CSCE 5210 - Fundamentals of Artificial Intelligence
Final Report
Group 11
Credit Risk Assessment and Stock Market Analysis

Team Members:
Suresh John Wesly Ginjupalli – 11629157
Manohar Bravara – 11556820
Jaideep Janapati -- 11636384
Navyatha Yamsani -- 11636386
Cherishma Meher Medisetty – 11638081

Abstract:
The Credit Risk Assessment System and Stock Market Analysis System represent cutting-edge financial technology solutions designed to empower institutions and market participants. The Credit Risk Assessment System employs advanced machine learning and data analysis to optimize credit evaluation, aiding banks in making precise loan decisions, reducing risks, and ensuring fair lending standards. Simultaneously, the Stock Market Analysis System equips investors with a comprehensive toolkit, leveraging data analytics and forecasting models to provide real-time market insights, predictive analytics, and sentiment analysis. Developed with scalability in mind, these systems utilize Python for algorithm implementation and web-based interfaces for user interaction. Incorporating features such as Credit Scoring, Fraud Detection, Stock Price Prediction, and Trend Analysis, these solutions offer users a versatile platform for making data-driven decisions in the finance sector.

Introduction:
In the dynamic and intricate landscape of financial technology, the Credit Risk Assessment System and Stock Market Analysis System emerge as cutting-edge solutions, purposefully crafted to empower individuals and organizations within the finance sector. These state-of-the-art systems have a singular objective: to enhance decision-making processes by deploying advanced machine learning and data analysis techniques. The Credit Risk Assessment System, at its core, transforms the conventional credit evaluation process for banks, lending institutions, and credit providers. Through the adept utilization of historical borrower data, demographic information, and predictive models, this system not only facilitates more accurate loan decisions but also mitigates risks and ensures adherence to fair lending standards, optimizing the allocation of financial resources.

Simultaneously, the Stock Market Analysis System caters to the diverse needs of investors, traders, and financial analysts by offering a comprehensive toolkit for navigating the intricate world of financial markets. In an era where real-time market insights, predictive analytics, and sentiment analysis are paramount, this system leverages advanced data analytics, time-series forecasting models, and natural language processing techniques. By amalgamating historical stock data with sentiment analysis derived from news articles and social media, the Stock Market Analysis System empowers users to formulate informed trading strategies, optimize portfolios, and stay ahead in the ever-evolving financial landscape. Together, these systems represent not only technological advancements but also a transformative force in enabling more informed, data-driven decision-making in the complex realm of finance.

Area of Application:

Credit Risk Assessment System: Primarily designed for the finance sector, this system targets banks, lending institutions, and credit providers. Its application lies in transforming the credit evaluation process, assisting in assessing the creditworthiness of loan applicants, reducing risks, and ensuring compliance with fair lending standards.

Stock Market Analysis System: Tailored for investors, traders, and financial analysts, this system finds its application in navigating the complexities of financial markets. It provides tools for real-time market insights, predictive analytics, and sentiment analysis, catering to a broad spectrum of market participants from individual investors to institutional traders.

Dataset:

The dataset captures diverse socio-demographic and financial attributes of individuals. Attributes such as 'Gender,' 'Has a car,' 'Has a property,' 'Children count,' 'Income,' 'Employment status,' 'Education level,' 'Marital status,' 'Dwelling,' 'Age,' 'Employment length,' 'Has a mobile phone,' 'Has a work phone,' 'Has a phone,' 'Has an email,' 'Job title,' 'Family member count,' and 'Account age' are used. These attributes suggest a comprehensive exploration of factors including personal and financial details, housing, and employment characteristics. The dataset is likely to be valuable for analyses related to credit risk assessment, financial profiling, or demographic studies, given the richness of information captured across various dimensions of individuals lives.

```
cc_data_full_data = cc_data_full_data.rename(columns={
    'CODE_GENDER':'Gender',
    'FLAG_OWN_CAR':'Has a car',
    'FLAG_OWN_REALTY':'Has a property',
    'CNT_CHILDREN':'Children count',
    'AMT_INCOME_TOTAL':'Income',
    'NAME_INCOME_TYPE':'Employment status',
    'NAME_EDUCATION_TYPE':'Education level',
    'NAME_FAMILY_STATUS':'Marital status',
    'NAME_HOUSING_TYPE':'Dwelling',
    'DAYS_BIRTH':'Age',
    'DAYS_EMPLOYED': 'Employment length',
    'FLAG_MOBIL': 'Has a mobile phone',
    'FLAG_WORK_PHONE': 'Has a work phone',
    'FLAG_PHONE': 'Has a phone',
    'FLAG_EMAIL': 'Has an email',
    'OCCUPATION_TYPE': 'Job title',
    'CNT_FAM_MEMBERS': 'Family member count',
    'Account age': 'Account age'
    })
```

	ID	Children count	Income	Age	Employment length	Has a mobile phone	Has a work phone	Has a phone	Has an email	Family member count	Account age
count	3.645700e+04	36457.000000	3.645700e+04	36457.000000	36457.000000	36457.0	36457.000000	36457.000000	36457.000000	36457.000000	36457.000000
mean	5.078227e+06	0.430315	1.866857e+05	-15975.173382	59262.935568	1.0	0.225526	0.294813	0.089722	2.198453	-26.164193
std	4.187524e+04	0.742367	1.017892e+05	4200.549944	137651.334859	0.0	0.417934	0.455965	0.285787	0.911686	16.501854
min	5.008804e+06	0.000000	2.700000e+04	-25152.000000	-15713.000000	1.0	0.000000	0.000000	0.000000	1.000000	-60.000000
25%	5.042028e+06	0.000000	1.215000e+05	-19438.000000	-3153.000000	1.0	0.000000	0.000000	0.000000	2.000000	-39.000000
50%	5.074614e+06	0.000000	1.575000e+05	-15563.000000	-1552.000000	1.0	0.000000	0.000000	0.000000	2.000000	-24.000000
75%	5.115396e+06	1.000000	2.250000e+05	-12462.000000	-408.000000	1.0	0.000000	1.000000	0.000000	3.000000	-12.000000
max	5.150487e+06	19.000000	1.575000e+06	-7489.000000	365243.000000	1.0	1.000000	1.000000	1.000000	20.000000	0.000000

Features:

Credit Risk Assessment System:

1. Credit Scoring Model: Evaluates creditworthiness and predicts the likelihood of loan default.
2. Fraud Detection: Implements algorithms to identify suspicious patterns in loan applications, reducing the risk of fraudulent loans.
3. Machine Learning Model Monitoring: Continuously monitors the performance and accuracy of machine learning models used in credit scoring.
4. Machine Learning Model Ensemble: Combines multiple machine learning models into an ensemble for more robust and accurate credit scoring.

Stock Market Analysis System:

1. Stock Price Prediction: Provides real-time and historical stock price predictions, aiding investors and traders.
2. Trend Analysis: Identifies and visualizes stock price trends and patterns, helping users understand market behavior.
3. Multi-Asset Support: Expands the system's coverage to include various asset classes, such as stocks, bonds, commodities, and forex, allowing for diversified analysis.
4. Alternative Data Integration: Explores integration with alternative data sources, such as satellite imagery or social media data, for unique trading insights.

Methods:

The Credit Risk Assessment System and Stock Market Analysis System employ sophisticated methods tailored to their specific financial applications.

Credit Risk Assessment System:

1. Machine Learning Models: The Credit Risk Assessment System leverages advanced machine learning techniques to transform the credit evaluation process. By harnessing historical borrower data, demographic information, and predictive models, the system employs machine learning algorithms to assess creditworthiness and predict the likelihood of loan defaults, contributing to more accurate loan decisions.

2. Fraud Detection Algorithms: The system incorporates sophisticated fraud detection algorithms to identify irregular patterns or inconsistencies in loan applications. This method enhances the system's ability to detect and mitigate the risk of fraudulent loans, ensuring the integrity of the credit evaluation process.

3. Ensemble Learning: To enhance the robustness of credit scoring, the system utilizes ensemble learning, combining multiple machine learning models. This approach aims to improve the accuracy of credit scoring by aggregating insights from diverse models, offering a more comprehensive assessment of creditworthiness.

Stock Market Analysis System:

1. Data Analytics and Time-Series Forecasting Models: The Stock Market Analysis System employs advanced data analytics and time-series forecasting models to provide real-time market insights. By analyzing historical stock data and identifying trends, these techniques contribute to accurate stock price predictions and a deeper understanding of market behavior.

2. Natural Language Processing (NLP): The system incorporates natural language processing techniques for sentiment analysis of news articles and social media data. This method allows the system to gauge market sentiment by analyzing textual data, providing valuable insights for users making investment decisions.

3. Alternative Data Integration: In the realm of stock market analysis, the system explores the integration of alternative data sources, such as satellite imagery or social media data. This approach broadens the scope of analysis beyond traditional financial data, offering unique perspectives and insights for users navigating the financial markets.

4. Web-Based Interface and Python Programming: Both systems prioritize user interaction through a web-based interface. The implementation of algorithms and system development is carried out using Python, a widely adopted programming language known for its versatility and effectiveness in data analysis and machine learning applications.

These methods collectively underscore the advanced and comprehensive approaches taken in the development of these financial technology solutions.

Modules:

The code used showcases a diverse set of Python libraries, each serving a distinct purpose in data manipulation, visualization, machine learning, and financial data retrieval. Here's a concise overview of the imported libraries:

1. random: Facilitates the generation of random numbers and sequences, applicable in scenarios like simulations or data shuffling.

2. pandas: A robust data manipulation library, pivotal for data analysis and manipulation. It features DataFrame structures for efficient handling of structured data.

3. streamlit: A library designed for creating web applications with minimal code, commonly utilized for developing interactive dashboards in data science and machine learning.

4. yfinance: Enables the retrieval of financial data from Yahoo Finance, particularly valuable for obtaining stock market data.

5. matplotlib and seaborn: Visualization libraries employed to create diverse plots and charts, aiding in the visual representation of data.

6. numpy: A numerical operations library supporting large, multi-dimensional arrays and matrices, crucial for efficient numerical computations.

7. Imblearn: Specializing in handling imbalanced datasets, this library includes tools like SMOTE for oversampling techniques in machine learning.

8. sklearn: Scikit-learn, a comprehensive machine learning library, encompasses classifiers, preprocessing techniques, and model evaluation tools.

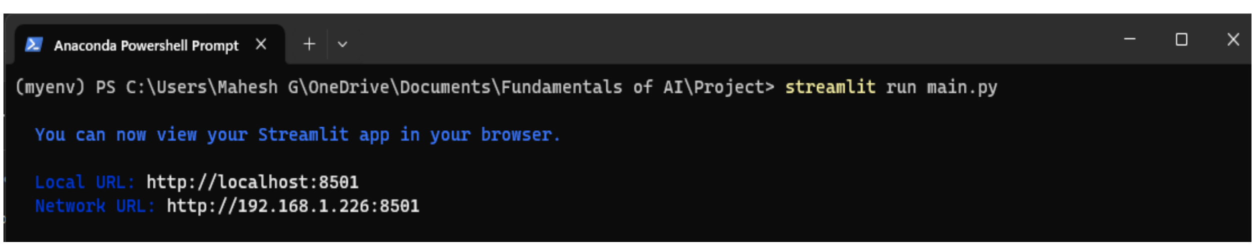
9. tensorflow: Developed by Google, TensorFlow is an open-source machine learning library, particularly favored for deep learning applications.

10. datetime: A module dedicated to managing dates and times, essential for handling time-related data.

11. pandas_datareader: Facilitates the fetching of financial data from diverse online sources, commonly utilized for retrieving stock prices and related financial information.

12. joblib: Primarily used for lightweight pipelining in Python, this library efficiently saves and loads large objects, such as machine learning models.

The code exhibits a versatile combination of functionalities, including data manipulation, financial data retrieval, machine learning, and deep learning.



Code:

Credit Risk Assessment:

```
def full_pipeline(df):
    # Create the pipeline that will call all the class from OutlierRemoval to OversampleSMOTE in one go
    pipeline = Pipeline([
        ('outlier_remover', OutlierRemover()),
        ('feature_dropper', DropFeatures()),
        ('time_conversion_handler', TimeConversionHandler()),
        ('retiree_handler', RetireeHandler()),
        ('skewness_handler', SkewnessHandler()),
        ('binning_num_to_yn', BinningNumToYN()),
        ('one_hot_with_feat_names', OneHotWithFeatNames()),
        ('ordinal_feat_names', OrdinalFeatNames()),
        ('min_max_with_feat_names', MinMaxWithFeatNames()),
        ('change_to_num_target', ChangeToNumTarget()),
        ('oversample_smote', OversampleSMOTE())
    ])
    df_pipe_prep = pipeline.fit_transform(df)
    return df_pipe_prep

# Save the entire pipeline (including transformers and model)
pipeline_model_path = "pipeline_model.joblib"

#Running the Model
loaded_pipeline = joblib.load(pipeline_model_path)

    profile_to_predict_df = pd.DataFrame([profile_to_predict],columns=test_copy.columns)

    # add the profile to predict as a last row in the train data
    test_copy_with_profile_to_pred = pd.concat([test_copy,profile_to_predict_df],ignore_index=True)

    # whole dataset prepared
    test_copy_with_profile_to_pred_prep = full_pipeline(test_copy_with_profile_to_pred)

    # Get the row with the ID = 0, and drop the ID, and target(placeholder) column
    profile_to_pred_prep = test_copy_with_profile_to_pred_prep.drop(['Is high risk'], axis = 1)
```


```
# add the profile to predict as a last row in the train data
test_copy_with_profile_to_pred = pd.concat([test_copy,profile_to_predict_df],ignore_index=True)

# whole dataset prepared
test_copy_with_profile_to_pred_prep = full_pipeline(test_copy_with_profile_to_pred)

# Get the row with the ID = 0, and drop the ID, and target(placeholder) column
profile_to_pred_prep = test_copy_with_profile_to_pred_prep.drop(['Is high risk'], axis = 1)
```


```
# Use the loaded model for prediction
# Button
if st.button('Predict'):

    prediction = loaded_pipeline.predict(profile_to_pred_prep)
    print(prediction)
    if prediction[0]==1:
        st.warning("Fraud Detected: This application has a high likelihood of fraud.")
    else:
        st.success("No Fraud Detected: The application is likely legitimate.")
```

The code implements a pipeline model for data preprocessing and prediction, specifically designed for fraud detection.

1. Pipeline Definition:

The `full_pipeline` function is defined to create a comprehensive data preprocessing pipeline. It includes various data transformation steps encapsulated in custom classes (`OutlierRemover`, `DropFeatures`, `TimeConversionHandler`,`RetireeHandler`,`SkewnessHandler`, `BinningNumToYN`, `OneHotWithFeatNames`, `OrdinalFeatNames`, `MinMaxWithFeatNames`, `ChangeToNumTarget`, and `OversampleSMOTE`).

These classes handle tasks such as outlier removal, feature dropping, time conversion, handling retiree-related data, skewness correction, binning numerical features, one-hot encoding, handling ordinal features, applying min-max scaling, changing the target variable to numerical format, and oversampling using SMOTE (Synthetic Minority Over-sampling Technique).

2. Pipeline Model Saving:

The entire pipeline, including transformers and the model, is saved as a joblib file named "pipeline_model.joblib."

3. Pipeline Execution:

The saved pipeline is loaded using `joblib.load pipeline_model_path. A new data profile `profile_to_predict` is created and appended to the existing test dataset `test_copy`. The entire dataset, including the new profile, undergoes preprocessing using the `full_pipeline` function. The processed dataset is then prepared for prediction by dropping the target variable and the placeholder column.

4. Prediction:

Upon clicking the 'Predict' button, the loaded pipeline is utilized to predict whether the newly added profile is associated with fraud. The prediction result is displayed, with corresponding messages indicating whether fraud is detected or not.

Stock Market Analysis:

We implement a time series prediction model using a Long Short-Term Memory (LSTM) neural network.

1. LSTM Model Definition:

A Sequential model is created using Keras, a high-level neural networks API. The model architecture consists of two LSTM layers with 128 and 64 units, respectively. The first layer returns sequences, and the second does not. Two Dense layers with 25 and 1 units are added, responsible for transforming the LSTM output into the final prediction. The model is compiled using the Adam optimizer and mean squared error loss.

```
model = Sequential()
model.add(LSTM(128, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(LSTM(64, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mean_squared_error')
```

2. Model Training:

The model is trained using the training data (`x_train` and `y_train`) with a batch size of 1 and for one epoch.

```
model.fit(x_train, y_train, batch_size=1, epochs=1)
```

3. Prediction and Evaluation:

The model is then used to predict values on the test data (`x_test`), and the predictions are transformed back to the original scale using an inverse scaler. The root mean squared error (RMSE) is calculated to evaluate the model's performance.

```
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
rmse = np.sqrt(np.mean(((predictions - y_test) ** 2)))
```

4. Visualization:

The predicted values are visualized alongside the actual data using matplotlib. The training data, validation data `valid`, and the model predictions are plotted.

```
plt.figure(figsize=(16, 6))
plt.title('Model')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Close Price USD ($)', fontsize=18)
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
plt.legend(['Train', 'Val', 'Predictions'], loc='lower right')
```

5. **Streamlit Integration:**

- The Streamlit library is used for creating an interactive web application.
- The `simulate_credit_risk_assessment` and `simulate_stock_market_analysis` functions seem to simulate different aspects of the financial analysis, with results being displayed using Streamlit.

```
st.line_chart(valid)
st.write(valid)
```

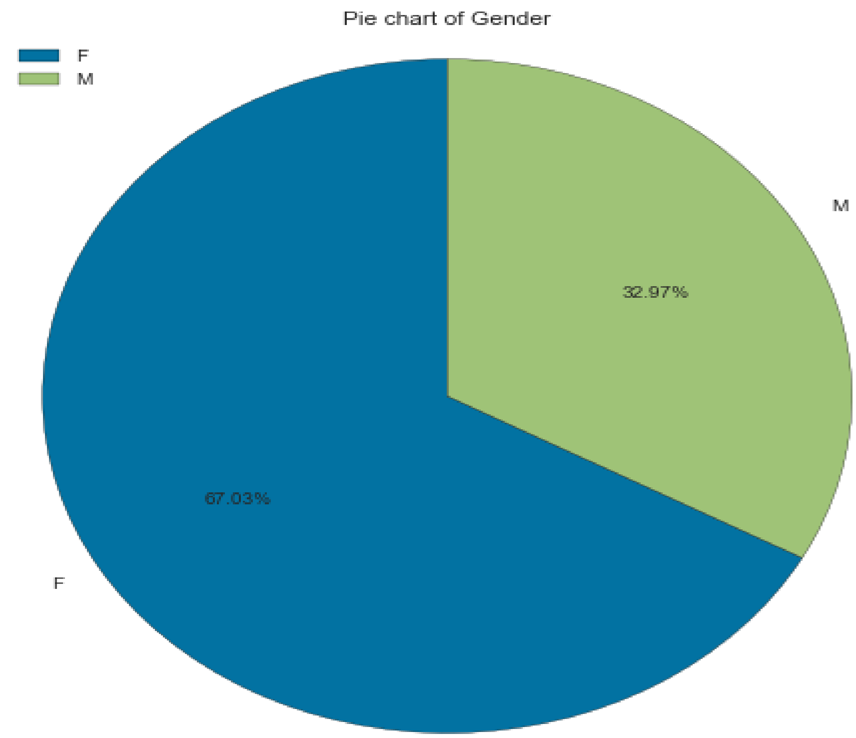
6. Main Method:

The `main` function sets up a Streamlit sidebar with two options: "Credit Risk Assessment" and "Stock Market Analysis." Based on the selected option, the corresponding simulation function is called.

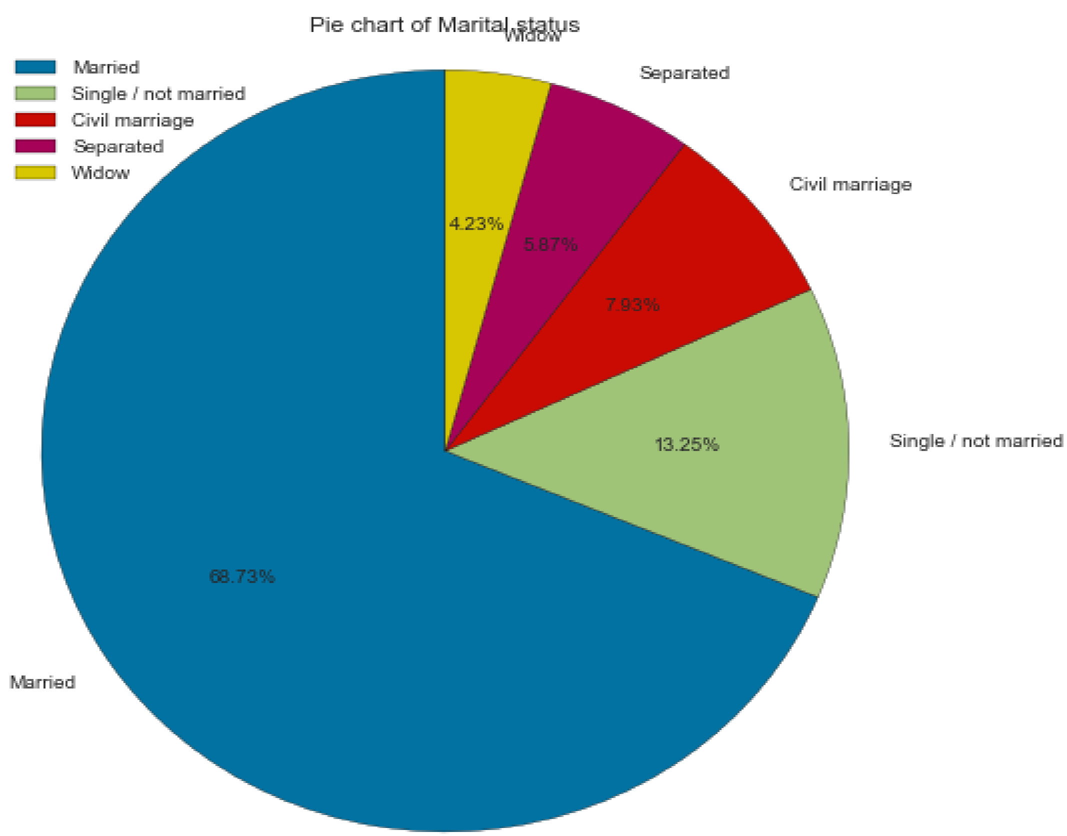
```
if __name__ == "__main__":
    main()
```

Overall, the code integrates LSTM for time series prediction, evaluates the model, visualizes the results, and uses Streamlit for creating an interactive web application to simulate financial analyses.

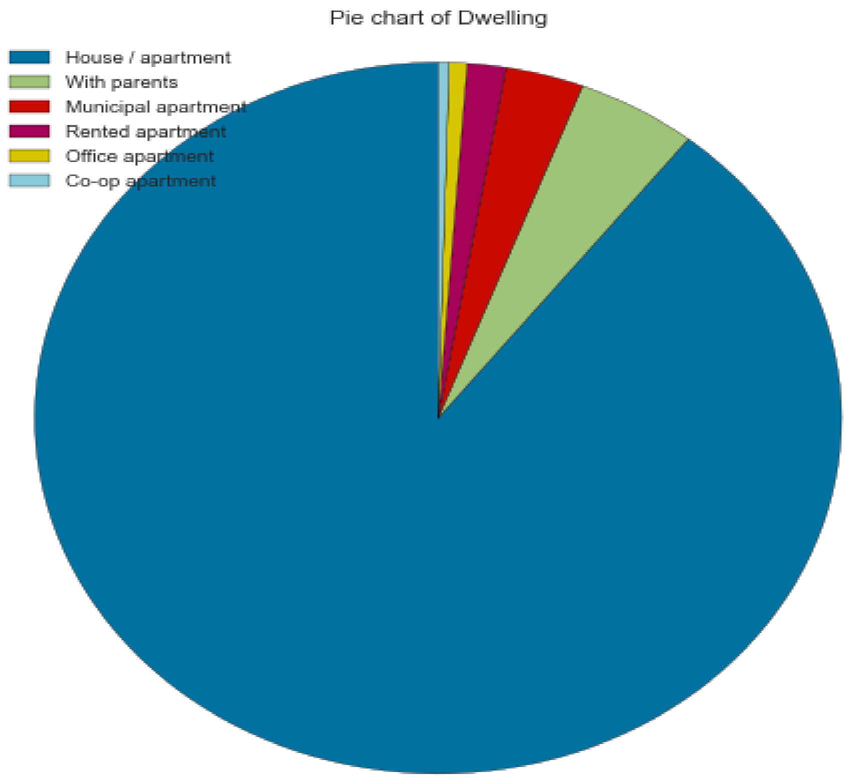
Results:



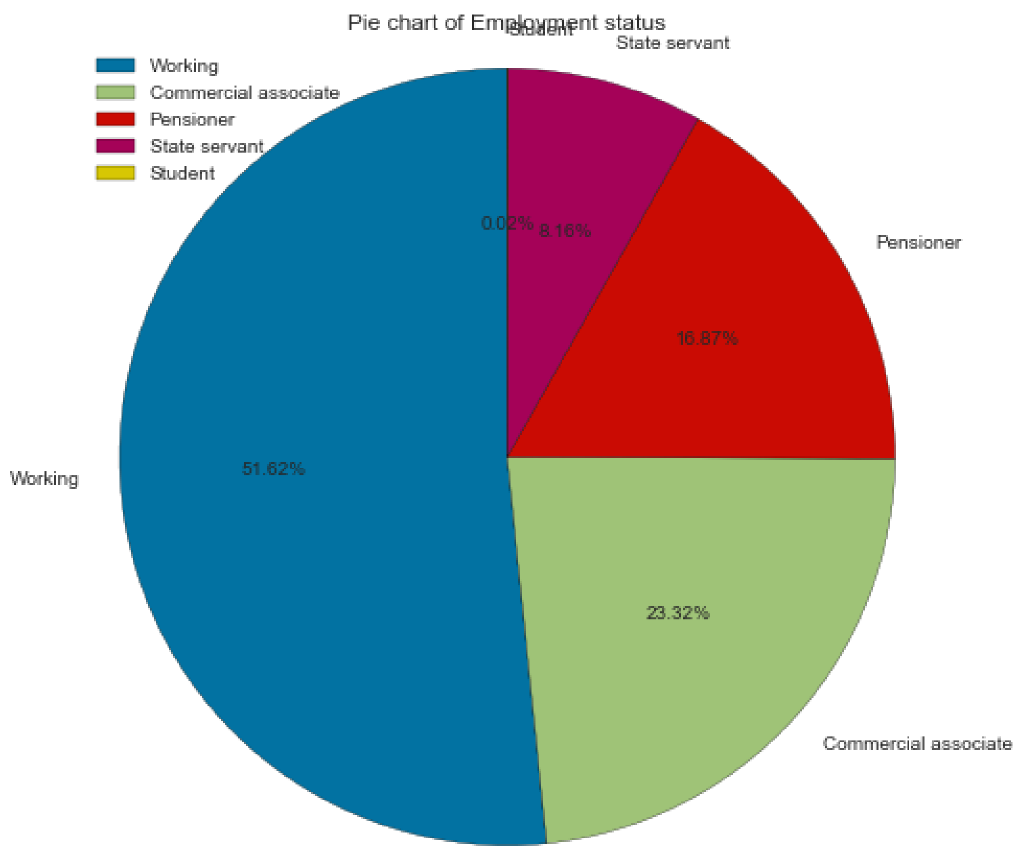
Pie chart of Gender
F
M
M
32.97%
67.03%
F



Pie chart of Marital status
Married
Single / not married
Civil marriage
Separated
Widow
Widow
Separated
Civil marriage
4.23%
5.87%
7.93%
13.25%
Single / not married
68.73%
Married

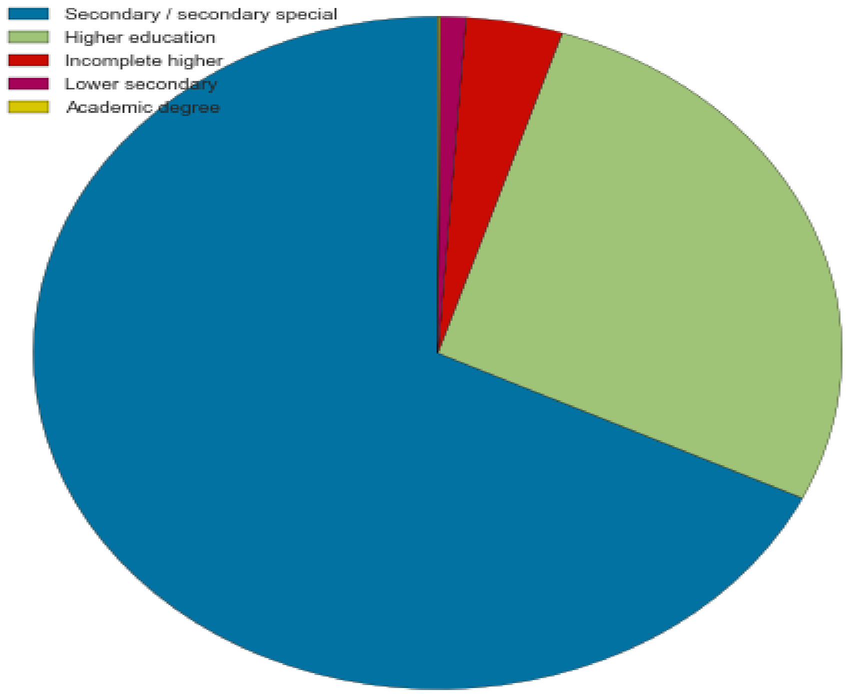


Pie chart of Dwelling
House / apartment
With parents
Municipal apartment
Rented apartment
Office apartment
Co-op apartment

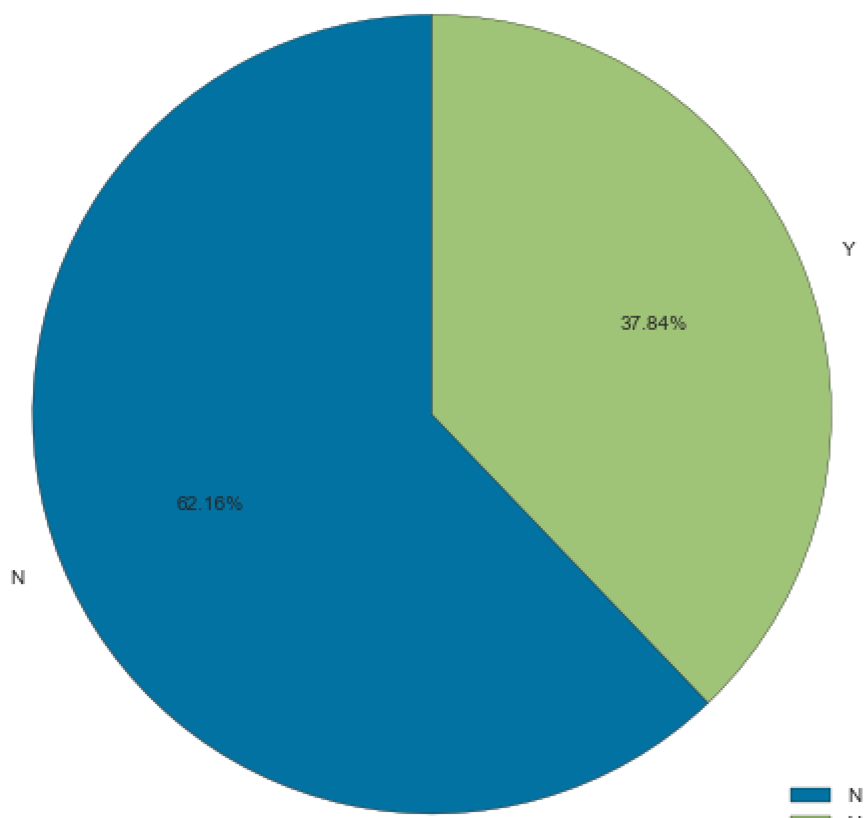


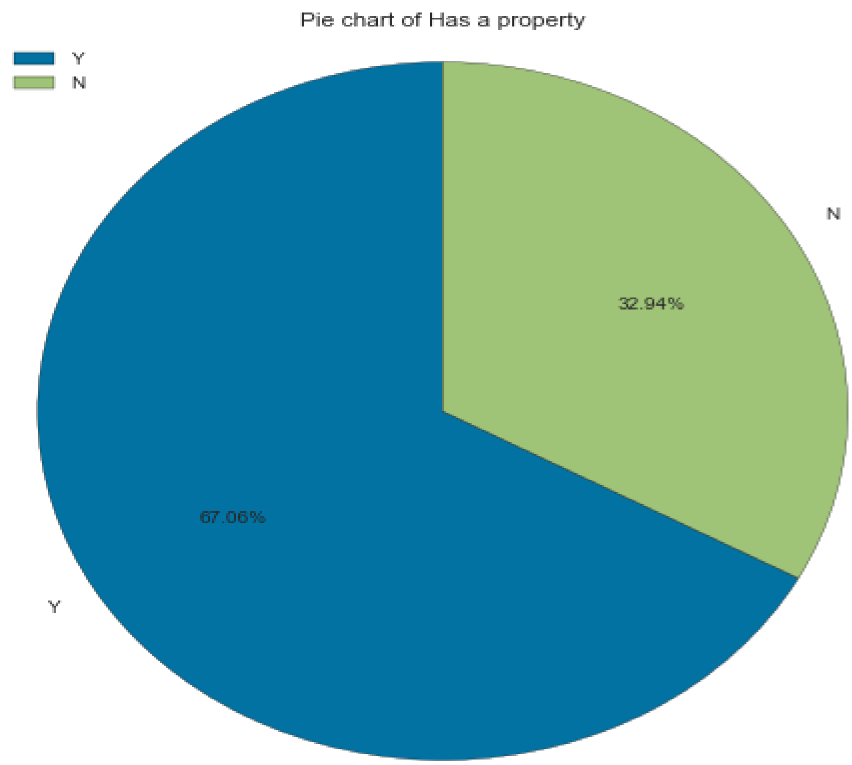
Pie chart of Employment status
Working
Commercial associate
Pensioner
State servant
Student
Student
State servant
0.02%
8.16%
Pensioner
16.87%
Working
51.62%
23.32%
Commercial associate

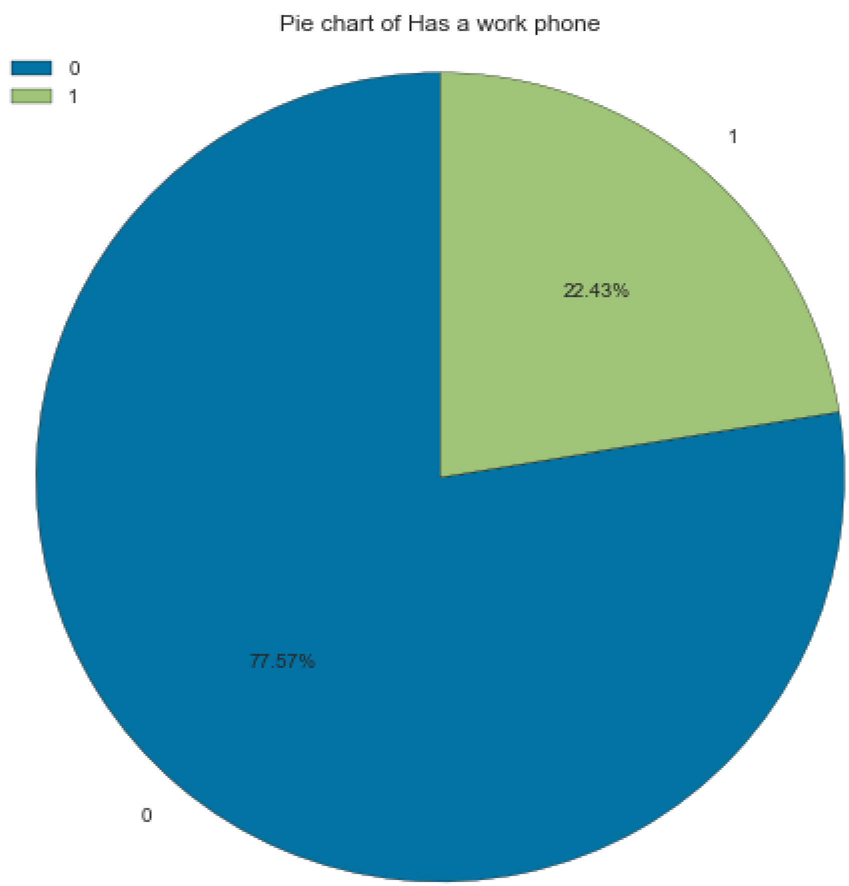
Pie chart of Education level

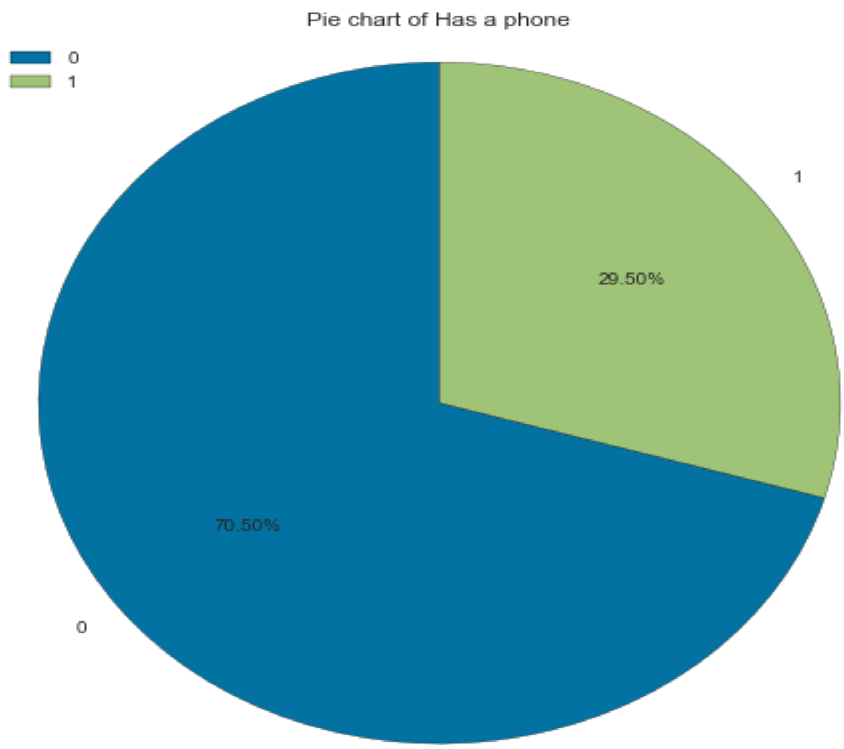


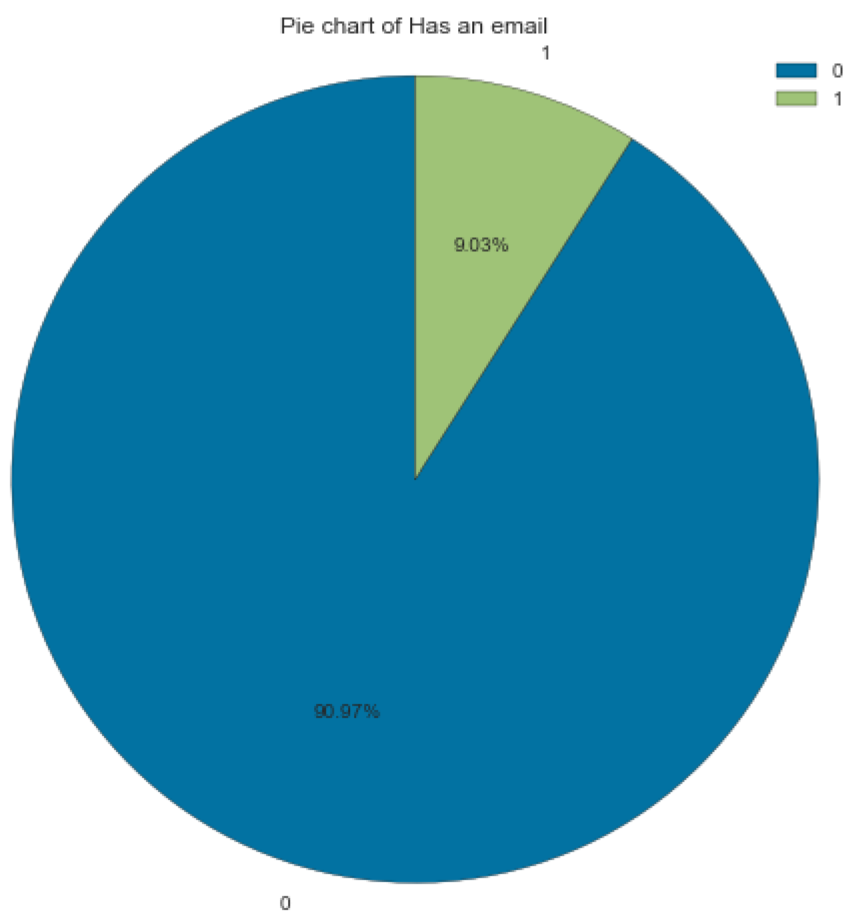
Pie chart of Has a car

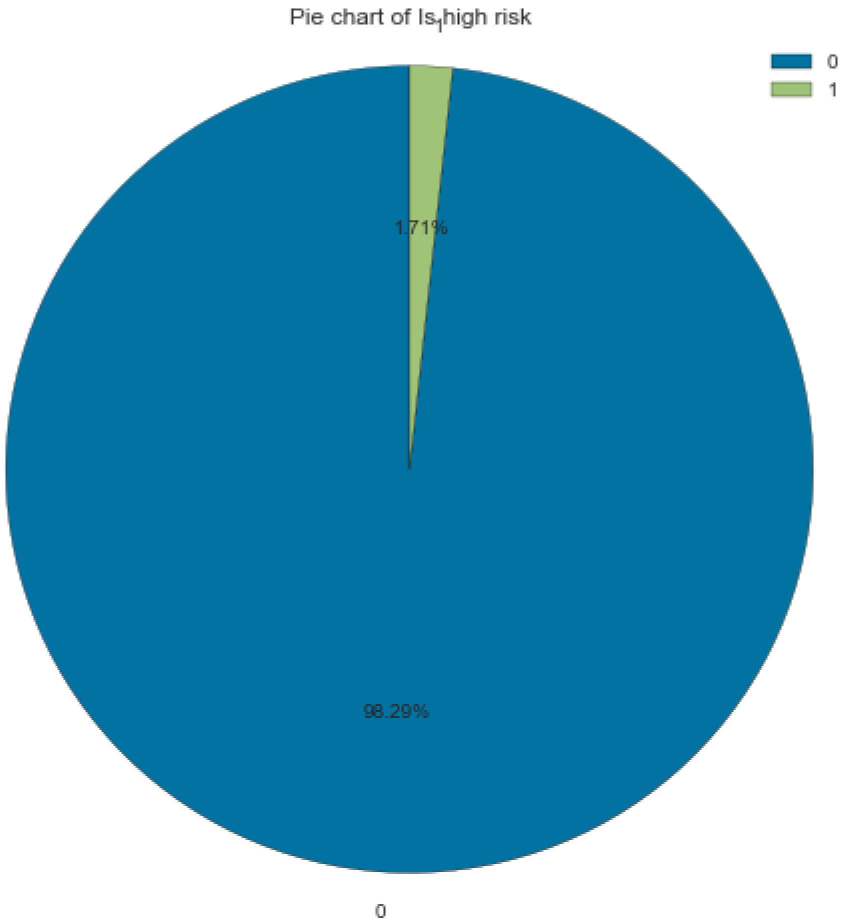


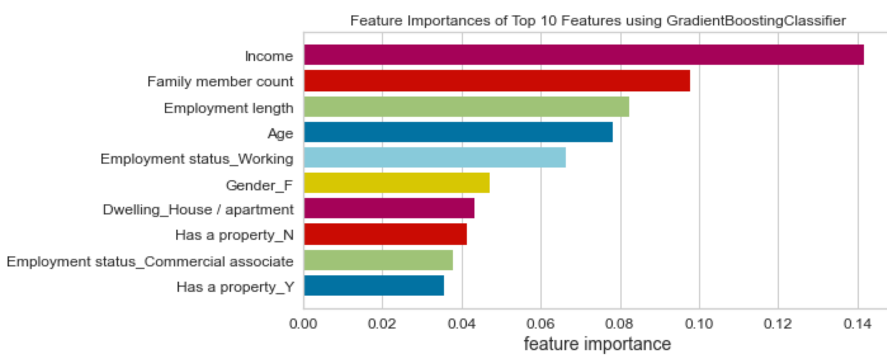

Pie chart of Has a property
Y
N
N
32.94%
67.06%
Y

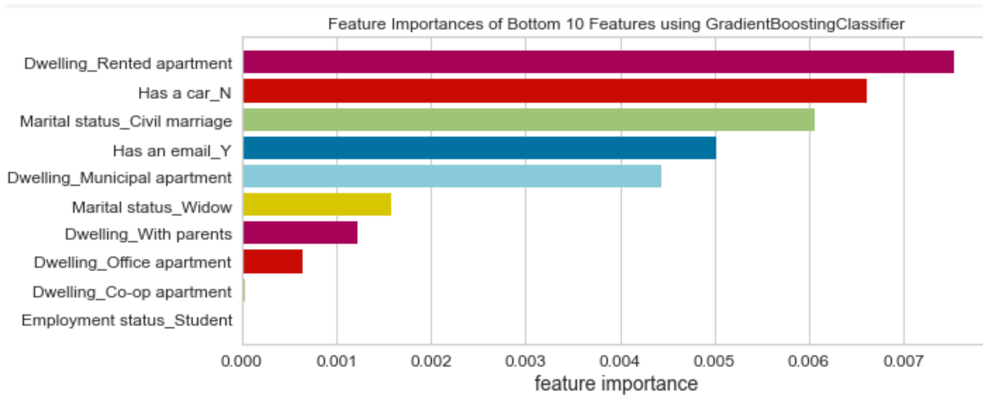

Pie chart of Has a work phone
0
1
1
22.43%
77.57%
0


Pie chart of Has a phone
0
1
1
29.50%
70.50%
0


Pie chart of Has an email
1
0
1
9.03%
90.97%
0

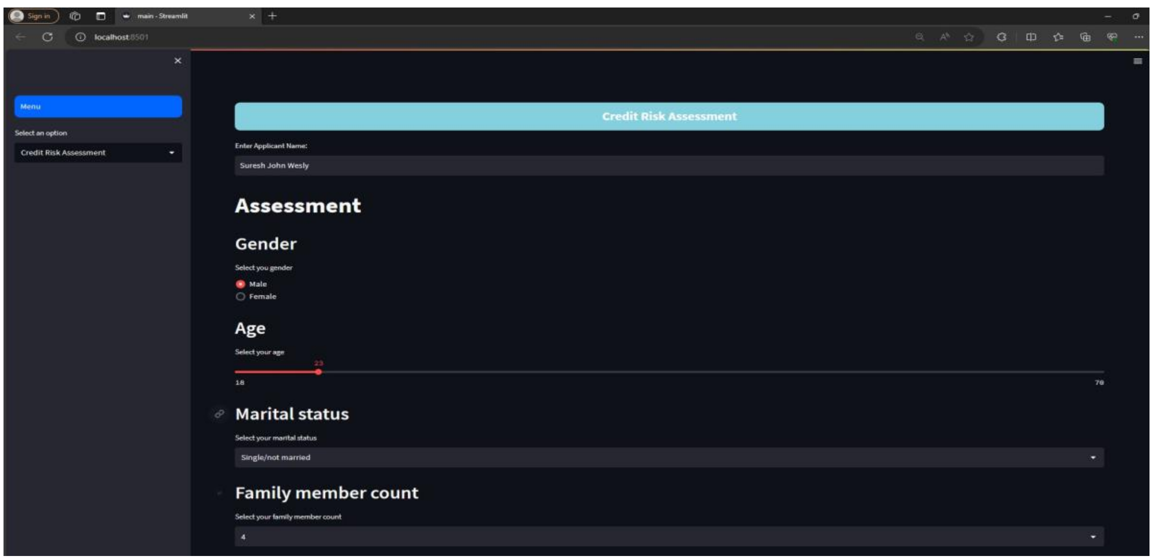

Pie chart of Is high risk
0
1
1.71%
98.29%
0

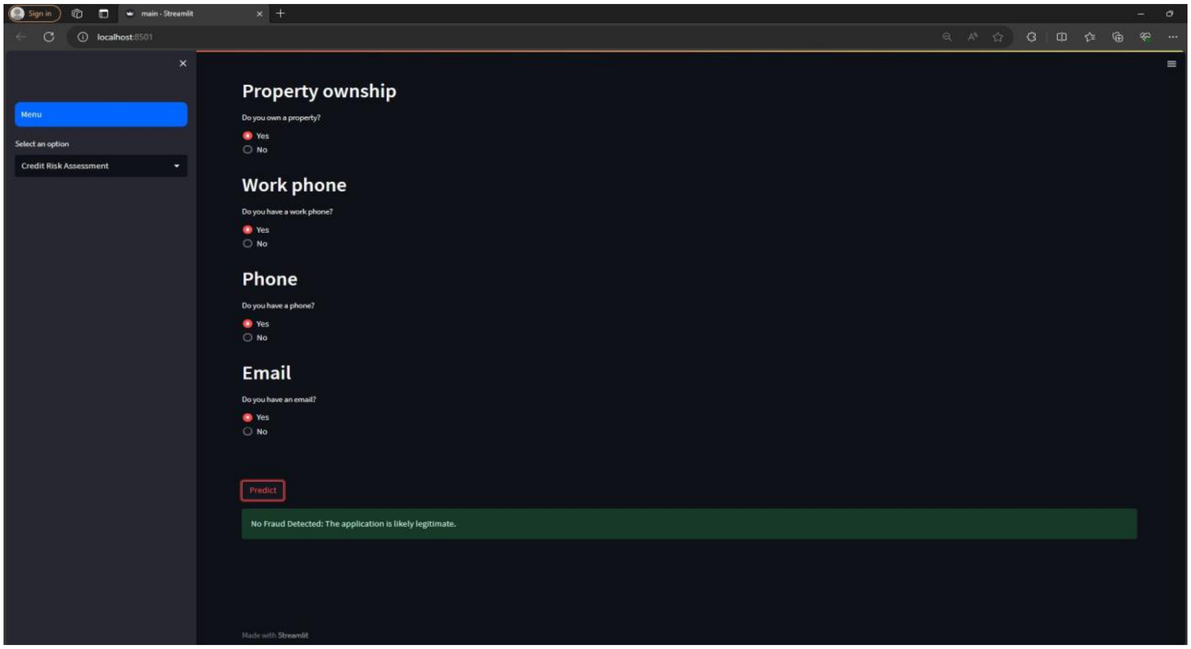

Feature Importances of Top 10 Features using GradientBoostingClassifier
Income
Family member count
Employment length
Age
Employment status_Working
Gender_F
Dwelling_House / apartment
Has a property_N
Employment status_Commercial associate
Has a property_Y
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
feature importance


Feature Importances of Bottom 10 Features using GradientBoostingClassifier
Dwelling_Rented apartment
Has a car_N
Marital status_Civil marriage
Has an email_Y
Dwelling_Municipal apartment
Marital status_Widow
Dwelling_With parents
Dwelling_Office apartment
Dwelling_Co-op apartment
Employment status_Student
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
feature importance

```
---------------------  Gradient Boosting Algorithm  ---------------------

              precision    recall  f1-score   support

           0       0.90      0.90      0.90     23272
           1       0.90      0.90      0.90     23272

    accuracy                           0.90     46544
   macro avg       0.90      0.90      0.90     46544
weighted avg       0.90      0.90      0.90     46544
```

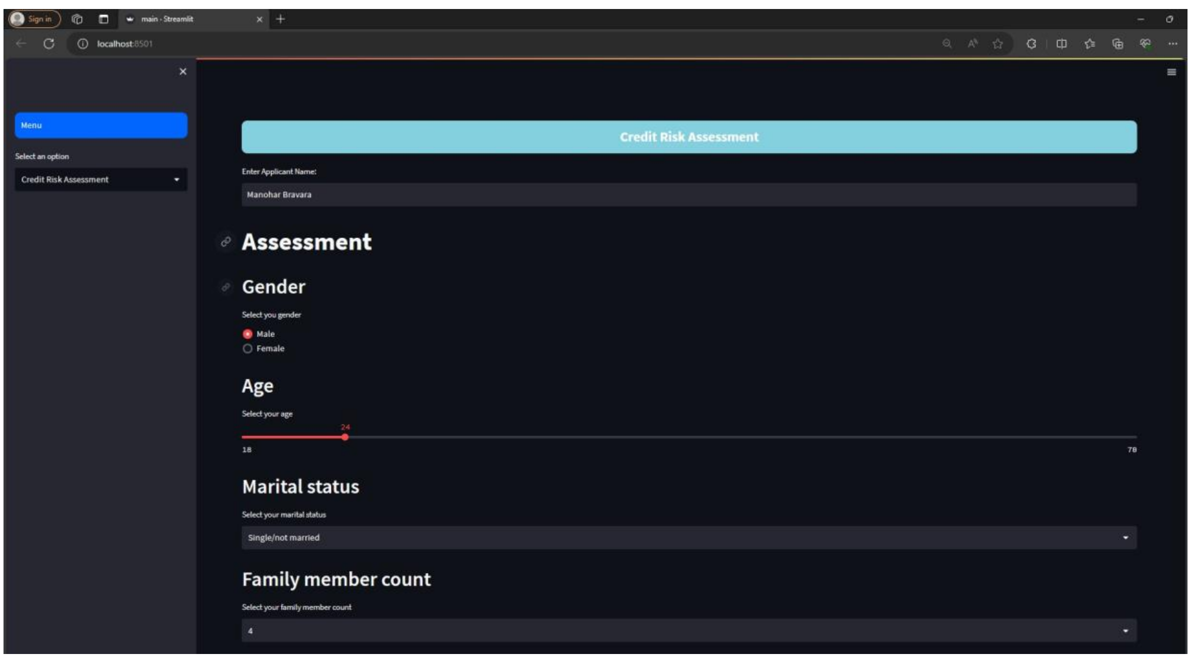
Assessment 1 :

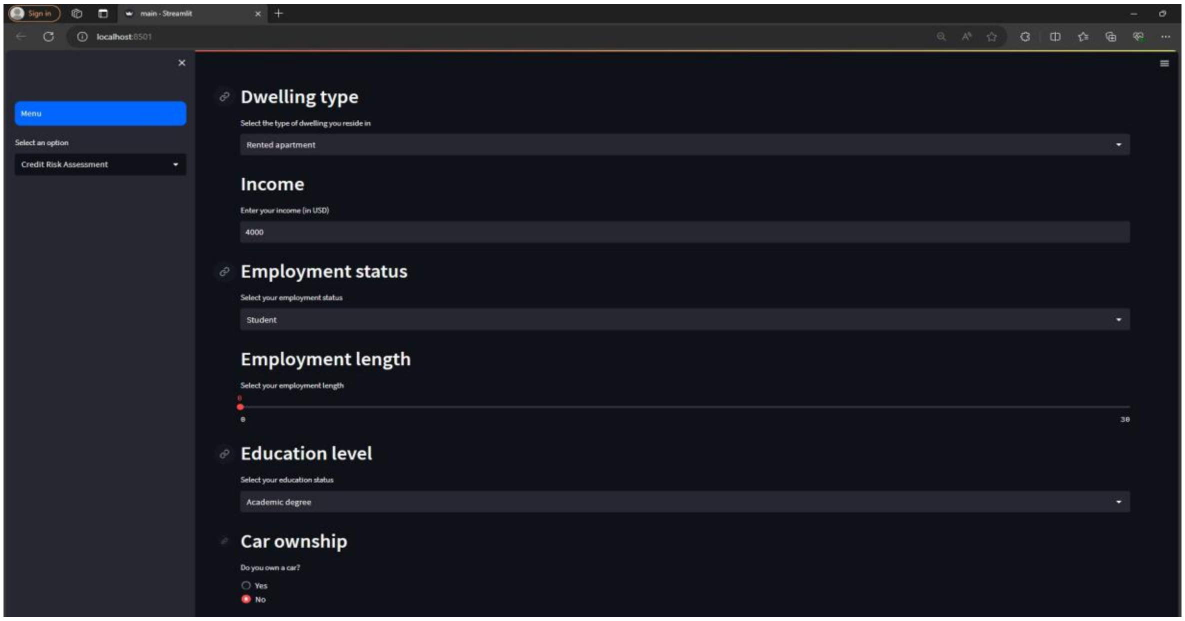


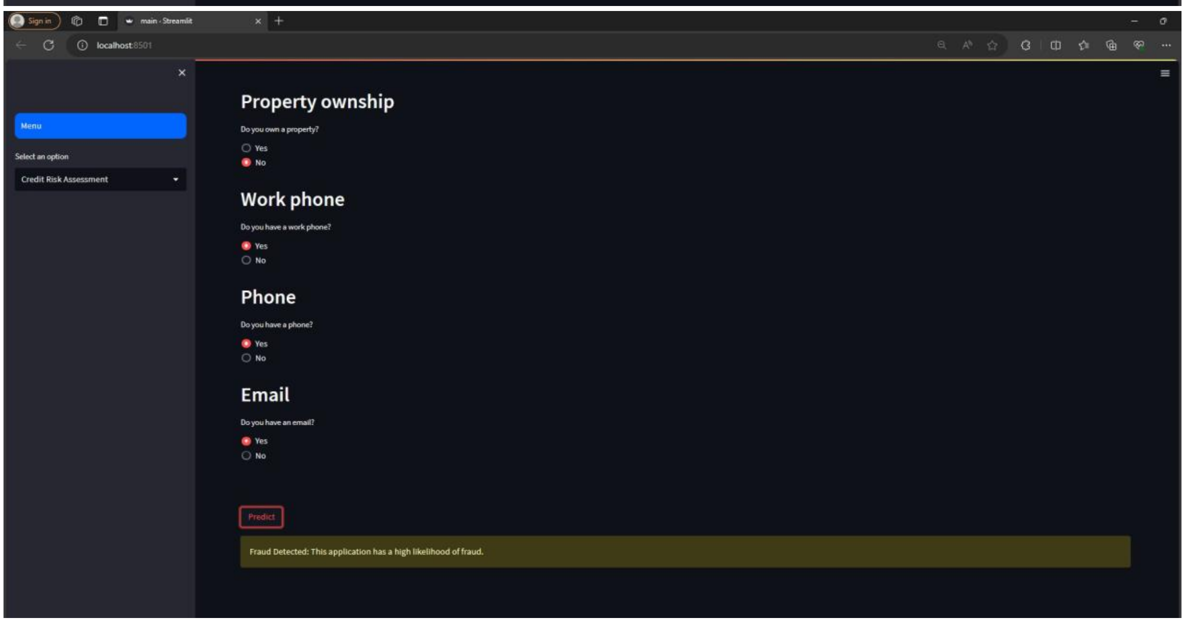


With this interface, we provided some specific set of values related to a person and clicked predict for the provided data to be tested on loaded gradient boosting model using joblib. It analyzed and returned the application as legitimate.

Assessment 2:

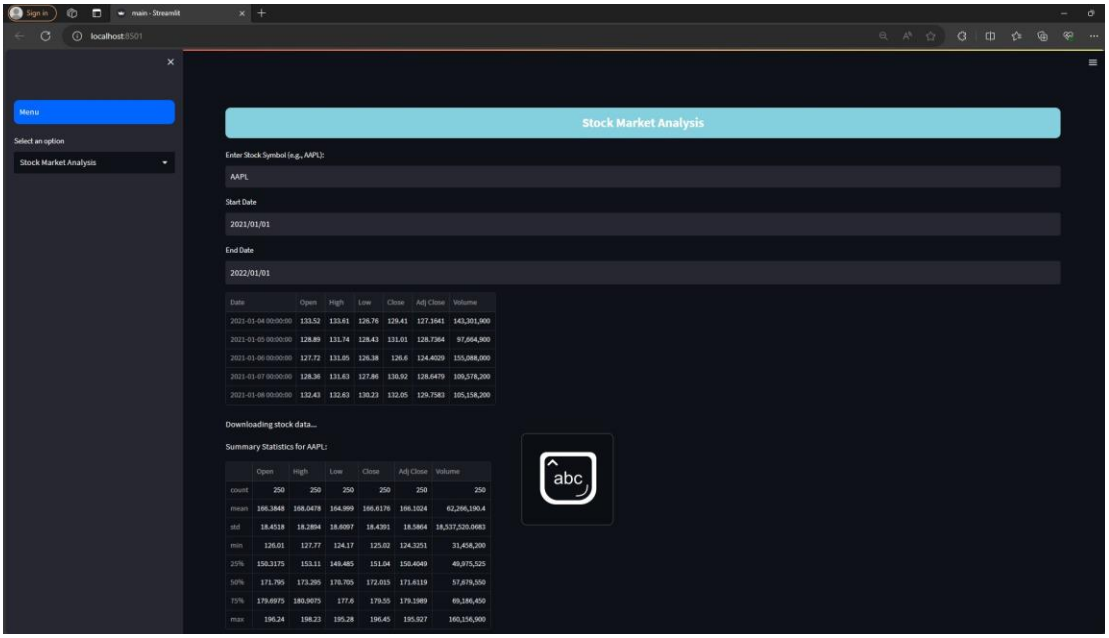




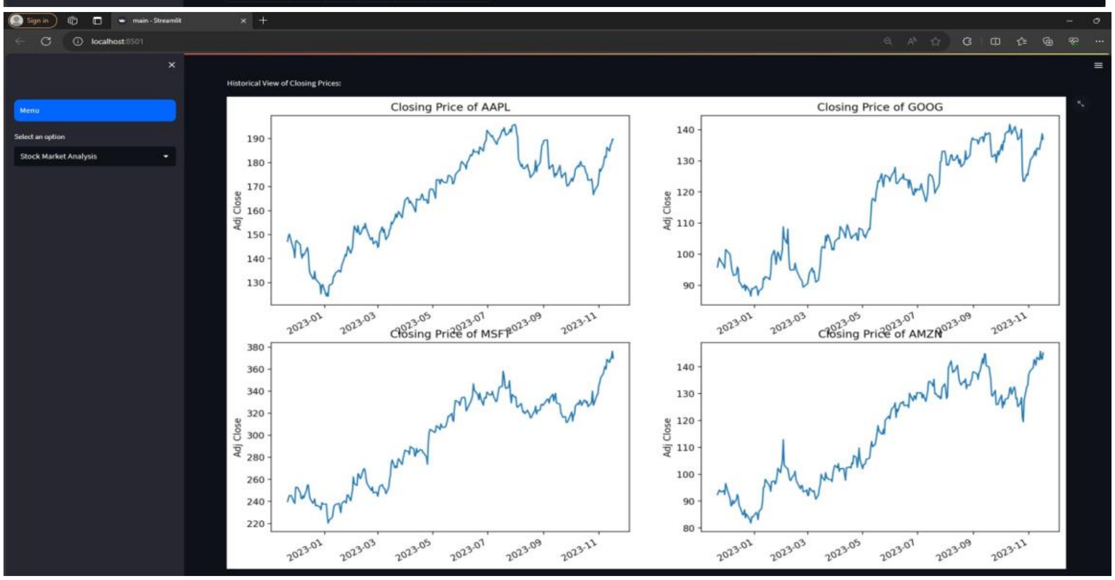


And again, we used the interface to provide another set of values, this time due to the existence of some certain set of values, it detected the application as Fraud. We intended to add some more values and extend the features, such that the result will be more meaningful and efficient.

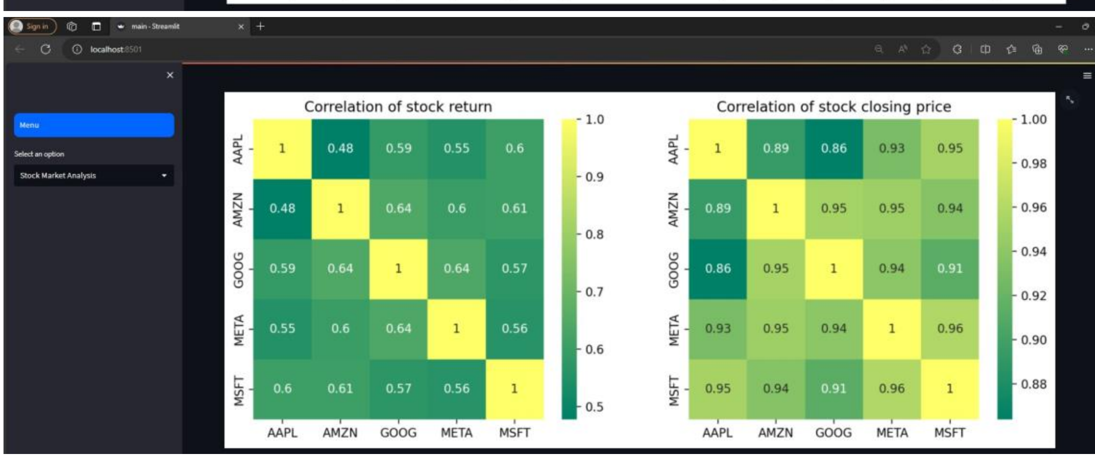
In this system, we are not training the model every time we run the system because it is so time-consuming, so we opted to save model in to local folder and whenever an application input is provided, it is provided to the model by loading using joblib module. This will take a few milliseconds to provide us with the result.



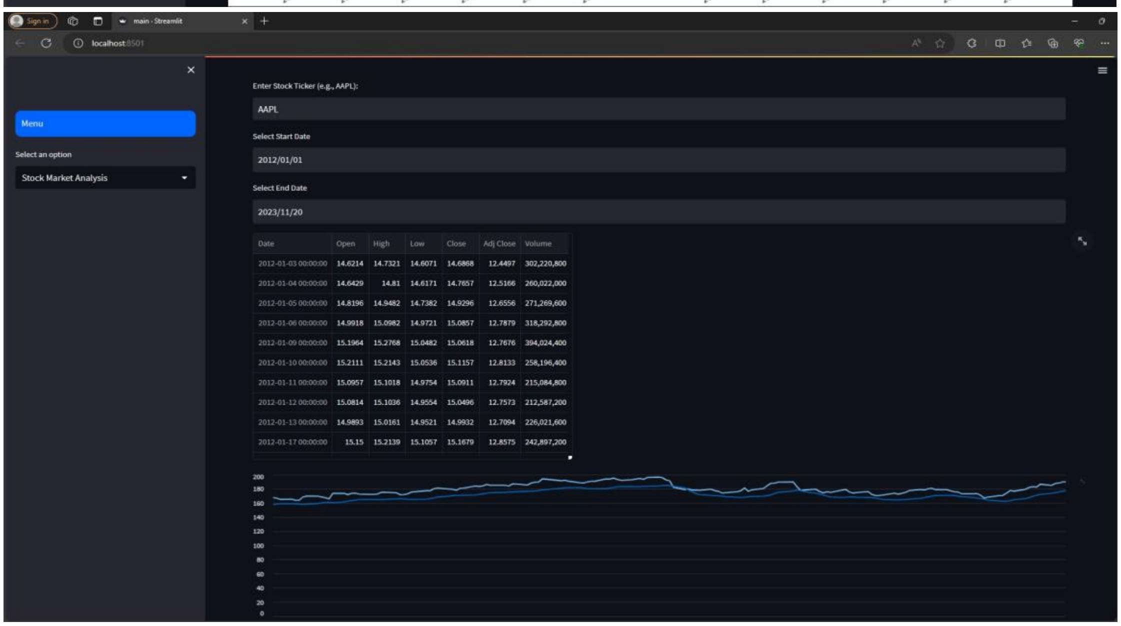
Sign in
main - Streamlit
localhost:8501
Menu
Select an option
Stock Market Analysis
Stock Market Analysis
Enter Stock Symbol (e.g., AAPL):
AAPL
Start Date
2021/01/01
End Date
2022/01/01
Date Open High Low Close Adj Close Volume
2021-01-04 00:00:00 133.52 133.61 126.76 129.41 127.1641 143,301,900
2021-01-05 00:00:00 128.89 131.74 128.43 131.01 128.7364 97,664,900
2021-01-06 00:00:00 127.72 131.05 126.38 126.6 124.4029 155,088,000
2021-01-07 00:00:00 128.36 131.63 127.86 130.92 128.6479 109,578,200
2021-01-08 00:00:00 132.43 132.63 130.23 132.05 129.7583 105,158,200
Downloading stock data...
Summary Statistics for AAPL:
Open High Low Close Adj Close Volume
count 250 250 250 250 250 250
mean 166.3848 168.0478 164.999 166.6176 166.1024 62,266,190.4
std 18.4518 18.2894 18.6097 18.4391 18.5864 18,537,520.0683
min 126.01 127.77 124.17 125.02 124.3251 31,458,200
25% 150.3175 153.11 149.485 151.04 150.4049 49,975,525
50% 171.795 173.295 170.705 172.015 171.6119 57,679,550
75% 179.6975 180.9075 177.6 179.55 179.1989 69,186,450
max 196.24 198.23 195.28 196.45 195.927 160,156,900
abc



Sign in
main - Streamlit
localhost:8501
Menu
Select an option
Stock Market Analysis
Historical View of Closing Prices:
Closing Price of AAPL
Closing Price of GOOG
Closing Price of MSFT
Closing Price of AMZN
Adj Close
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11

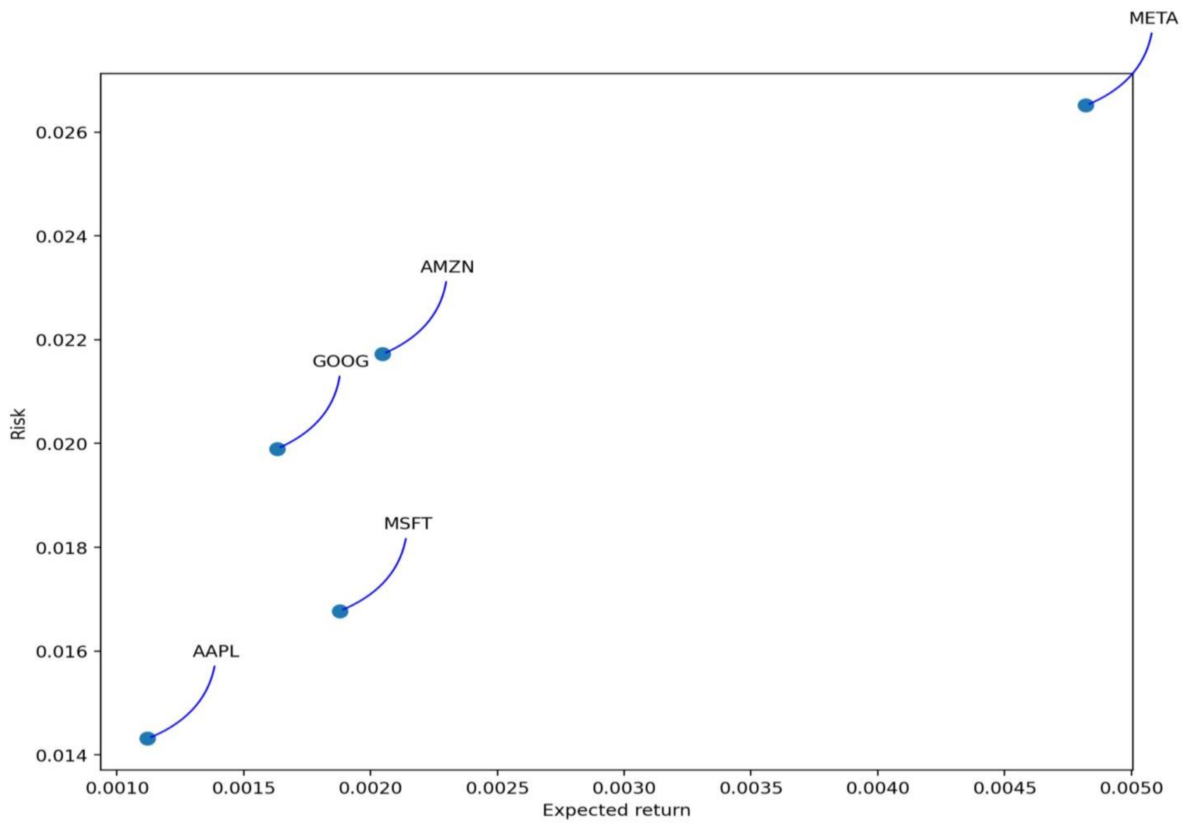


Sign in
main - Streamlit
localhost:8501
Menu
Select an option
Stock Market Analysis
Correlation of stock return
AAPL 1 0.48 0.59 0.55 0.6
AMZN 0.48 1 0.64 0.6 0.61
GOOG 0.59 0.64 1 0.64 0.57
META 0.55 0.6 0.64 1 0.56
MSFT 0.6 0.61 0.57 0.56 1
AAPL AMZN GOOG META MSFT
Correlation of stock closing price
AAPL 1 0.89 0.86 0.93 0.95
AMZN 0.89 1 0.95 0.95 0.94
GOOG 0.86 0.95 1 0.94 0.91
META 0.93 0.95 0.94 1 0.96
MSFT 0.95 0.94 0.91 0.96 1
AAPL AMZN GOOG META MSFT

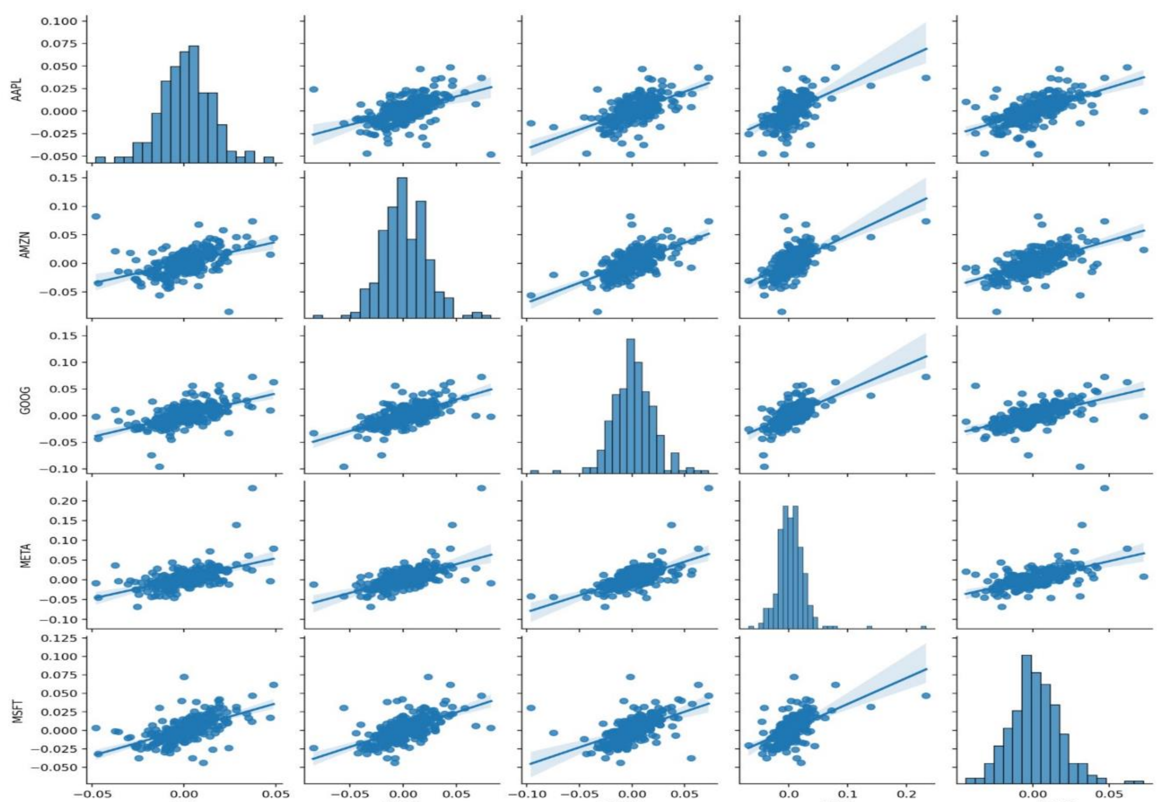


Date	Open	High	Low	Close	Adj Close	Volume
2012-01-03 00:00:00	14.6214	14.7321	14.6071	14.6868	12.4497	302,220,800
2012-01-04 00:00:00	14.6429	14.81	14.6171	14.7657	12.5166	260,022,000
2012-01-05 00:00:00	14.8196	14.9482	14.7382	14.9296	12.6556	271,269,600
2012-01-06 00:00:00	14.9918	15.0982	14.9721	15.0857	12.7879	318,292,800
2012-01-09 00:00:00	15.1964	15.2768	15.0482	15.0618	12.7676	394,024,400
2012-01-10 00:00:00	15.2111	15.2143	15.0536	15.1157	12.8133	258,196,400
2012-01-11 00:00:00	15.0957	15.1018	14.9754	15.0911	12.7924	215,084,800
2012-01-12 00:00:00	15.0814	15.1036	14.9554	15.0496	12.7573	212,587,200
2012-01-13 00:00:00	14.9893	15.0161	14.9521	14.9932	12.7094	226,021,600
2012-01-17 00:00:00	15.15	15.2139	15.1057	15.1679	12.8575	242,897,200

Date	Close	Predictions
2023-11-06 00:00:00	179.23	174.0382
2023-11-07 00:00:00	181.82	175.5606
2023-11-08 00:00:00	182.89	177.3058
2023-11-09 00:00:00	182.41	179.0212
2023-11-10 00:00:00	186.4	180.3783
2023-11-13 00:00:00	184.8	181.9396
2023-11-14 00:00:00	187.44	183.0693
2023-11-15 00:00:00	188.01	184.2266
2023-11-16 00:00:00	189.71	185.263
2023-11-17 00:00:00	189.69	186.3272



META
AMZN
GOOG
MSFT
AAPL
Risk
Expected return



AAPL
AMZN
GOOG
META
MSFT

One of the fundamental methods to assess risk involves analyzing daily percentage returns and comparing the anticipated return with the standard deviation of these daily returns. This approach allows us to gauge the level of risk associated with an investment by considering both the expected return and the degree of variability in daily performance. It's a basic yet crucial way to evaluate and understand the potential risks involved in financial endeavors. We can observe from the plots that return and risk from META is comparatively higher than the other stocks.

The line plot in the interface provides a comparison between actual stock price and predicted stock price. We look forward to train this model more vigorously and make it more accurate.

Conclusion:

The Credit Risk Assessment System and Stock Market Analysis System stand as cutting-edge financial technology solutions, poised to empower finance sector individuals and organizations. Through the utilization of advanced machine learning, robust data analysis techniques, and a user-friendly interface, these systems provide valuable insights for credit evaluations and stock market analysis.

The Credit Risk Assessment System revolutionizes the credit evaluation process by leveraging historical borrower data and predictive models. Tailored for banks and lending institutions, it facilitates precise loan decisions, diminishes risks, and upholds compliance with fair lending standards. Conversely, the Stock Market Analysis System equips investors with a comprehensive toolkit, employing data analytics, time-series forecasting models, and sentiment analysis.

The outlined development timeline underscores a methodical approach, emphasizing extensive research, core functionality development, feature integration, and ongoing refinement in collaboration with instructors. Noteworthy features, including credit scoring models, fraud detection, stock price prediction, and trend analysis, collectively contribute to the systems' value proposition within the finance sector.

Future Work:

1. Enhanced Machine Learning Models: Continuous advancement and refinement of machine learning models for both credit risk assessment and stock market analysis to ensure sustained effectiveness.

2. Alternative Data Integration: Exploration of additional alternative data sources, such as satellite imagery or social media data, to augment stock market analysis with more diverse insights.

3. Scalability: Assurance of system scalability to accommodate increasing loan applications and real-time stock market data.

4. User Interface Enhancement: Ongoing refinement of the user interface for heightened intuitiveness, responsiveness, and visual appeal, ensuring a seamless user experience.

5. Algorithmic Trading Strategies: Further development and testing of algorithmic trading strategies within the Stock Market Analysis System to comprehensively assess profitability.

6. Continuous Collaboration and Refinement: Persistent collaboration with instructors and teaching assistants for feedback and ongoing refinement, ensuring the systems remain current and aligned with evolving industry standards.

7. Cloud Deployment: The deployment in to cloud would provide more resources and computation power and also it can be accessed from anywhere.

Future endeavors should concentrate on refining algorithms, exploring novel data sources, bolstering scalability, and embracing continuous improvement based on user feedback and the dynamic financial landscape.

References:

[1] https://www.upgrad.com/blog/top-artificial-intelligence-project-ideas-topics-forbeginners/

[2] https://arxiv.org/pdf/2212.12717.pdf

[3] https://dx.doi.org/10.2139/ssrn.4441658

[4] https://doi.org/10.1109/MLBDBI54094.2021.00034