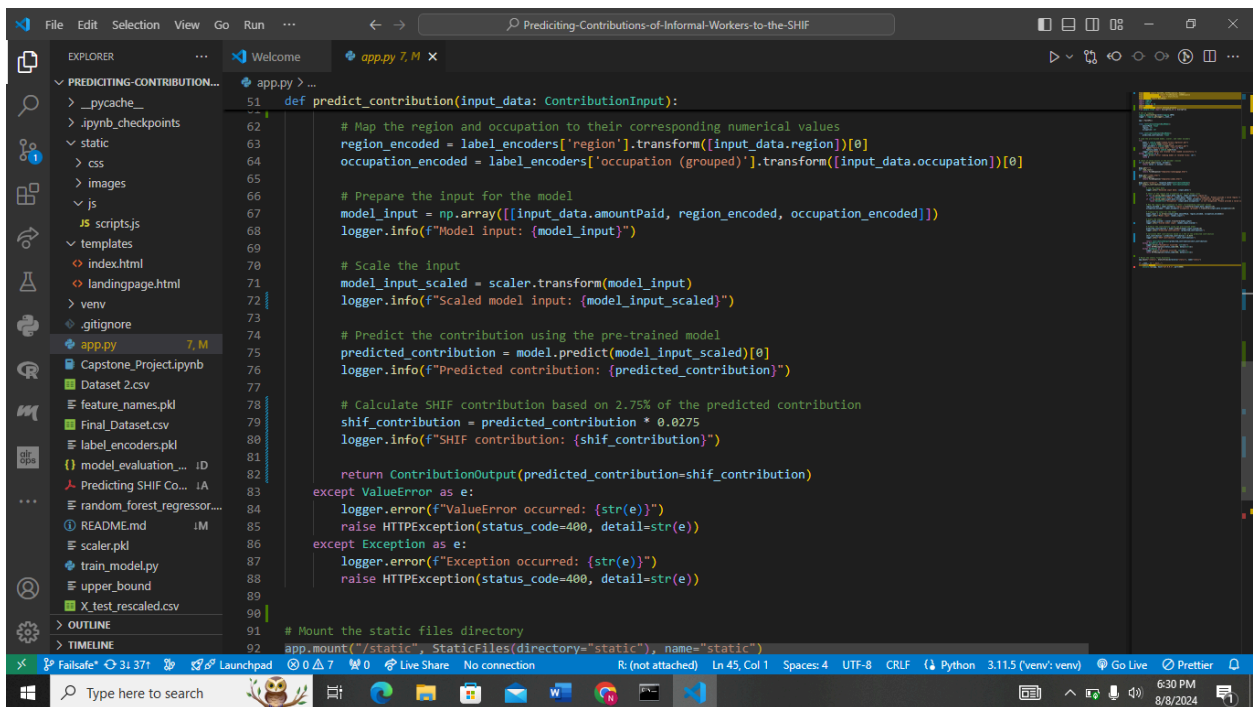
12. Deployment

The train model was Random Forest which was our best model and then flask API was used for our backend to make use of the models to make predictions. Below is a snippet of the code



This screenshot shows the initial setup of a FastAPI application in VS Code. The Explorer panel on the left displays the project structure, including files like `__pycache__`, `ipynb_checkpoints`, `static`, `css`, `images`, `js`, `templates`, `index.html`, `landingpage.html`, `venv`, `.gitignore`, `app.py`, `Capstone_Project.ipynb`, `Dataset 2.csv`, `feature_names.pkl`, `Final_Dataset.csv`, `label_encoders.pkl`, `model_evaluation.... ID`, `Predicting SHIF Co... IA`, `random_forest_regressor...`, `README.md`, `scaler.pkl`, `train_model.py`, `upper_bound`, `X_test_rescaled.csv`, `OUTLINE`, and `TIMELINE`. The main editor shows the `app.py` file with the following code:

```
1 from fastapi import FastAPI, HTTPException, Request
2 from fastapi.responses import FileResponse, JSONResponse
3 from fastapi.staticfiles import StaticFiles
4 from pydantic import BaseModel
5 import logging
6 import joblib
7 import numpy as np
8 import requests
9 from google.oauth2 import service_account
10 from google.cloud import dialogflow_v2 as dialogflow
11
12 # Set up logging
13 logging.basicConfig(level=logging.INFO)
14 logger = logging.getLogger(__name__)
15
16 app = FastAPI()
17
18 class ContributionInput(BaseModel):
19     amountPaid: float
20     region: str
21     occupation: str
22
23 class ContributionOutput(BaseModel):
24     predicted_contribution: float
25
26 # Load the pre-trained model, scaler, and label encoders
27 try:
28     model = joblib.load("random_forest_regressor.pkl")
29     scaler = joblib.load("scaler.pkl")
30     label_encoders = joblib.load("label_encoders.pkl")
31
```



This screenshot shows the implementation of the `predict_contribution` function in the `app.py` file. The function takes `input_data: ContributionInput` as an argument and performs the following steps:

- Map the region and occupation to their corresponding numerical values using `label_encoders`.
- Prepare the input for the model as a numpy array.
- Scale the input using the `scaler`.
- Predict the contribution using the pre-trained model.
- Calculate the SHIF contribution based on 2.75% of the predicted contribution.
- Return the `ContributionOutput` object with the predicted contribution and SHIF contribution.
- Handle exceptions: `ValueError` and `Exception` are caught and logged, and `HTTPException` is raised with status code 400.

```
51 def predict_contribution(input_data: ContributionInput):
52
53     # Map the region and occupation to their corresponding numerical values
54     region_encoded = label_encoders['region'].transform([input_data.region])[0]
55     occupation_encoded = label_encoders['occupation (grouped)'].transform([input_data.occupation])[0]
56
57     # Prepare the input for the model
58     model_input = np.array([[input_data.amountPaid, region_encoded, occupation_encoded]])
59     logger.info(f"Model input: {model_input}")
60
61     # Scale the input
62     model_input_scaled = scaler.transform(model_input)
63     logger.info(f"Scaled model input: {model_input_scaled}")
64
65     # Predict the contribution using the pre-trained model
66     predicted_contribution = model.predict(model_input_scaled)[0]
67     logger.info(f"Predicted contribution: {predicted_contribution}")
68
69     # Calculate SHIF contribution based on 2.75% of the predicted contribution
70     shif_contribution = predicted_contribution * 0.0275
71     logger.info(f"SHIF contribution: {shif_contribution}")
72
73     return ContributionOutput(predicted_contribution-shif_contribution)
74
75 except ValueError as e:
76     logger.error(f"ValueError occurred: {str(e)}")
77     raise HTTPException(status_code=400, detail=str(e))
78
79 except Exception as e:
80     logger.error(f"Exception occurred: {str(e)}")
81     raise HTTPException(status_code=400, detail=str(e))
82
83 # Mount the static files directory
84 app.mount("/static", StaticFiles(directory="static"), name="static")
85
```

The user interface is done in html css and validation is javascript

