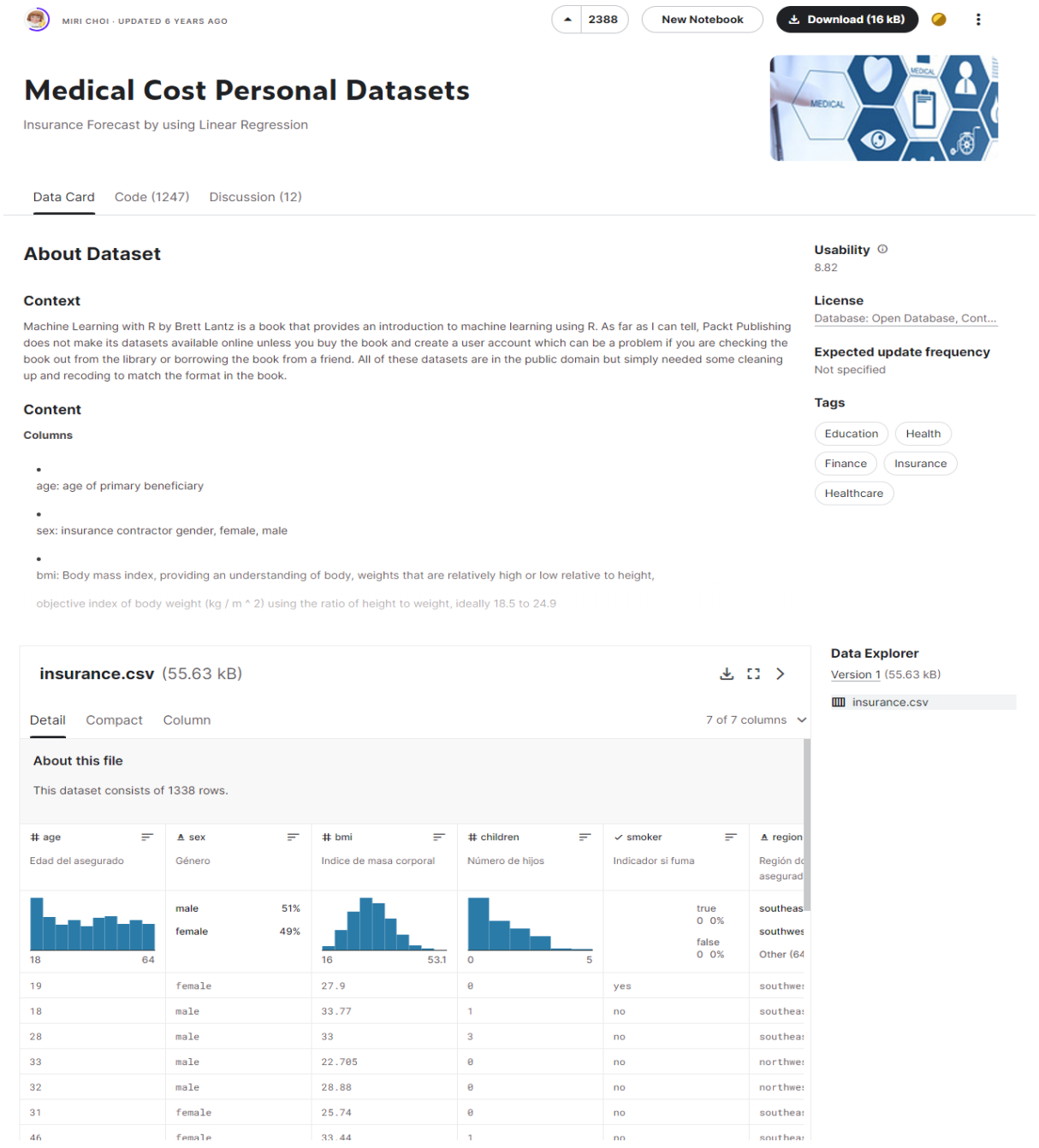
Assignment4 - Create Regression Model with Real Dataset & Model Deployment

1. Choose a real-world dataset that contains at least three features and a numeric target variable. that you want to predict. Ensure the dataset is suitable for regression analysis.



Dataset: https://www.kaggle.com/datasets/mirichoi0218/insurance

2. Create a regression model to predict a target variable based on features from the dataset.

Step 1: Create a Linear Regression Model

```
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np

# Load the dataset
df = pd.read_csv("insurance.csv")

# Define the features and target variable
features = ['age', 'bmi', 'children']
target = 'charges'

X = df[features]
y = df[target]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Create and train the Linear Regression model with named features
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)

# Calculate R-squared (R²)
r_squared = r2_score(y_test, y_pred)

# Print the metrics
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r_squared:.2f}")
```

Step 2: Save the Model

```
import joblib

# Save the trained model to a file
model_filename = 'linear_regression_model.pkl'
joblib.dump(model, model_filename)
```

Step 3: Load the Model

```
import joblib

# Define the filename where the model was saved in Step 2
model_filename = 'linear_regression_model.pkl'

# Load the saved model
loaded_model = joblib.load(model_filename)
```

Step 4: Show an Example of Prediction

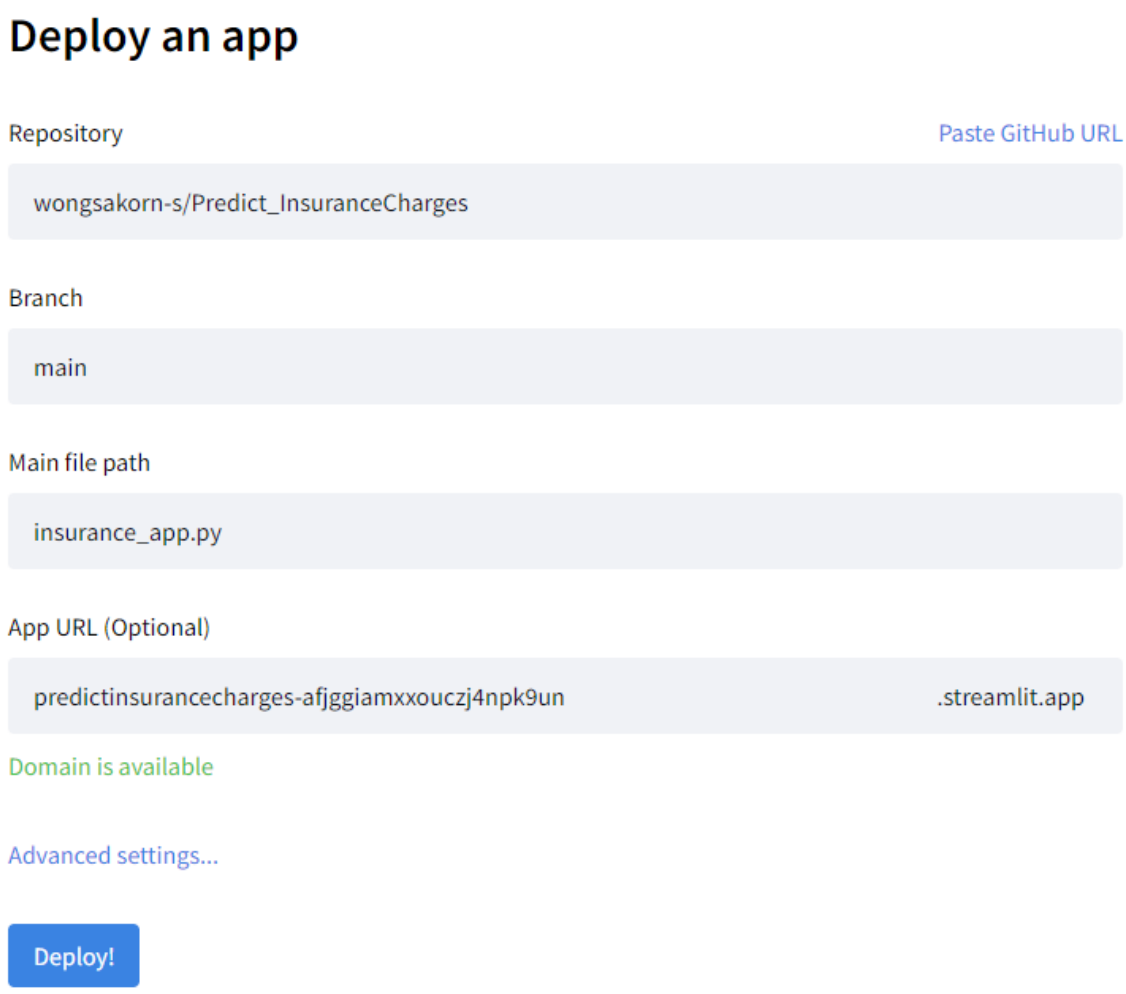
```
# Example input for prediction with feature names
example_data = pd.DataFrame([[30, 25, 2]], columns=features)  # Provide age,
bmi, and children values with feature names

# Use the loaded model to make predictions
predicted_charges = loaded_model.predict(example_data)

# Print the predicted charges
print(f"Predicted Charges: {predicted_charges[0]:.2f}")
```

Step 5: Create a Streamlit App with new file ‘insurance_app.py’

```
import streamlit as st
import joblib
import pandas as pd
from sklearn.metrics import mean_squared_error
import numpy as np

# Load the saved model
loaded_model = joblib.load('linear_regression_model.pkl')
```

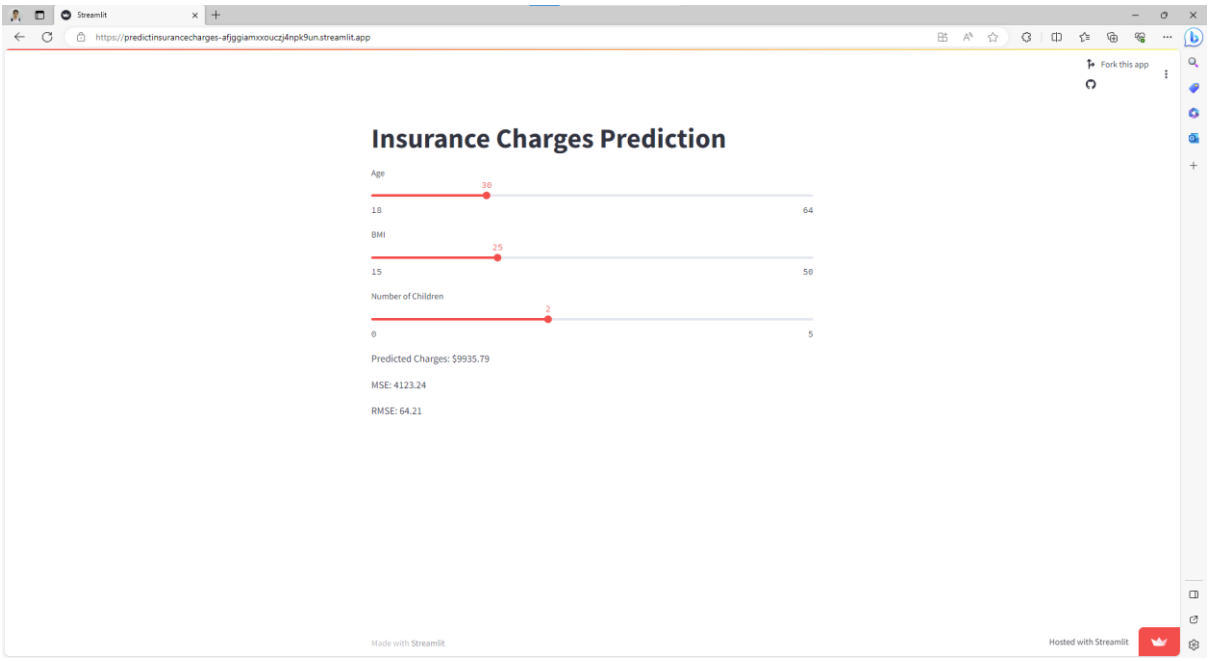
```
# Define the features for input
features = ['age', 'bmi', 'children']

# Define the Streamlit app
st.title("Insurance Charges Prediction")

# Add input fields for user to enter data
age = st.slider("Age", min_value=18, max_value=64, value=30)
bmi = st.slider("BMI", min_value=15, max_value=50, value=25)
children = st.slider("Number of Children", min_value=0, max_value=5, value=2)

# Create a DataFrame with the user input
example_data = pd.DataFrame([[age, bmi, children]], columns=features)

# Make predictions
predicted_charges = loaded_model.predict(example_data)

# Display the prediction
st.write(f"Predicted Charges: ${predicted_charges[0]:.2f}")

# Add calculation of Mean Squared Error (MSE) and Root Mean Squared Error
(RMSE)
actual_charges = 10000  # Replace with the actual charges if available
if actual_charges:
    mse = mean_squared_error([actual_charges], [predicted_charges[0]])
    rmse = np.sqrt(mse)
    st.write(f"MSE: {mse:.2f}")
    st.write(f"RMSE: {rmse:.2f}")
else:
    st.write("Actual charges not provided. Unable to calculate MSE and RMSE.")
```

3. Deploy the regression model as a web application using Streamlit.io.

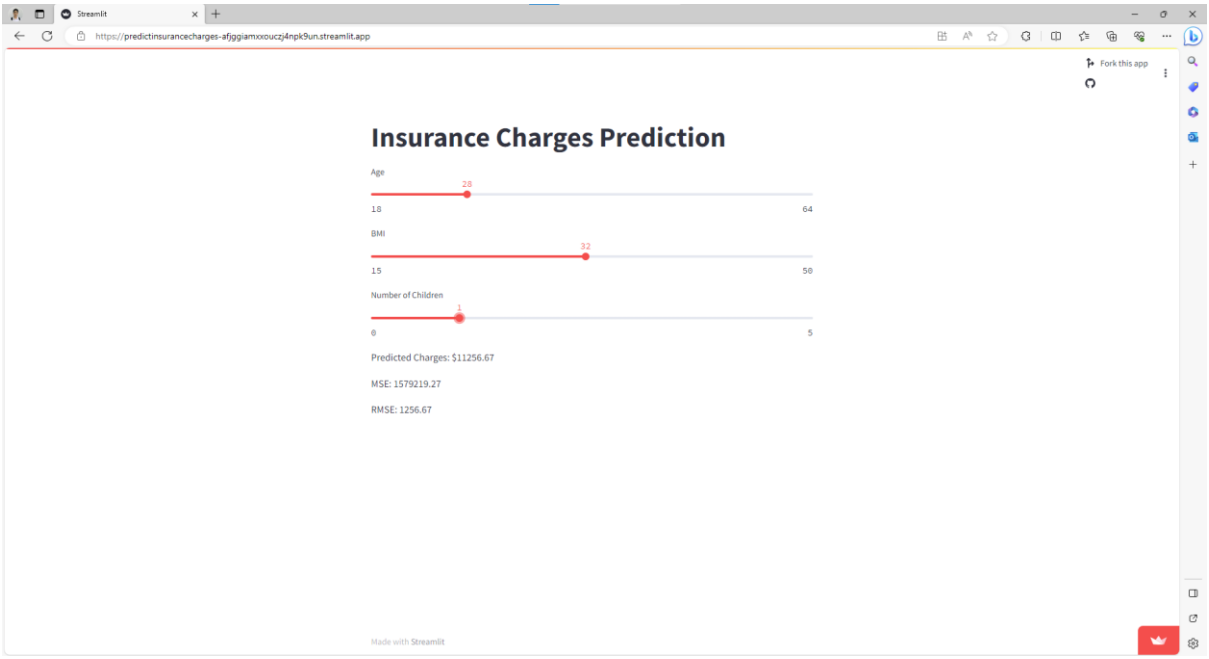


4. Put the evaluation result on your web app.

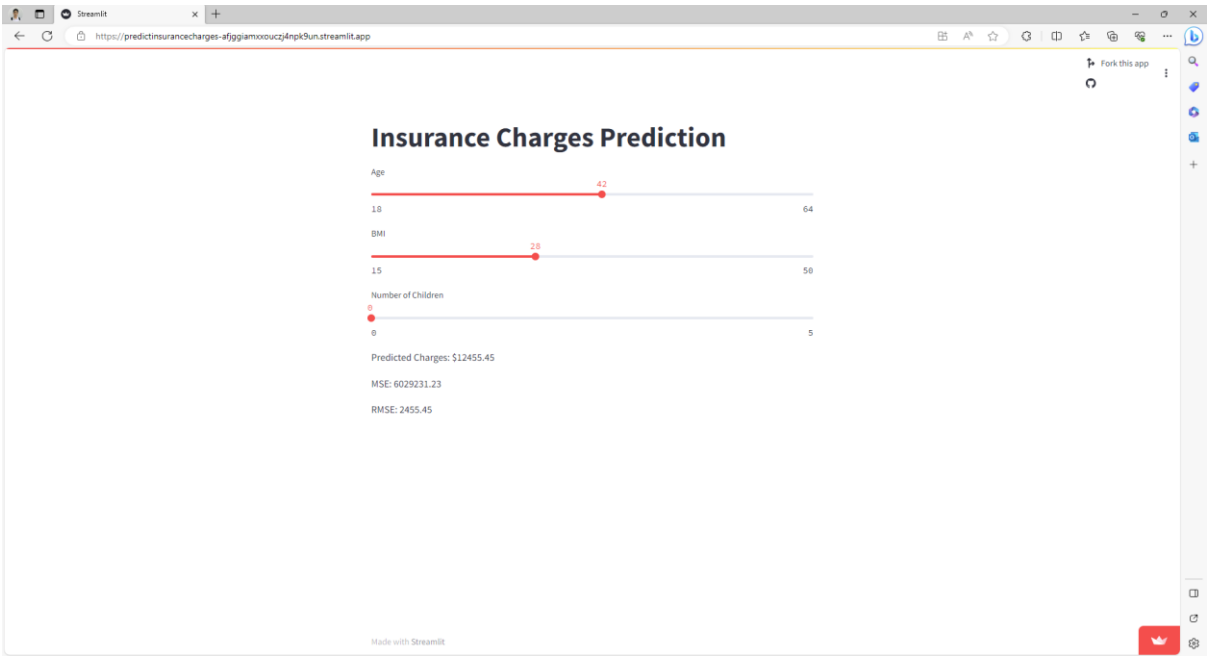
Result 1



Result 2



Result 3



5. In the Streamlit Sharing settings for your app, make sure to set it to "Public" or "Anyone with the link can view.

Link: https://predictinsurancecharges-afjggiamxxouczj4npk9un.streamlit.app/