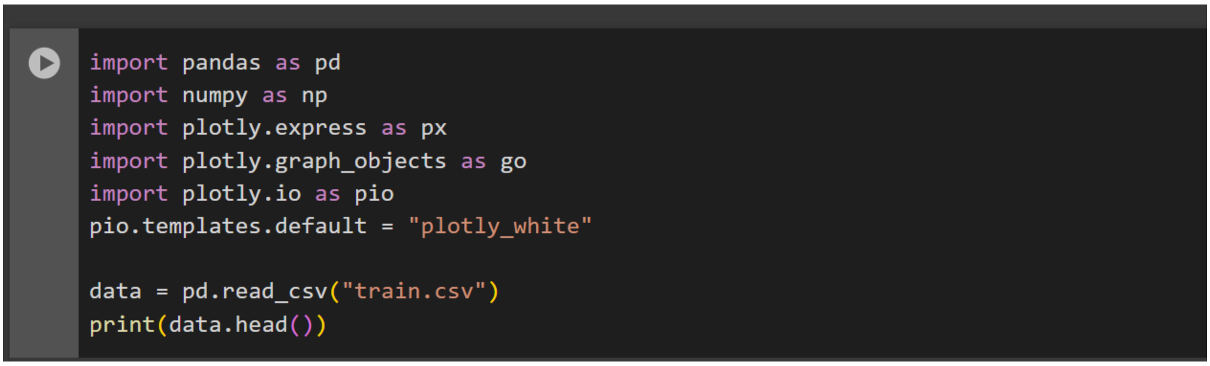
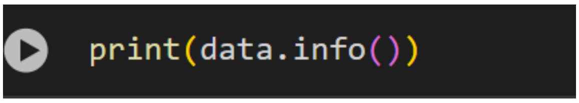
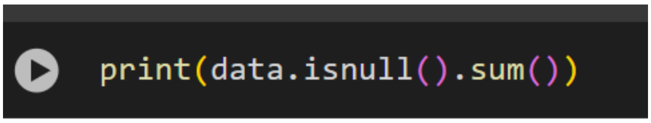
Code on python for making the machine learning model and producing ‘credit_score_model.pkl’ and ‘scaler.pkl’:



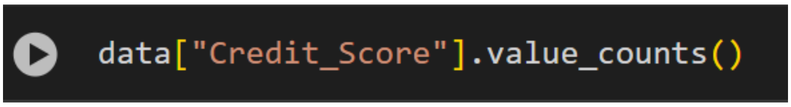
```
import pandas as pd
import numpy as np
import plotly.express as px
import plotly.graph_objects as go
import plotly.io as pio
pio.templates.default = "plotly_white"

data = pd.read_csv("train.csv")
print(data.head())
```



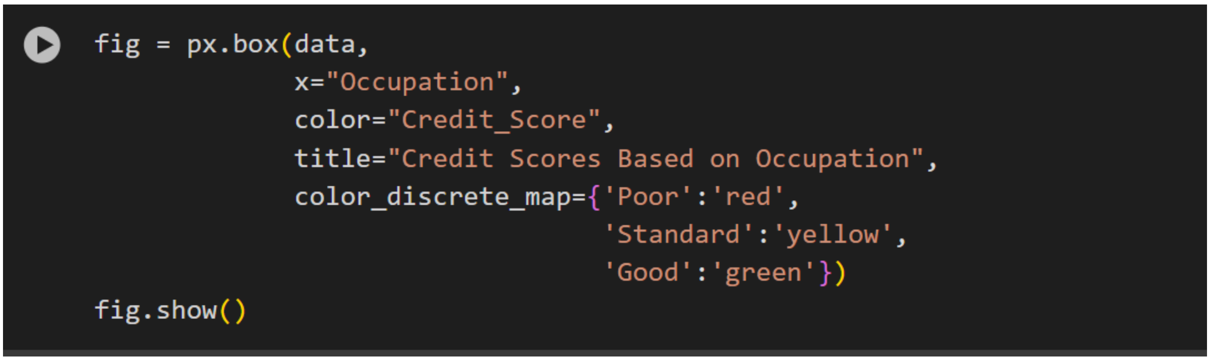
```
print(data.info())
```



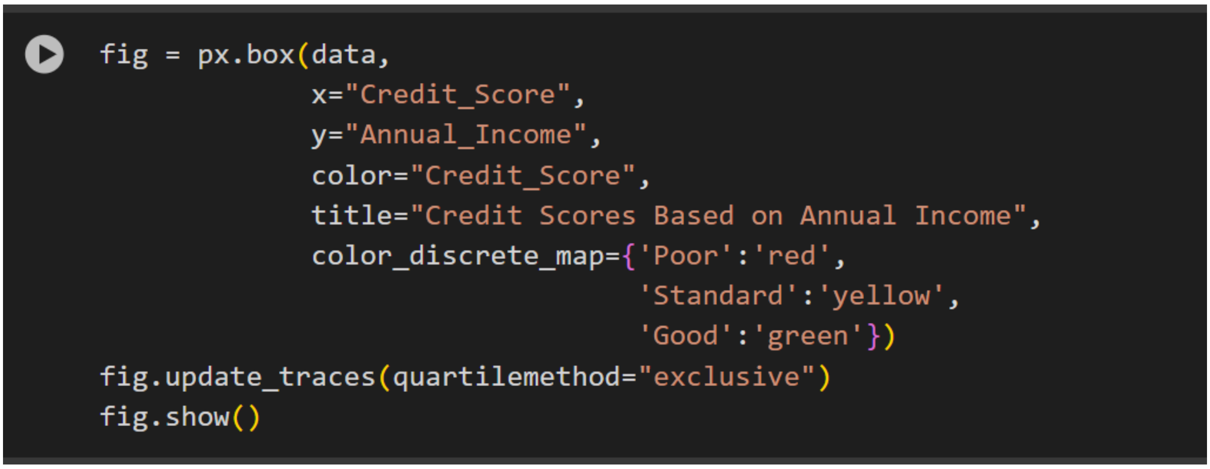
```
print(data.isnull().sum())
```



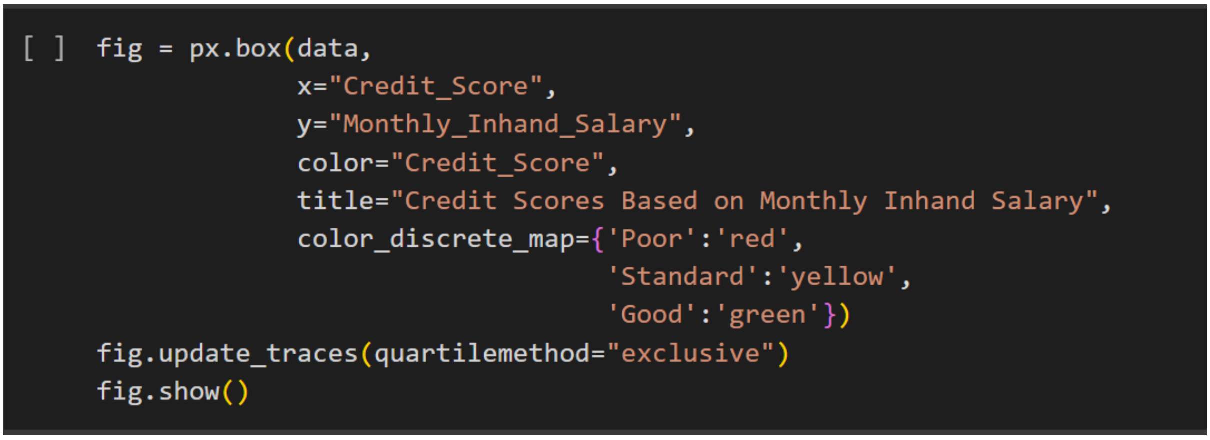
```
data["Credit_Score"].value_counts()
```



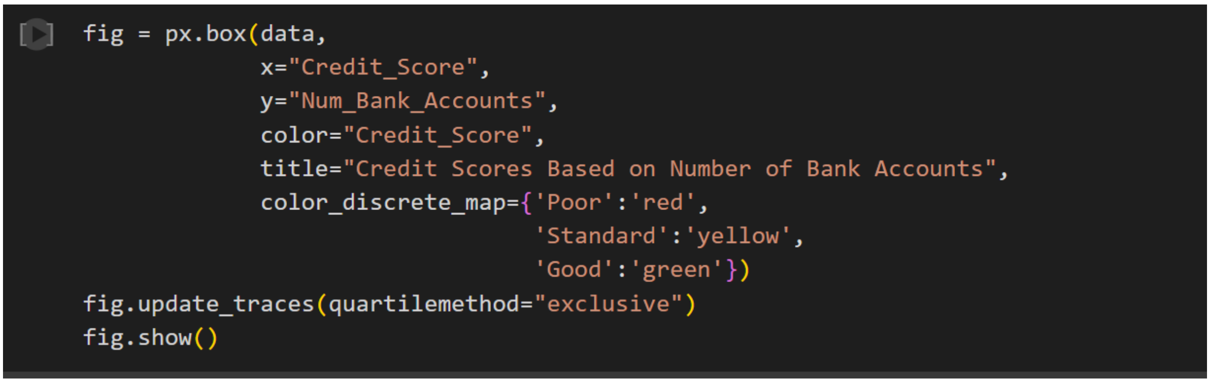
```
fig = px.box(data,
             x="Occupation",
             color="Credit_Score",
             title="Credit Scores Based on Occupation",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.show()
```



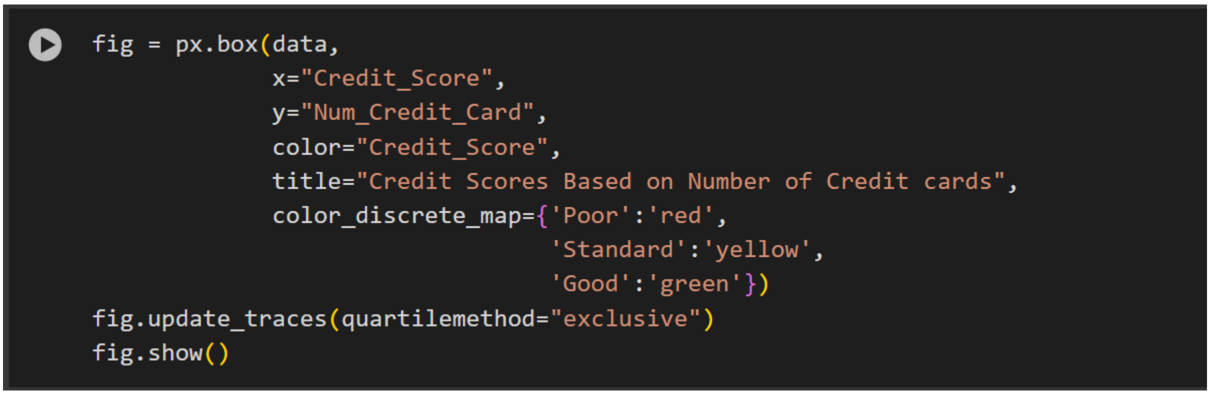
```
fig = px.box(data,
             x="Credit_Score",
             y="Annual_Income",
             color="Credit_Score",
             title="Credit Scores Based on Annual Income",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



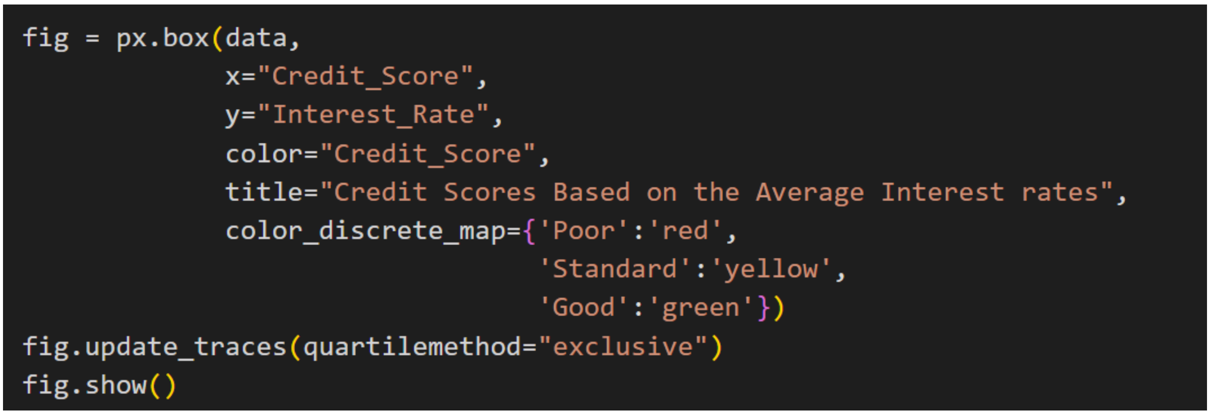
```python
fig = px.box(data,
             x="Credit_Score",
             y="Monthly_Inhand_Salary",
             color="Credit_Score",
             title="Credit Scores Based on Monthly Inhand Salary",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



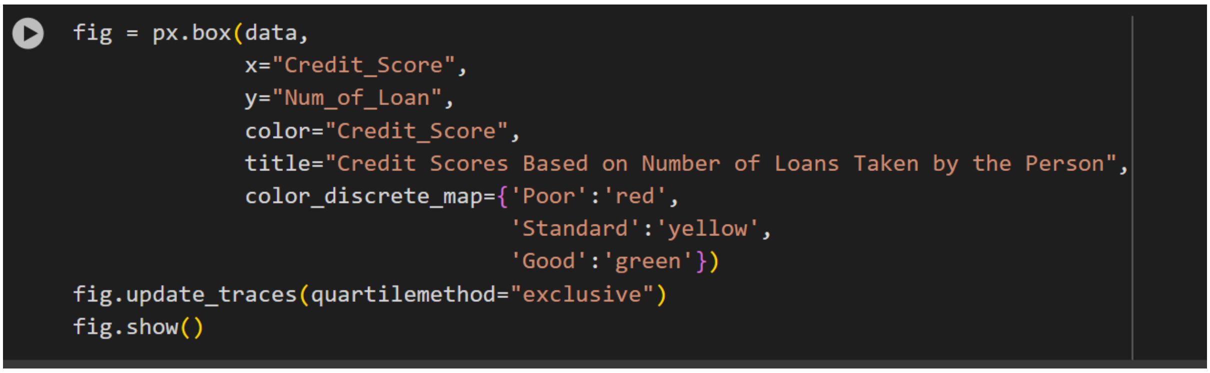
```python
fig = px.box(data,
             x="Credit_Score",
             y="Num_Bank_Accounts",
             color="Credit_Score",
             title="Credit Scores Based on Number of Bank Accounts",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



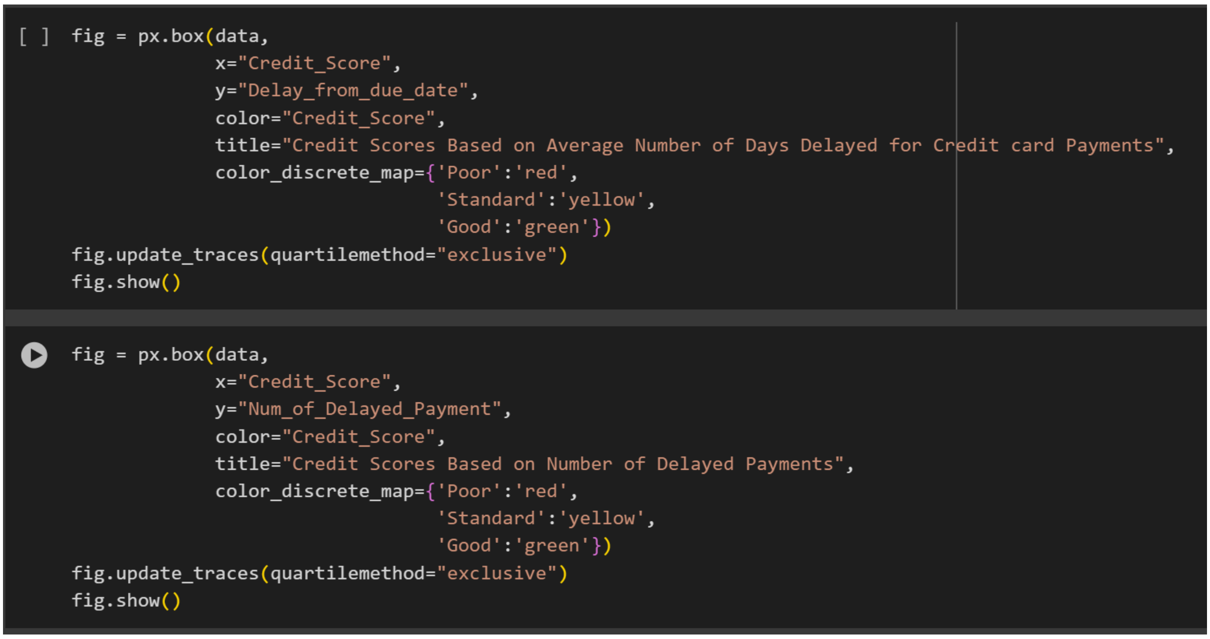
```python
fig = px.box(data,
             x="Credit_Score",
             y="Num_Credit_Card",
             color="Credit_Score",
             title="Credit Scores Based on Number of Credit cards",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



```python
fig = px.box(data,
             x="Credit_Score",
             y="Interest_Rate",
             color="Credit_Score",
             title="Credit Scores Based on the Average Interest rates",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

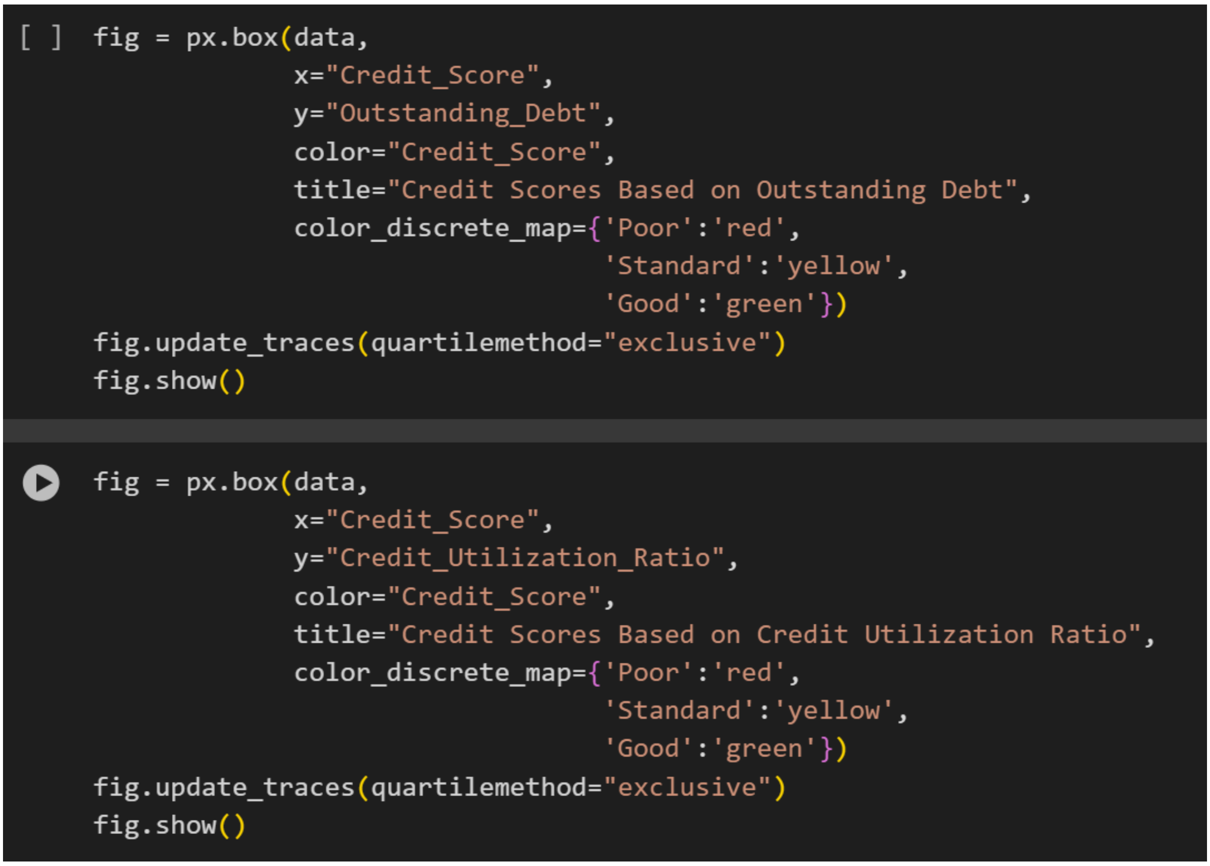


```python
fig = px.box(data,
             x="Credit_Score",
             y="Num_of_Loan",
             color="Credit_Score",
             title="Credit Scores Based on Number of Loans Taken by the Person",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



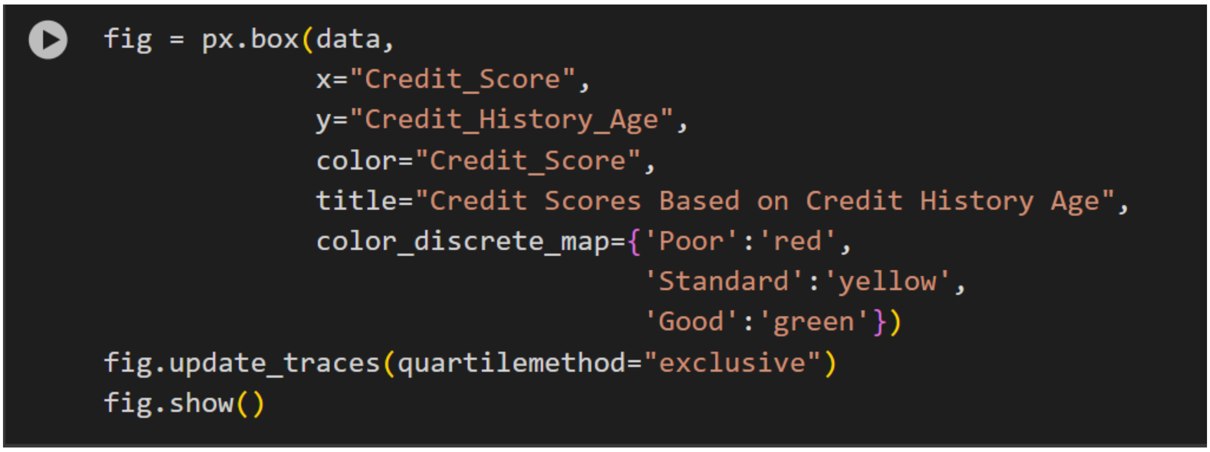
```python
fig = px.box(data,
             x="Credit_Score",
             y="Delay_from_due_date",
             color="Credit_Score",
             title="Credit Scores Based on Average Number of Days Delayed for Credit card Payments",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

```python
fig = px.box(data,
             x="Credit_Score",
             y="Num_of_Delayed_Payment",
             color="Credit_Score",
             title="Credit Scores Based on Number of Delayed Payments",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

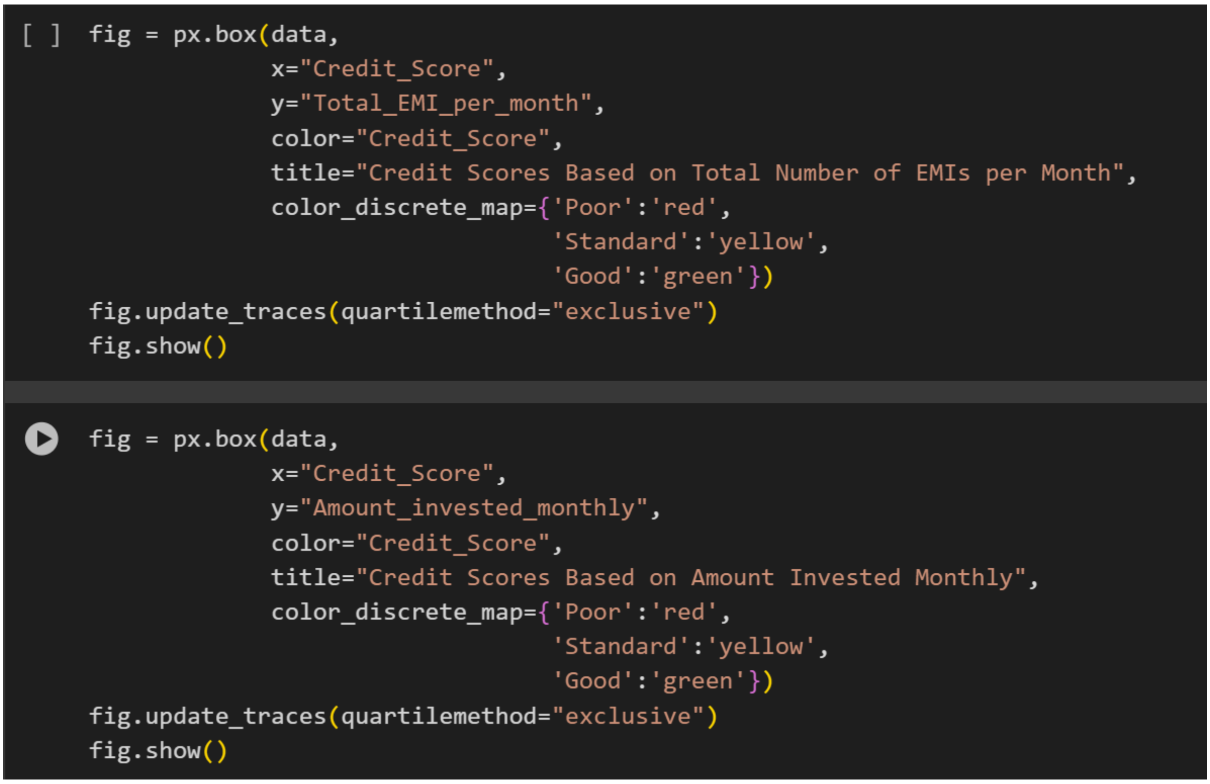


```
fig = px.box(data,
             x="Credit_Score",
             y="Outstanding_Debt",
             color="Credit_Score",
             title="Credit Scores Based on Outstanding Debt",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

```
fig = px.box(data,
             x="Credit_Score",
             y="Credit_Utilization_Ratio",
             color="Credit_Score",
             title="Credit Scores Based on Credit Utilization Ratio",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

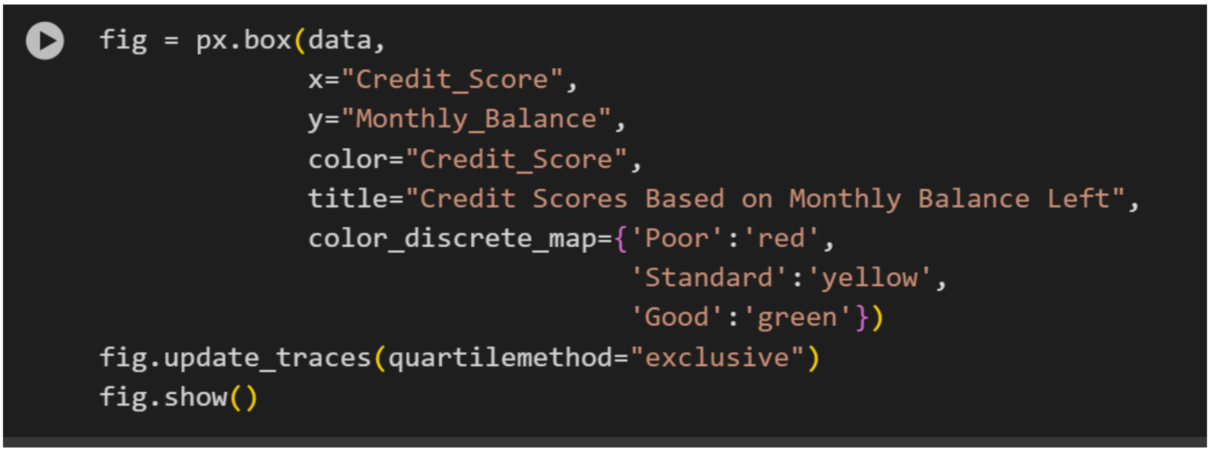


```
fig = px.box(data,
             x="Credit_Score",
             y="Credit_History_Age",
             color="Credit_Score",
             title="Credit Scores Based on Credit History Age",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

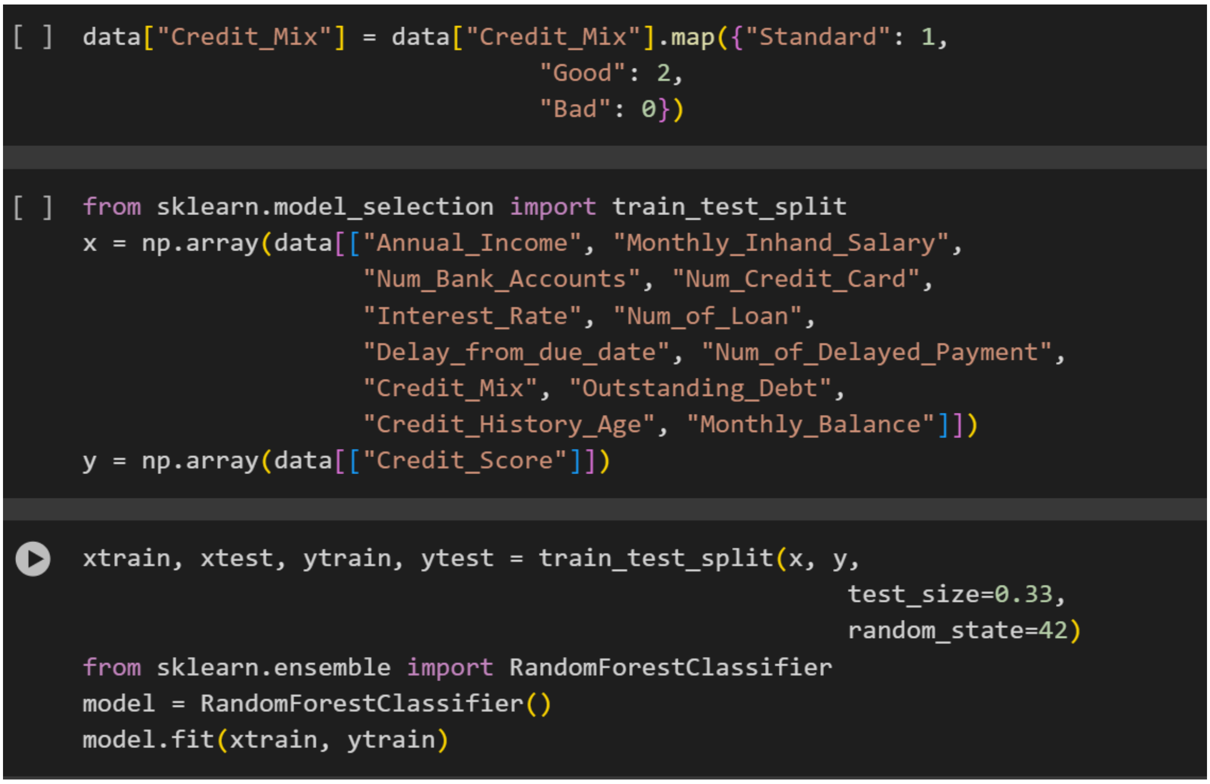


```
fig = px.box(data,
             x="Credit_Score",
             y="Total_EMI_per_month",
             color="Credit_Score",
             title="Credit Scores Based on Total Number of EMIs per Month",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

```
fig = px.box(data,
             x="Credit_Score",
             y="Amount_invested_monthly",
             color="Credit_Score",
             title="Credit Scores Based on Amount Invested Monthly",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```



```
fig = px.box(data,
             x="Credit_Score",
             y="Monthly_Balance",
             color="Credit_Score",
             title="Credit Scores Based on Monthly Balance Left",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()
```

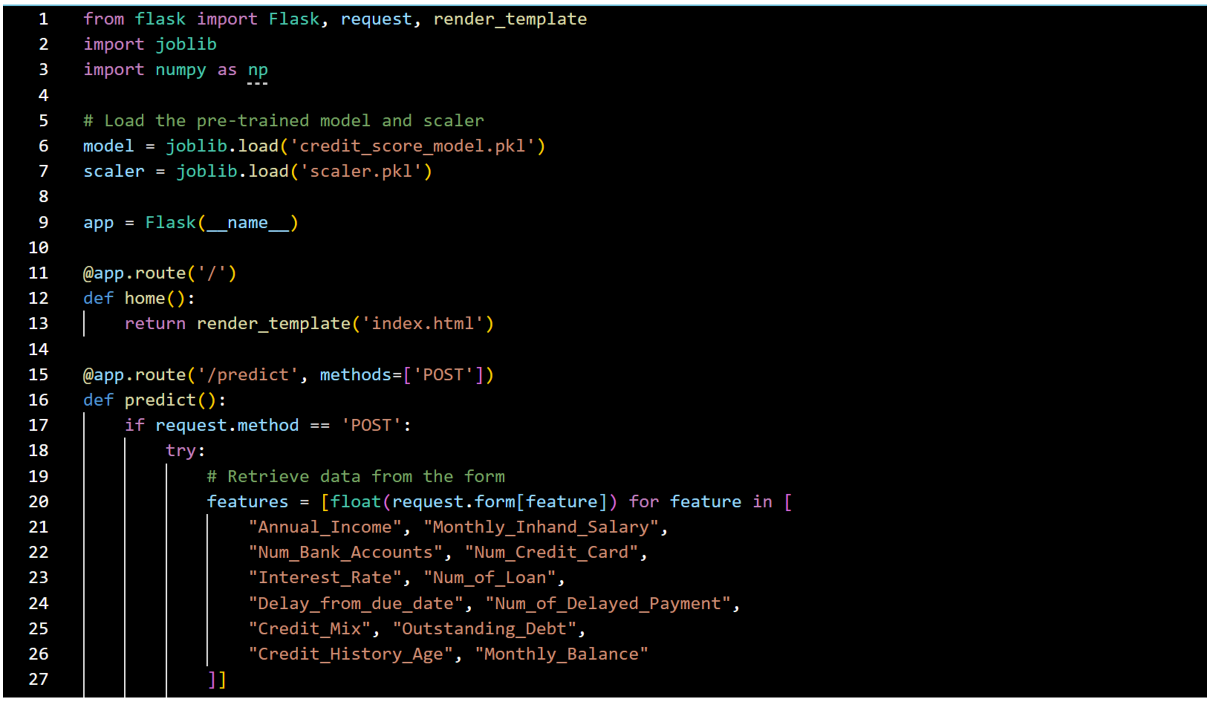


```
data["Credit_Mix"] = data["Credit_Mix"].map({"Standard": 1,
                                "Good": 2,
                                "Bad": 0})
```

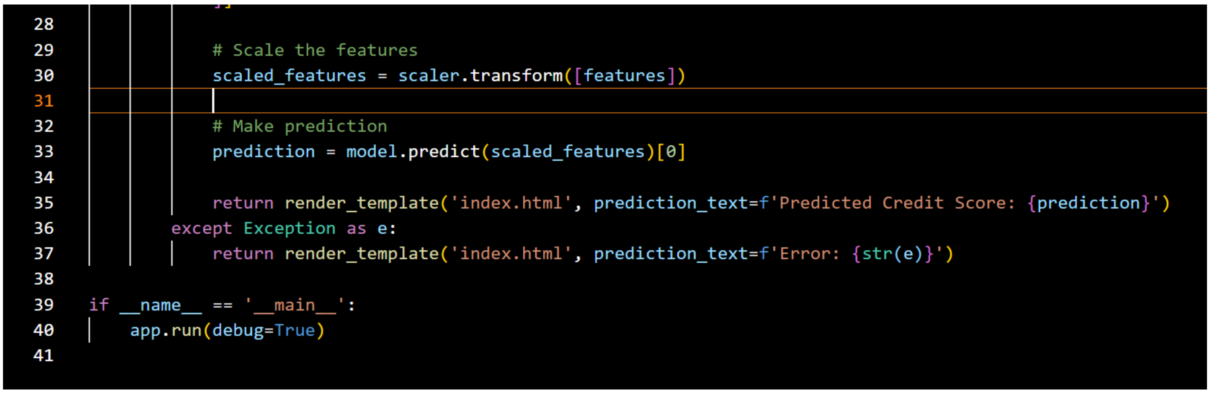
```
from sklearn.model_selection import train_test_split
x = np.array(data[["Annual_Income", "Monthly_Inhand_Salary",
                   "Num_Bank_Accounts", "Num_Credit_Card",
                   "Interest_Rate", "Num_of_Loan",
                   "Delay_from_due_date", "Num_of_Delayed_Payment",
                   "Credit_Mix", "Outstanding_Debt",
                   "Credit_History_Age", "Monthly_Balance"]])
y = np.array(data[["Credit_Score"]])
```

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
                                                     test_size=0.33,
                                                     random_state=42)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier()
model.fit(xtrain, ytrain)
```

Code for creating the app:



```
from flask import Flask, request, render_template
import joblib
import numpy as np

# Load the pre-trained model and scaler
model = joblib.load('credit_score_model.pkl')
scaler = joblib.load('scaler.pkl')

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    if request.method == 'POST':
        try:
            # Retrieve data from the form
            features = [float(request.form[feature]) for feature in [
                "Annual_Income", "Monthly_Inhand_Salary",
                "Num_Bank_Accounts", "Num_Credit_Card",
                "Interest_Rate", "Num_of_Loan",
                "Delay_from_due_date", "Num_of_Delayed_Payment",
                "Credit_Mix", "Outstanding_Debt",
                "Credit_History_Age", "Monthly_Balance"
            ]]
```



```python
            ]]

            # Scale the features
            scaled_features = scaler.transform([features])

            # Make prediction
            prediction = model.predict(scaled_features)[0]

            return render_template('index.html', prediction_text=f'Predicted Credit Score: {prediction}')
        except Exception as e:
            return render_template('index.html', prediction_text=f'Error: {str(e)}')

if __name__ == '__main__':
    app.run(debug=True)
```

Code for the html and css WEBSITE:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Credit Score Prediction</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
    <style>
        body {
            font-family: montserrat, sans-serif;
        }
        .help-modal {
            display: none;
            position: fixed;
            z-index: 1;
            left: 0;
            top: 0;
            width: 100%;
            height: 100%;
            overflow: auto;
            background-color: rgba(0, 0, 0, 0.5);
        }
        .help-content {
            background-color: #fefefe;
            margin: 15% auto;
            padding: 20px;
            border: 1px solid #888;
            width: 80%;
            max-width: 600px;
        }
        .help-content h1, .help-content h2 {
            color: #2c3e50;
        }
        .help-content p {
            margin-bottom: 20px;
        }
        .close-btn {
            color: #aaa;
            float: right;
            font-size: 28px;
            font-weight: bold;
        }
        .close-btn:hover,
        .close-btn:focus {
            color: black;
            text-decoration: none;
            cursor: pointer;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Credit Score Prediction</h2>
```

```
<form action="/predict" method="post">
    <div class="form-group">
        <label for="Annual_Income">Annual Income</label>
        <input type="text" name="Annual_Income" required>
    </div>
    <div class="form-group">
        <label for="Monthly_Inhand_Salary">Monthly In-hand Salary</label>
        <input type="text" name="Monthly_Inhand_Salary" required>
    </div>
    <div class="form-group">
        <label for="Num_Bank_Accounts">Number of Bank Accounts</label>
        <input type="text" name="Num_Bank_Accounts" required>
    </div>
    <div class="form-group">
        <label for="Num_Credit_Card">Number of Credit Cards</label>
        <input type="text" name="Num_Credit_Card" required>
    </div>
    <div class="form-group">
        <label for="Interest_Rate">Interest Rate</label>
        <input type="text" name="Interest_Rate" required>
    </div>
    <div class="form-group">
        <label for="Num_of_Loan">Number of Loans</label>
        <input type="text" name="Num_of_Loan" required>
    </div>
    <div class="form-group">
        <label for="Delay_from_due_date">Average Days Delayed</label>
        <input type="text" name="Delay_from_due_date" required>
    </div>
    <div class="form-group">
```

```
            <label for="Num_of_Delayed_Payment">Number of Delayed Payments</label>
            <input type="text" name="Num_of_Delayed_Payment" required>
        </div>
        <div class="form-group">
            <label for="Credit_Mix">Credit Mix (1: Bad, 2: Standard, 3: Good)</label>
            <input type="text" name="Credit_Mix" required>
        </div>
        <div class="form-group">
            <label for="Outstanding_Debt">Outstanding Debt</label>
            <input type="text" name="Outstanding_Debt" required>
        </div>
        <div class="form-group">
            <label for="Credit_History_Age">Credit History Age</label>
            <input type="text" name="Credit_History_Age" required>
        </div>
        <div class="form-group">
            <label for="Monthly_Balance">Monthly Balance</label>
            <input type="text" name="Monthly_Balance" required>
        </div>
        <button type="submit">Predict</button>
    </form>
    {% if prediction_text %}
        <h3>{{ prediction_text }}</h3>
    {% endif %}
</div>
<a href="#" id="help-link">Help</a>

<div id="help-modal" class="help-modal">
    <div class="help-content">
        <span class="close-btn" id="close-help">&times;</span>
```

```html
<h1>Credit Information</h1>


<h2>Credit Scores</h2>

<p>There are three credit scores that banks and credit card companies use to label their customers:</p>

<ul>

   <li>Good</li>

   <li>Standard</li>

   <li>Poor</li>

</ul>

<p>A person with a good credit score will get loans from any bank and financial institution.</p>


<h2>Annual Income</h2>

<p>Annual income refers to the total amount of money earned by an individual or entity over the course of a year. It encompasses all sources of income, which can include wages, salaries, bonuses, investments, rental income, and any other forms of earnings.</p>


<h2>Monthly In-hand Salary</h2>

<p>Monthly in-hand salary, also known as net salary or take-home pay, is the amount of money an individual receives after all deductions have been made from their gross monthly salary. These deductions can include taxes, insurance premiums, retirement contributions, and other mandatory or voluntary deductions.</p>


<h2>Credit Card</h2>

<p>A credit card is a financial tool issued by banks or financial institutions that allows cardholders to borrow funds to pay for goods and services. These borrowed funds must be repaid, typically with interest, according to the terms set by the credit card issuer.</p>


<h2>Interest Rate</h2>

<p>An interest rate is the percentage charged on the total amount of money borrowed or earned, typically expressed on an annual basis. Interest rates are used in various financial products and transactions, such as loans, mortgages, savings accounts, and credit cards. The rate determines the cost of borrowing money or the return on investment.</p>
```

```
        <h2>Credit Mix</h2>
        <p>Credit mix refers to the variety of credit accounts that a borrower has. It is an important factor in determining a person's credit score and reflects the different types of credit they are managing. A diverse credit mix demonstrates a borrower’s ability to handle different types of credit responsibly, which can positively impact their credit score.</p>

        <h2>Outstanding Debt</h2>
        <p>Outstanding debt refers to the total amount of money that an individual, company, or entity currently owes to creditors. This includes all unpaid balances on loans, credit cards, lines of credit, and any other forms of borrowing. Outstanding debt represents the amount that is still due to be paid back, including any accrued interest and fees.</p>

        <h2>Credit History Age</h2>
        <p>Credit history age, also known as the length of credit history, refers to the duration of time that has passed since a person first opened their credit accounts. It is a key factor in determining an individual's credit score and reflects how long they have been using credit.</p>

        <h2>Monthly Balance</h2>
        <p>Monthly balance is the average, ending, or outstanding amount in an account or on a credit card at the end of a month.</p>
      </div>
    </div>

    <script>
      document.getElementById('help-link').onclick = function() {
        document.getElementById('help-modal').style.display = 'block';
      }
      document.getElementById('close-help').onclick = function() {
        document.getElementById('help-modal').style.display = 'none';
      }
      window.onclick = function(event) {
        if (event.target == document.getElementById('help-modal')) {
                document.getElementById('help-modal').style.display = 'none';
            }
        }
    </script>
</body>
</html>
```

CSS:

```
body {
    font-family: montserrat, sans-serif;
    background-color: #f8f9fa;
    background-image: url('gettyimages.jpg');
    background-size: cover;
    background-position: center;
}

.container {
    max-width: 500px;
    margin: 50px auto;
    padding: 20px;
    background-color: white;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

h2 {
    text-align: center;
    margin-bottom: 20px;
}
```

```
.form-group {
    margin-bottom: 15px;
}

label {
    display: block;
    margin-bottom: 5px;
}

input[type="text"] {
    width: 100%;
    padding: 8px;
    box-sizing: border-box;
}

button {
    width: 100%;
    padding: 10px;
    background-color: #007bff;
    color: white;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}
```