

# **INTRODUCTION**

Welcome to our state-of-the-art AutoML platform leveraging the incredible capabilities of PyCaret! Our groundbreaking application redefines the landscape of machine learning by offering a seamless and comprehensive solution for all levels of expertise. Whether you're a seasoned data scientist or a newcomer eager to delve into the world of predictive modeling, our user-friendly interface powered by PyCaret simplifies the complex task of model creation, making it accessible and efficient for everyone.

PyCaret's robust suite of tools integrates seamlessly into our application, providing access to an extensive array of machine learning algorithms, automated feature engineering, hyperparameter tuning, model selection, and evaluation—all within a single unified environment. This means that even without extensive coding knowledge, you can harness the full potential of PyCaret's capabilities to generate highly accurate models tailored to your specific needs.

Our AutoML app is designed to streamline your workflow, saving you valuable time and effort by automating the repetitive and resource-intensive tasks involved in the model development process. With PyCaret at its core, the platform empowers you to explore diverse modeling techniques, compare multiple algorithms, and optimize model performance without the steep learning curve typically associated with machine learning.

# **LITERATURE SURVEY**

## **Introduction**

Automated Machine Learning (AutoML) has emerged as a transformative approach to democratize the process of developing machine learning models, enabling both experts and novices to create powerful predictive models without deep expertise in data science or programming. PyCaret, an advanced machine learning library in Python, has gained traction for its ability to simplify and expedite the model-building process. This literature survey aims to explore the evolution, advancements, and applications of AutoML, specifically focusing on the utilization of PyCaret.

## **Evolution of AutoML**

AutoML research has evolved rapidly, driven by the necessity to democratize machine learning and alleviate the barriers posed by technical complexities. Early AutoML frameworks focused on automating model selection and hyperparameter tuning. Recent advancements, such as PyCaret, have expanded these capabilities by incorporating automated feature engineering, pipeline optimization, and model explanation, thereby providing end-to-end automation for the entire machine learning workflow.

## **PyCaret: A Comprehensive AutoML Toolkit**

PyCaret has garnered attention as a versatile and user-friendly AutoML library due to its rich suite of functionalities. It offers a wide range of algorithms and streamlined workflows that automate complex tasks like feature selection, hyperparameter optimization, model comparison, and deployment. PyCaret's intuitive interface and comprehensive documentation make it accessible to both beginners and experienced practitioners, enabling rapid experimentation and model iteration.

## **Applications of PyCaret in Various Domains**

The application of PyCaret spans across diverse domains, demonstrating its flexibility and utility in solving real-world problems. In healthcare, PyCaret has

been employed for disease prediction, patient outcome analysis, and medical image classification. In finance, it aids in fraud detection, risk assessment, and stock market forecasting. Additionally, PyCaret finds use in marketing for customer segmentation, churn prediction, and recommendation systems, among others.

### Challenges and Future Directions

While PyCaret significantly simplifies the AutoML process, challenges persist, such as interpretability of complex models generated by automated systems, handling imbalanced datasets, and scalability issues. Future research directions may include enhancing interpretability, expanding support for specialized domains, addressing ethical considerations, and improving model robustness and generalizability.

### Conclusion

PyCaret stands at the forefront of AutoML tools, democratizing machine learning by providing a user-friendly interface that automates the end-to-end model development process. Its wide-ranging applications across various domains showcase its adaptability and potential. As research continues to evolve, PyCaret is expected to play a pivotal role in accelerating the adoption of machine learning across industries, making predictive analytics more accessible and impactful.

This literature survey aims to provide an overview of the evolution, applications, challenges, and future prospects of using PyCaret within the realm of Automated Machine Learning.

# **PROBLEM STATEMENT**

## **Addressing Complexity and Accessibility in Machine Learning through an Automated Machine Learning (AutoML) Application:**

Machine learning (ML) has become a cornerstone of modern data-driven decision-making across industries. However, its widespread adoption faces significant challenges, primarily centered around the complexities inherent in model development and the expertise required to navigate this intricate landscape effectively.

The conventional process of developing machine learning models demands a high level of expertise in data preprocessing, algorithm selection, hyperparameter tuning, and model evaluation. This complexity poses a barrier, limiting the accessibility of machine learning to only those with specialized skills, thereby hindering its potential impact across diverse sectors.

Moreover, the shortage of skilled data scientists further exacerbates this issue, impeding organizations from harnessing the insights hidden within their data effectively. As a result, crucial opportunities for data-driven decision-making, predictive analytics, and innovation remain underutilized.

Therefore, the need for an Automated Machine Learning (AutoML) application becomes evident. Such a tool would democratize the process of model development, enabling users with varying degrees of expertise to harness the power of machine learning without extensive knowledge of algorithms, coding, or data science intricacies.

An AutoML application would streamline the entire machine learning workflow, automating labor-intensive tasks such as feature selection, hyperparameter tuning, model selection, and evaluation. By providing an intuitive and user-friendly interface, it would empower domain experts, business analysts, and novices alike to leverage the predictive potential of their data, thereby fostering a data-driven culture within organizations.

# **PROPOSED SOLUTION**

Our solution entails the creation of a comprehensive and user-friendly AutoML application that leverages the capabilities of advanced machine learning libraries, such as PyCaret, to streamline the entire model development process.

## **Key Features:**

1. **Intuitive Interface:** Designing a user-friendly interface that requires minimal coding knowledge, allowing users with diverse backgrounds to navigate effortlessly through the machine learning workflow.
2. **Automated Workflows:** Implementing automation for critical tasks like data preprocessing, feature selection, algorithm selection, hyperparameter tuning and model evaluation minimizing manual intervention.
3. **Algorithm Diversity:** Incorporating a broad spectrum of machine learning algorithms and techniques within the application, catering to various use cases and enabling users to explore and compare different models effortlessly.
4. **Model Interpretability:** Prioritizing model interpretability by integrating tools and techniques that explain model predictions, thereby enhancing transparency and understanding.
5. **Scalability and Customizability:** Ensuring the application's scalability to handle large datasets while allowing for customization to accommodate domain-specific requirements and advanced user preferences.

6. Educational Resources: Providing comprehensive documentation, tutorials, and educational materials to facilitate learning and empower users to make informed decisions throughout the modeling process.

#### Expected Outcome:

The proposed AutoML application aims to democratize machine learning by lowering the entry barrier, enabling users across disciplines to harness the power of predictive analytics without extensive expertise. By simplifying and automating the model development process, the application empowers users to extract valuable insights from their data efficiently and effectively.

# EXPERIMENTAL SETUP

Source Code:

## 1. Importing Dependencies :

```
import streamlit as st
import pandas as pd
import os

from ydata_profiling import ProfileReport
from streamlit_pandas_profiling import st_profile_report

from pycaret.classification import setup, compare_models, pull, save_model, load_model, predict_model
```

## 2. Creating First Page in Streamlit:

Streamlit enables us to easily build the frontend for deploying machine learning apps. In the following code snippet we build the front page of the app.

```
with st.sidebar:
    st.image('icon.png')
    st.title('Automated ML')
    choice = st.radio("Navigation", ['Upload', 'Data Analysis', 'Model Creation', 'Download Model', 'Test Model Predictions'])
    st.info('An AI-driven AutoML web app that automates the process of building, training, and deploying machine learning models, making data science accessible to everyone.')
```

Next we need to setup a way to upload a dataset for EDA, target selection and model building.

We can do so easily using the file uploader method in streamlit library.

### 3. Choice #1: Uploading Dataset:

```
if os.path.exists('original_data.csv'):
    df = pd.read_csv('original_data.csv', index_col=None)
if choice == 'Upload':
    file = st.file_uploader('Upload your Data here')
    if file:
        df = pd.read_csv(file, index_col=None)
        df.to_csv('original_data.csv', index=None)
        st.dataframe(df)
```

The uploaded data can be analysed for trends and patterns so that we can check for missing values and perform operations scaling, tokenization etc.

However even if we don't perform this step manually pycaret will automatically perform some cleaning and feature engineering on the data.

### 4. Choice #2: Exploratory Data Analysis:

```
if choice == 'Data Analysis':
    st.title('Exploratory Data Analysis')
    report = ProfileReport(df)
    st_profile_report(report)
```



### 5. Choice #3: Model Creation and training:

After preprocessing we need to build and train a relevant model for the given problem. We can leave the choosing of the best model up to pycaret, however we need to select the target variable for the given problem.

```
if choice == 'Model Creation':  
    st.title('Generation of ML model')  
    target = st.selectbox('Select Target Parameter', df.columns)  
    if st.button('Train Model'):  
        setup(df, target=target)  
        setup_df = pull()  
        st.info('Model Experimentation Settings')  
        st.dataframe(setup_df)  
        best_model = compare_models()  
        compare_df = pull()  
        st.info('Generated Model')  
        st.dataframe(compare_df)  
        best_model  
        save_model(best_model, 'best_model')
```

### 6. Choice #4: Downloading the best generated model:

After generating the best model we can setup a download path for the user to obtain the model and use it in their respective program.

```
if choice == 'Download Model':  
    st.info('You Can Download the generated model from here')  
    with open('best_model.pkl', 'rb') as f:  
        st.download_button('Download the Model', f, 'trained_model.pkl')
```

## 7. Choice #5: Testing the Generated Model:

We can test the generated model before implementing anywhere by simply uploading the test dataset.

The app will return a new dataset with the prediction labels along with accuracy scores for each prediction.

```
if choice == 'Test Model Predictions':  
    st.title('Model Testing')  
    st.info('Upload your test dataset')  
    test = st.file_uploader('Upload Test csv file')  
    df2 = pd.read_csv(test)  
    if test:  
        st.info('This is your test dataset')  
        st.dataframe(df2)  
    pipeline = load_model('trained_model')  
    st.info('This is the dataframe with the prediction labels')  
    st.dataframe(predict_model(pipeline, df2))
```

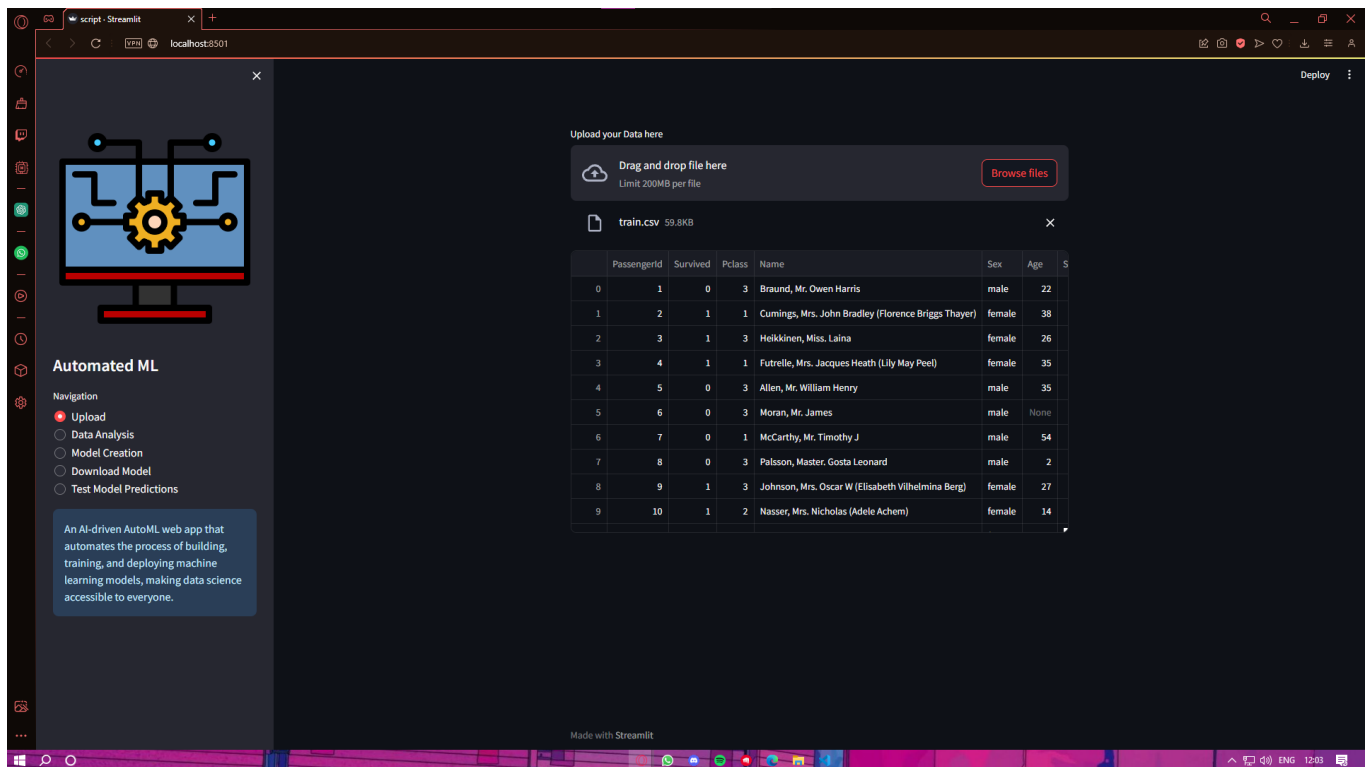
Once the model is tested and generated the user can implement into any other scripts using the following snippet:

```
from pycaret.classification import load_model, predict_model  
import numpy as np  
import pandas as pd  
  
test = pd.read_csv('test.csv')  
  
pipeline = load_model('trained_model')  
  
predict_model(pipeline, test)
```

# FINAL WEBAPP SHOWCASE

## Landing Page:

1. Here the users can upload the training dataset using the file uploader.
2. After uploading they can move on to the next steps using the menu selection on the sidebar.

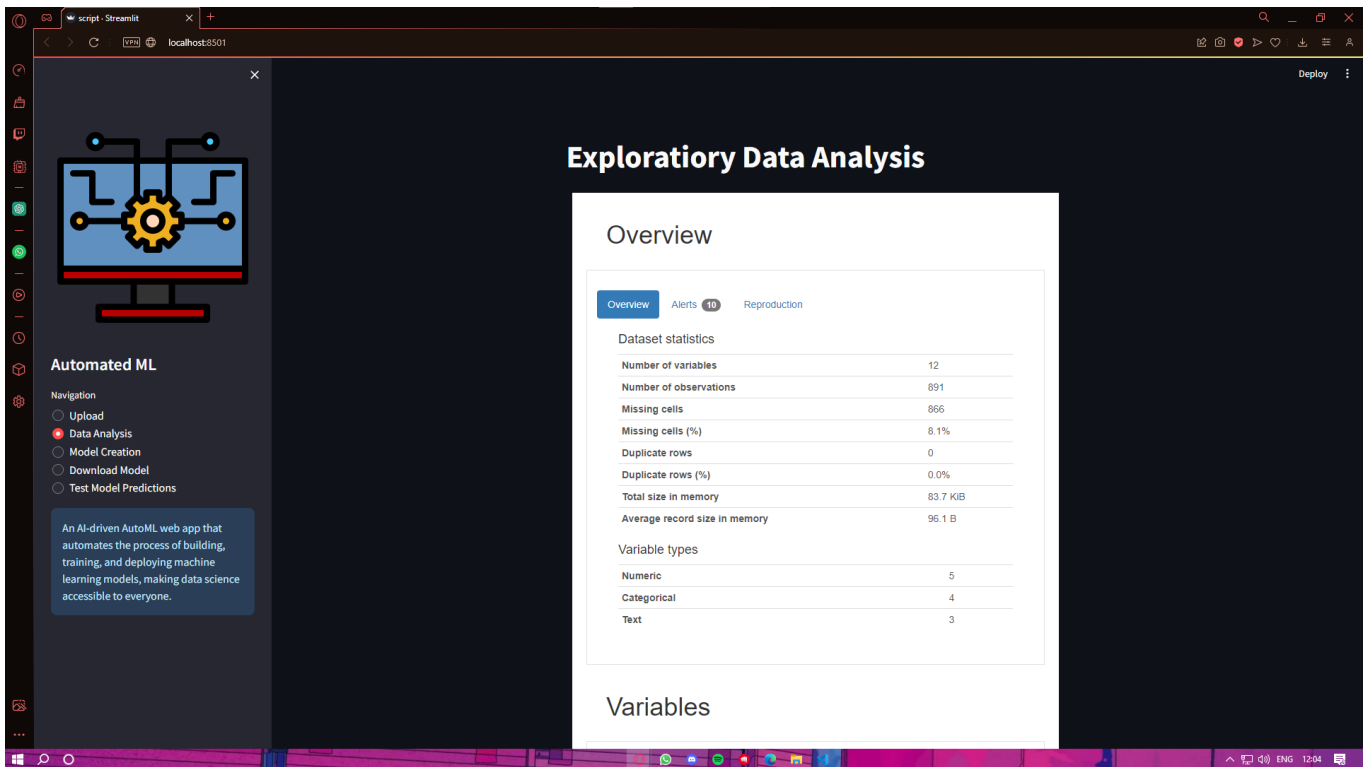


## Exploratory Data Analysis page:

In this tab EDA is performed on the uploaded data to display all relevant information needed for data cleaning and preprocessing.

Following are some examples of the displayed data:(contd)

1.



**Automated ML**

Navigation

- ☐ Upload
- ☒ Data Analysis
- ☐ Model Creation
- ☐ Download Model
- ☐ Test Model Predictions

An AI-driven AutoML web app that automates the process of building, training, and deploying machine learning models, making data science accessible to everyone.

## Exploratory Data Analysis

### Overview

Overview Alerts 10 Reproduction

**Dataset statistics**

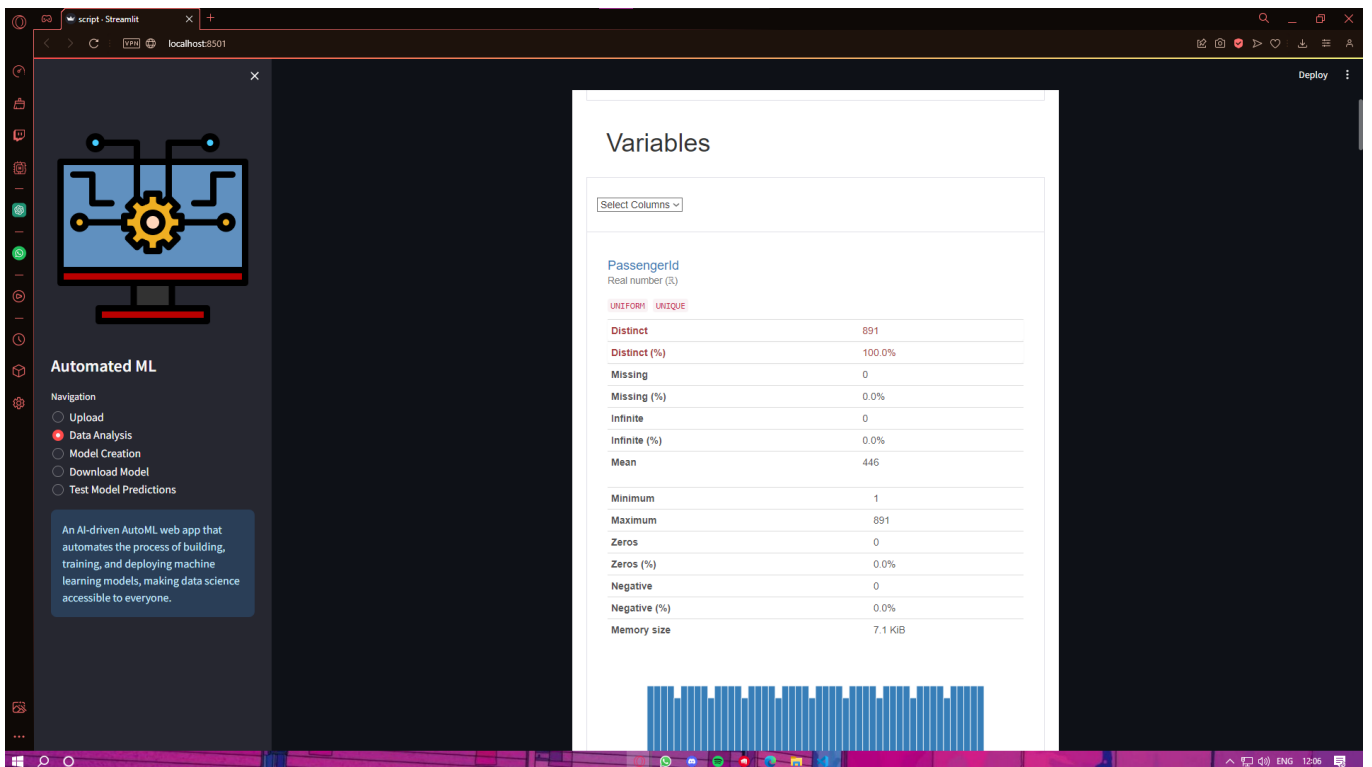
Number of variables	12
Number of observations	891
Missing cells	866
Missing cells (%)	8.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	83.7 KiB
Average record size in memory	96.1 B

**Variable types**

Numeric	5
Categorical	4
Text	3

### Variables

2.



## Variables

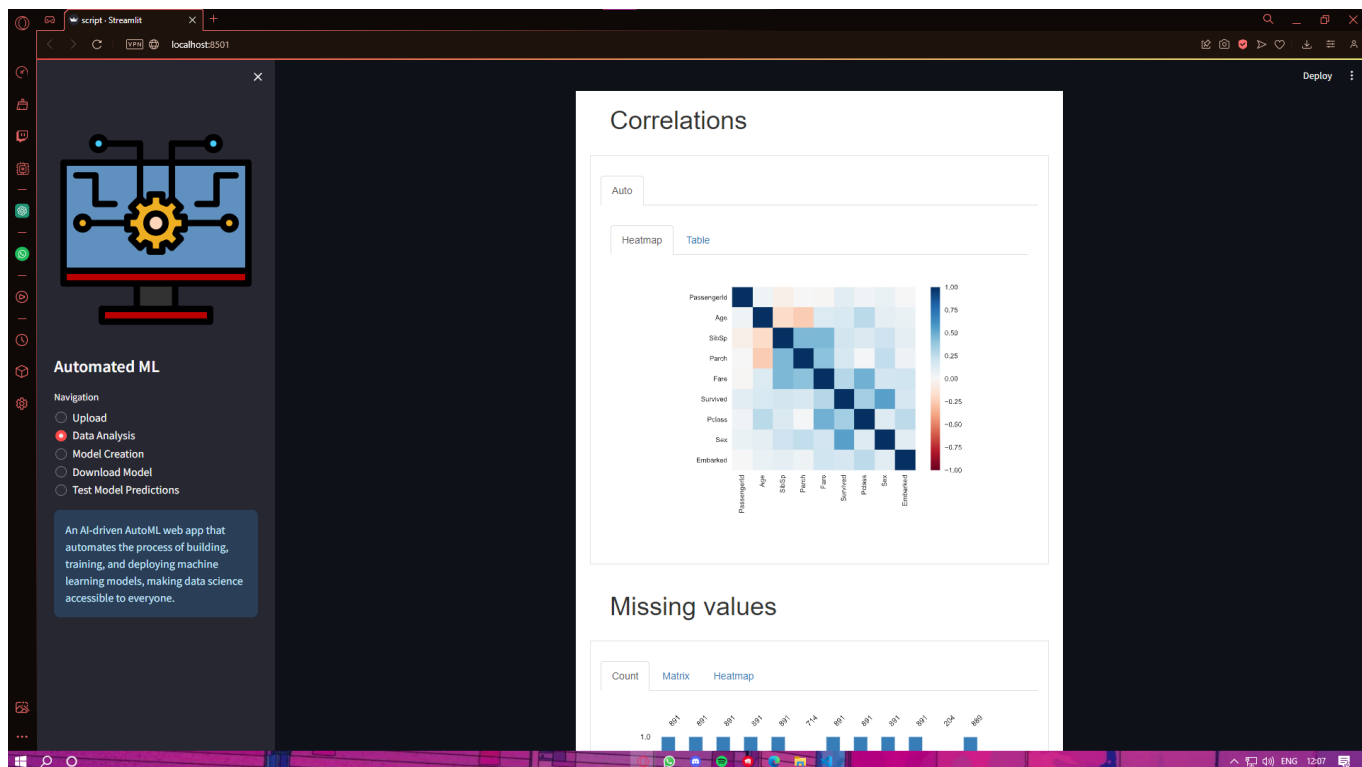
Select Columns ▾

**PassengerId**  
Real number (3)

UNIFORM UNIQUE

Distinct	891
Distinct (%)	100.0%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Mean	446
Minimum	1
Maximum	891
Zeros	0
Zeros (%)	0.0%
Negative	0
Negative (%)	0.0%
Memory size	7.1 KiB

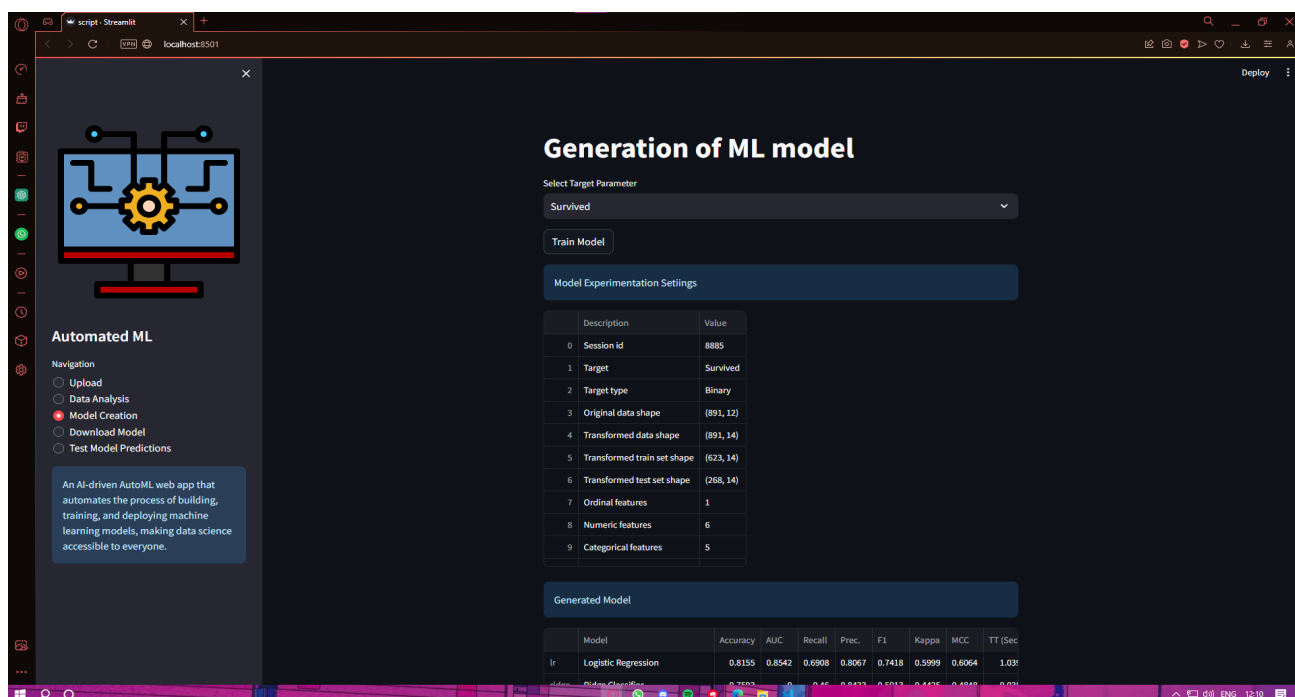
3.



## Model Creation Page:

In this we select the target variable and let the webapp build and compare different training algorithm results and bring out the best model.

1.



2.

Automated ML

Navigation

- ☐ Upload
- ☐ Data Analysis
- ☒ Model Creation
- ☐ Download Model
- ☐ Test Model Predictions

An AI-driven AutoML web app that automates the process of building, training, and deploying machine learning models, making data science accessible to everyone.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lr Logistic Regression	0.8155	0.8542	0.6908	0.8067	0.7418	0.5999	0.6064	1.03
ridge Ridge Classifier	0.7593	0	0.46	0.8433	0.5913	0.4425	0.4848	0.03
nb Naive Bayes	0.6628	0.7859	0.1592	0.8342	0.2619	0.1596	0.2511	0.02
et Extra Trees Classifier	0.6612	0.7744	0.1382	0.7933	0.2315	0.1479	0.2412	0.04
knn K Neighbors Classifier	0.644	0.6207	0.3935	0.5672	0.459	0.2063	0.218	0.1
rf Random Forest Classifier	0.6436	0.8002	0.0841	0.74	0.1468	0.0913	0.1794	0.10
lda Linear Discriminant Analysis	0.6196	0.5236	0.0261	0.06	0.0364	0.0179	0.0208	0.02
dt Decision Tree Classifier	0.6164	0.5	0	0	0	0	0	0.02
qda Quadratic Discriminant Analysis	0.6164	0.5	0	0	0	0	0	0.03
ada Ada Boost Classifier	0.6164	0.5	0	0	0	0	0	0.02

Generated Model

Deploy

## Downloading the generated model:

In this page the best generated model can be downloaded as pkl file.

Automated ML

Navigation

- ☐ Upload
- ☐ Data Analysis
- ☐ Model Creation
- ☒ Download Model
- ☐ Test Model Predictions

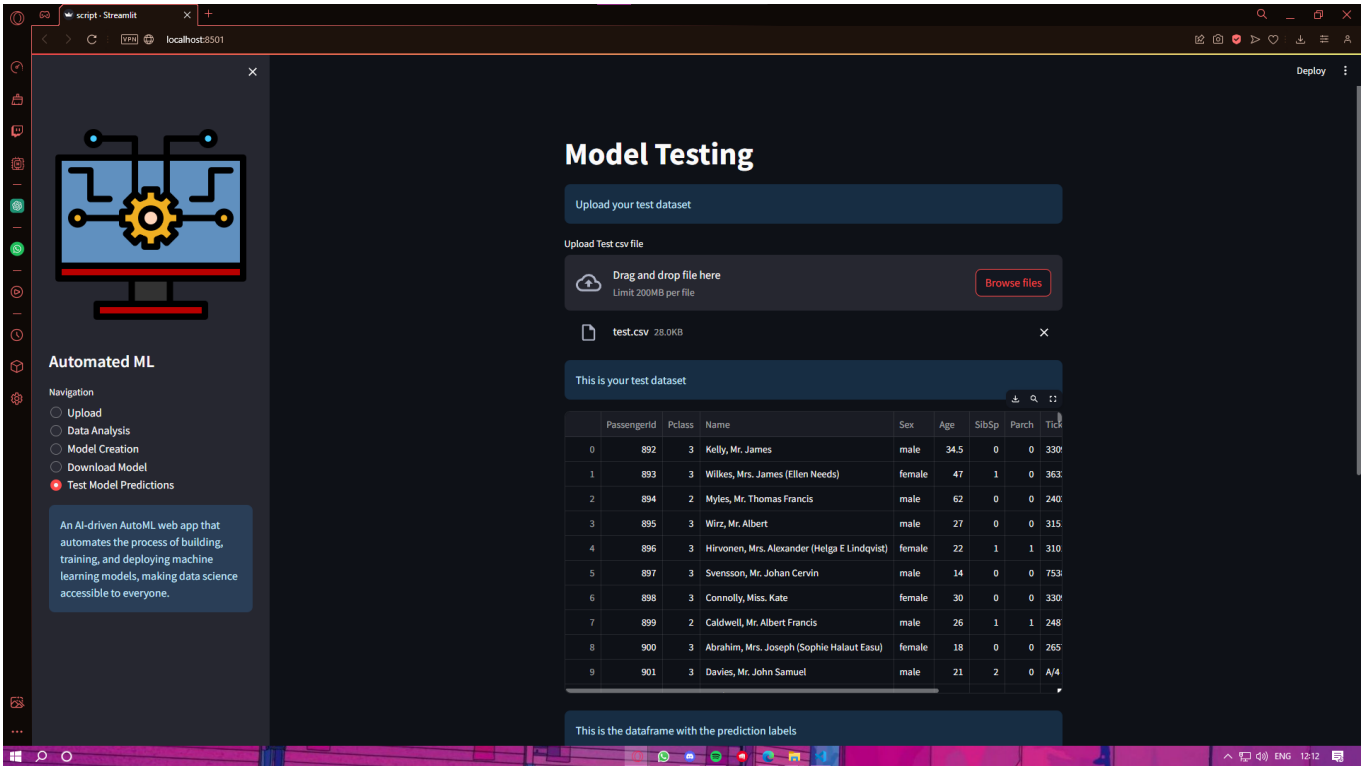
An AI-driven AutoML web app that automates the process of building, training, and deploying machine learning models, making data science accessible to everyone.

You Can Download the generated model from here

Download the Model

# Testing the generated model:

1.



2.

Here the prediction label is the target column as predicted by the model.

This is the dataframe with the prediction labels

	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	prediction_label	prediction_score
0	34.5	0	0	330911	7.8292	None	Q	0	0.8576
1	47	1	0	363272	7	None	S	0	0.6806
2	62	0	0	240276	9.6875	None	Q	0	0.822
3	27	0	0	315154	8.6625	None	S	0	0.8436
4	22	1	1	3101298	12.2875	None	S	1	0.5123
5	14	0	0	7538	9.225	None	S	0	0.7835
6	30	0	0	330972	7.6292	None	Q	1	0.6181
7	26	1	1	248738	29	None	S	0	0.6226
8	18	0	0	2657	7.2292	None	C	1	0.6849
9	21	2	0	A/4 48871	24.15	None	S	0	0.9083

# **CONCLUSION**

In conclusion, the proposed development of an Automated Machine Learning (AutoML) application stands poised to redefine the landscape of predictive modeling by addressing the critical challenges surrounding accessibility, complexity, and expertise in machine learning.

The vision behind this project was to democratize machine learning by introducing a user-friendly, comprehensive AutoML platform that encapsulates the power of advanced machine learning libraries, such as PyCaret, to streamline and automate the model development process. By incorporating features like intuitive interfaces, automated workflows, diverse algorithms, model interpretability, scalability, and educational resources, the application aims to break down the barriers hindering widespread adoption of machine learning.

The potential impact of this AutoML application extends far beyond simplifying model development. It represents a catalyst for change across industries, empowering individuals from diverse backgrounds to leverage data-driven insights for informed decision-making. By making machine learning accessible to a wider audience, organizations can unlock previously untapped potential within their datasets, driving innovation, improving operational efficiency, and fostering a culture of data-driven decision-making.

Moreover, the democratization of machine learning facilitated by this application aligns with the broader goal of empowering individuals and organizations to harness the transformative capabilities of data. It bridges the gap between technical complexities and practical application, empowering users to unlock actionable insights and drive impactful changes within their respective domains.

As this AutoML application paves the way for democratized machine learning, its potential to revolutionize industries, drive innovation, and enable informed decision-making cannot be overstated. It represents a significant step towards a future where the power of predictive analytics is wielded by a diverse array of individuals, revolutionizing how we interact with and derive value from data.



## **FUTURE SCOPE**

This future scope envisions a roadmap for expanding and enhancing the AutoML application, incorporating advanced features, addressing specific industry needs, promoting ethical AI, and fostering a collaborative and adaptive environment to cater to evolving user requirements and technological advancements.

Following are some improvements that can be made to the current application:

1. Adding a separate tab for classification problems.
2. Exception handling for certain misinputs on the user's end.
3. Pointing out to the user on what steps to take on the data based on the EDA performed.
4. Deploying the model on a live server.

# **BIBLIOGRAPHY**

We would like to thank our respected mentors for helping with guiding us throughout the entire process of building the project application.

Other helpful sources include:

1. freecodecamp
2. pycaret documentation
3. streamlit community and documentations
4. Youtube channels(Codebasics, Krish Naik, CodewithHarry etc)