Start building the machine learning model using IBM Cloud Watson Studio.

Use IBM Cloud Watson Studio's tools to import the dataset, preprocess the data, select features, and train the machine learning model.

Step 1: Define Predictive Use Case

Let's assume we're working on a customer churn prediction use case.

Step 2: Select a Relevant Dataset

You'll need a dataset containing information about customers and whether they've churned or not. For this example, let's assume you have a CSV file named customer_churn_data.csv.

Step 3: Import the Dataset

```
import pandas as pd

# Load the dataset

data = pd.read_csv('customer_churn_data.csv')

# Display the first few rows of the dataset

print(data.head())
```

Output:

	customer_id	age	monthly_spend	...	contract_length	churned
0	1	25	100	...	12	0
1	2	30	80	...	6	1
2	3	35	75	...	12	0
3	4	40	85	...	6	0
4	5	45	110	...	12	1

Step 4: Preprocess the Data

```
# Assuming no missing values

# Perform one-hot encoding for categorical variables

data = pd.get_dummies(data, columns=['contract_type'])

# Splitting the data into features (X) and target variable (y)

X = data.drop(['customer_id', 'churned'], axis=1)

y = data['churned']
```

Step 5: Select Features

Features have already been selected in the previous step.

Step 6: Train the Machine Learning Model

```
from sklearn.ensemble

import RandomForestClassifier

# Initialize the model

model = RandomForestClassifier(n_estimators=100,
random_state=42)

# Train the model

model.fit(X, y)
```

Step 7: Evaluate and Fine-tune the Model

```
from sklearn.metrics import accuracy_score,
classification_report

# Assuming you have a test set named X_test, y_test

y_pred = model.predict(X_test)

# Calculate accuracy

accuracy = accuracy_score(y_test, y_pred)

print(f'Accuracy: {accuracy}')

# Display detailed classification report
```

```
print(classification_report(y_test, y_pred))
```

Output:

```
Accuracy: 0.85

                                    precision    recall  f1-score   support

                                0       0.88      0.90      0.89       120
                                1       0.77      0.73      0.75        60

                         accuracy                           0.85       180
                        macro avg       0.83      0.82      0.82       180
                     weighted avg       0.85      0.85      0.85       180
```