



EXPECT LAPTOP PRICES IN NORTHERN SYRIA

INTERNATIONAL UNIVERSITY OF SCIENCE AND RENAISSANCE

THE RESEARCH AIMS TO PROVIDE ASSISTANCE TO ALL
PEOPLE IN CHOOSING THE APPROPRIATE COMPUTER BASED
ON THE PRICE

IT HELPS THEM DETERMINE THE PRICE SO THAT THEY ARE
NOT DECEIVED

PREPARING STUDENTS:
YUSUF EL HMDO
BLAL SED ASAD
KALED KDE

ACADEMIC SUPERVISION:
ABD ELMHEMN AL ESMAEEL

مقدمة:

الذكاء الاصطناعي (AI) هو مجال متقدم في علوم الحاسوب يهدف إلى تطوير أنظمة ذكية قادرة على أداء مهام تتطلب التفكير البشري. يسعى الذكاء الاصطناعي إلى إنشاء برامج وأنظمة قادرة على فهم البيئة المحيطة بها، واتخاذ قرارات مستنيرة، وتعلم من الخبرة، مما يسمح لها بتحسين أدائها مع مرور الوقت.

يعتمد الذكاء الاصطناعي على العديد من التقنيات والمفاهيم، منها تعلم الآلة (Machine Learning) الذي يمكن الأنظمة من التكيف وتطوير أدائها دون الحاجة إلى برمجة يدوية. يشمل ذلك تعلم الآلة العميق (Deep Learning)، الذي يعتمد على الشبكات العصبية الاصطناعية لمحاكاة الطريقة التي يعمل بها الدماغ البشري في معالجة المعلومات.

توسيع نطاق الذكاء الاصطناعي يتيح للأنظمة التفاعل بشكل أكثر فهمًا مع البيئة والمستخدمين، ويمكنها تنفيذ مهام معقدة تتنوع من التعرف على الصوت والصورة إلى فهم اللغة الطبيعية واتخاذ قرارات استنادًا إلى بيانات هائلة.

يعد التطور السريع في مجال الذكاء الاصطناعي مصدر إلهام وتحدي للباحثين والمهندسين في جميع أنحاء العالم. يُتوقع أن يلعب الذكاء الاصطناعي دورًا حيويًا في تحسين العديد من الجوانب في حياتنا اليومية، بدءًا من الطب والتعليم وصولاً إلى الصناعة والاقتصاد.

مع هذا السياق المتزايد للتطور التكنولوجي، يثير الذكاء الاصطناعي أيضًا تحديات أخلاقية واجتماعية، وي طرح أسئلة حول الخصوصية والأمان، ويفتح أفقًا لمناقشات حول الآثار الاقتصادية والاجتماعية لهذه التكنولوجيا.

في هذه المقدمة، سنستكشف عمق مفاهيم الذكاء الاصطناعي، وكيف يشكل تقدمه تحولًا ثوريًا في عدة مجالات، مع التركيز على التحديات والمزايا التي قد تظهر مع تقدم هذا المجال المثير.

التعلم الآلي (Machine Learning) هو فرع من الذكاء الاصطناعي يركز على تطوير تقنيات تمكن الأنظمة الحاسوبية من تعلم وتحسين أدائها تلقائيًا بدون الحاجة إلى برمجة يدوية. في حين أن البرمجة التقليدية تتطلب تحديد صريح للقواعد والأوامر التي يجب أن تتبعها النظام، يعتمد التعلم الآلي على إعطاء الآلة القدرة على استخدام البيانات لتحسين أدائها.

أنواع التعلم الآلي: هناك ثلاثة أنواع رئيسية من التعلم الآلي:

١. تعلم الرقابة Supervised Learning:

- في هذا النوع، يتم تدريب النموذج باستخدام مجموعة من البيانات المسماة (مثل صور مع تصنيفاتها أو أسعار المنازل مع الميزات ذات الصلة).

- الهدف هو تعلم النموذج لتحديد العلاقة بين المدخلات والمخرجات حتى يكون قادرًا على التنبؤ بالمخرجات لبيانات جديدة غير مرصودة سابقًا.

٢. تعلم بدون رقابة Unsupervised Learning:

- هنا، يتم تدريب النموذج باستخدام بيانات غير مسماة، ويتعلم من هيكل البيانات دون توجيه خارجي.
- الهدف هو فهم التركيب والعلاقات في البيانات، ومن ثم استخراج المعلومات الكامنة فيها.

٣. تعلم التعزيز Reinforcement Learning:

- في هذا النوع، يتفاعل النموذج مع بيئة ديناميكية ويتلقى مكافآت أو عقوبات استنادًا إلى الإجراءات التي يقوم بها.
- الهدف هو تعلم النموذج كيفية اتخاذ سلسلة من القرارات لتحقيق هدف معين مع مرور الوقت.

تتضمن تقنيات التعلم الآلي العديد من الألوان، مثل تعلم الآلة العميق (Deep Learning)، والذي يستخدم شبكات عصبية متعددة الطبقات لفهم وتحليل بيانات معقدة. تطبيقات التعلم الآلي واسعة النطاق وتشمل تحليل البيانات، والتعرف على الصور، والترجمة الآلية، وألعاب الكمبيوتر، والطب، والمزيد.

في النهاية، يساهم التعلم الآلي في تمكين الأنظمة من الاستفادة من الخبرة وتكامل المعلومات بشكل أفضل، مما يجعلها قادرة على التكيف والتحسين مع تغييرات البيئة والمتطلبات.

واجهة التطبيق

هذا الكود هو تطبيق Flutter يستخدم لتوقع سعر اللابتوب باستخدام خدمة عبر الإنترنت. دعونا نقسم الشرح إلى عدة أقسام:

١. الجزء الرئيسي:

- يتم استيراد مكتبات Flutter ومكتبة "http" لإجراء طلبات HTTP.
- تُشغل الدالة "main" لتهيئة وتشغيل التطبيق.
- يتم تعريف تطبيق ("MyApp") كتطبيق واجهة المستخدم الرئيسي باستخدام مكتبة "Material".
- يتم استدعاء "MyApp" باستخدام "runApp" في الدالة "main".

٢. صفحة التطبيق:

- تمثل صفحة التطبيق الرئيسية الكلاس "LaptopPricePrediction" وهي StatefulWidget.
- يتم تعريف حقول الإدخال باستخدام "TextEditingController" لمراقبة قيمة الإدخال.

- هناك مجموعة من المتغيرات (`selectedCompany`, `selectedCore`, ... إلخ) التي تُستخدم لتخزين القيم المحددة من قبل المستخدم من القوائم المنسدلة.

٣. الطلب عبر الإنترنت:

- تم تعريف دالة "predictPrice" التي تقوم بإرسال طلب POST إلى خدمة عبر الإنترنت باستخدام مكتبة `http`.
- يتم تحميل البيانات المرسله في الطلب وفحص الرد للتحقق من نجاح العملية.
- في حال نجاح الطلب، يتم استخدام البيانات المتلقاة لتحديد السعر المتوقع، ويتم تحديث حالة الواجهة (`predictedPrice`).

٤. واجهة المستخدم:

- يتم بناء واجهة المستخدم باستخدام عناصر مثل `Scaffold`، و `AppBar`، و `TextField`، و `ElevatedButton`، و `DropDownButtonFormField`.
- هناك قائمة منسدلة لكل معلمة يقوم المستخدم بتحديد لها لتوقع سعر اللابتوب.
- يوجد زر "Predict" يقوم بفتح دالة `predictPrice` عند النقر عليه.
- يظهر سعر التوقع في الجزء السفلي من الواجهة.

٥. الملحقات:

- يوجد ملفات الاستيراد (`import`) في البداية لتضمين المكتبات اللازمة.

تلاحظ أن الكود يعتمد على استخدام خدمة عبر الإنترنت لتوقع سعر اللابتوب، ويفترض أن يكون هناك خادم يعمل على `http://localhost:5000/predict` ليتم التواصل معه والحصول على النتائج. يجب التحقق من أن الخدمة الخلفية تعمل بشكل صحيح قبل استخدام هذا التطبيق.

كود بايثون متعلق بعمليات التعلم

يستخدم مكتبتَي Pandas و Openpyxl في لغة Python لقراءة ملف إكسل وعرض أول ١٠ سجلات من البيانات. دعوني أشرح الكود لك:

١. استيراد المكتبات:

- `import pandas as pd`: يقوم باستيراد مكتبة Pandas ويعيد تسميتها إلى `pd` لسهولة الاستخدام.
- `import openpyxl`: يقوم باستيراد مكتبة Openpyxl التي تستخدم للتعامل مع ملفات Excel.

٢. تحديد مسار الملف:

- `'file_path = 'dataSetLabtupMassionLarning.xlsx'` : يقوم بتعيين اسم ومسار ملف Excel الذي يحتوي على البيانات.

٣. قراءة البيانات من ملف Excel:

- `df = pd.read_excel(file_path)` : يستخدم Pandas لقراءة البيانات من ملف Excel وتخزينها في DataFrame (جدول بيانات ثنائي الأبعاد) باسم `df`.

٤. عرض أول ١٠ سجلات:

- `df.head(١٠)` : يقوم بعرض أول ١٠ سجلات من DataFrame، حيث يمكنك تغيير العدد داخل `head` إذا كنت ترغب في عرض عدد مختلف من السجلات.

لضمان أن الكود يعمل بشكل صحيح، تأكد من وجود ملف Excel في المسار الذي تم تحديده `('dataSetLabtupMassionLarning.xlsx')`. كما يجب أيضاً أن يكون لديك Pandas و Openpyxl مثبتين في بيئة Python الخاصة بك.

هذا الكود يستخدم مكتبة Pandas ومكتبة LabelEncoder من مكتبة scikit-learn في Python لتحويل القيم النصية في عدة أعمدة في DataFrame إلى قيم رقمية. دعوني أشرح الكود لك:

١. استيراد المكتبات:

- `import pandas as pd` : يقوم باستيراد مكتبة Pandas ويعيد تسميتها إلى `pd` لسهولة الاستخدام.
- `from sklearn.preprocessing import LabelEncoder` : يستورد فقط `LabelEncoder` من مكتبة scikit-learn.

٢. تحديد مسار الملف:

- `'file_path = 'dataSetLabtupMassionLarning.xlsx'` : يعين اسم ومسار ملف Excel الذي يحتوي على البيانات.

٣. قراءة البيانات من ملف Excel:

- `df = pd.read_excel(file_path)`: يستخدم Pandas لقراءة البيانات من ملف Excel وتخزينها في DataFrame (جدول بيانات ثنائي الأبعاد) باسم `df`.

٤. تحويل القيم النصية إلى رقمية:

- `df['Core'] = df['Core'].astype(str)`: يقوم بتحويل العمود 'Core' إلى نوع نصي.

- `df['Ryzen'] = df['Ryzen'].astype(str)`: يقوم بتحويل العمود 'Ryzen' إلى نوع نصي.

- `text_columns = [...]`: يقوم بتعريف قائمة بأسماء الأعمدة التي تحتوي على قيم نصية.

- `label_encoder = LabelEncoder()`: يقوم بإنشاء كائن `LabelEncoder` لتحويل القيم النصية إلى قيم رقمية.

- يتم استخدام حلقة `for` لتحويل كل عمود نصي في القائمة `text_columns` باستخدام `fit_transform`.

٥. حفظ البيانات المحدثة في ملف Excel جديد:

- `output_file_path = 'newlearningmasheen4.xlsx'`: يقوم بتعيين اسم ومسار ملف Excel الجديد الذي سيحتوي على البيانات المحدثة.

- `df.to_excel(output_file_path, index=False)`: يقوم بحفظ DataFrame في ملف Excel جديد دون إضافة الفهرس (index).

هذا الكود يكون مفيداً عندما تحتاج إلى تحويل القيم النصية في البيانات إلى أشكال رقمية لاستخدامها في تدريب نموذج تعلم الآلة، حيث يعتبر `LabelEncoder` أحد الأدوات المفيدة لتحقيق ذلك.

هذا الكود يستخدم مكتبة Pandas ومكتبة `train_test_split` من scikit-learn لتقسيم البيانات إلى مجموعات تدريب واختبار. دعوني أشرح الكود لك:

١. استيراد المكتبات:

- `import pandas as pd`: يقوم باستيراد مكتبة Pandas ويعيد تسميتها إلى `pd` لسهولة الاستخدام.

- `from sklearn.model_selection import train_test_split`: يستورد `train_test_split` من مكتبة scikit-learn.

٢. اختيار الميزات والهدف:

- `features = df.drop('price', axis=1)`: يقوم بإنشاء DataFrame يحتوي على الميزات ('features') بناءً على البيانات الأصلية باستثناء عمود 'price'.

- `target = df['price']`: يقوم بإنشاء سلسلة (`target`) تحتوي على الهدف المرتبط بالتنبؤ والذي هو عمود `'price'`.

٣. قسم البيانات:

- `train_test_split(features, target, test_size=0.2, random_state=42)`: يقوم بتقسيم البيانات إلى مجموعتين، مجموعة تدريب (`X_train`, `y_train`) ومجموعة اختبار (`X_test`, `y_test`).

- `features` هي المتغيرات المستقلة (الميزات) و `target` هو المتغير التابع (الهدف).

- `test_size=0.2` يعني أن ٢٠٪ من البيانات ستذهب إلى مجموعة الاختبار، والبقية ستذهب إلى مجموعة التدريب.

- `random_state=42` يضبط البذرة لضمان تكرار الانقسام بشكل قابل للتكرار، مما يعني أن نفس الانقسام سيحدث إذا تم تشغيل الكود مرة أخرى.

بهذه الطريقة، يمكنك الحصول على مجموعات متزنة تحتوي على بيانات تدريب واختبار لاستخدامها في تدريب وتقييم نموذج تعلم الآلة الخاص بك.

هذا الكود يستخدم مكتبة `scikit-learn` لتدريب نموذج الانحدار الخطي (`Linear Regression`) على مجموعة البيانات وثم يستخدم النموذج لتوقع أسعار اللايتوب باستخدام مجموعة الاختبار. دعوني أشرح الكود لك:

١. استيراد المكتبات:

- `from sklearn.linear_model import LinearRegression`: يستورد نموذج الانحدار الخطي من `scikit-learn`.

٢. إنشاء نموذج الانحدار الخطي:

- `model = LinearRegression()`: يقوم بإنشاء كائن نموذج الانحدار الخطي.

٣. تدريب النموذج:

- `model.fit(X_train, y_train)`: يقوم بتدريب النموذج باستخدام مجموعة التدريب (`X_train`, `y_train`).

٤. توقع الأسعار باستخدام النموذج:

- `predicted_prices = model.predict(X_test)`: يقوم بتوقع أسعار اللايتوب باستخدام النموذج ومجموعة الاختبار (`X_test`).

بعد تشغيل هذا الكود، يمكنك الوصول إلى `predicted_prices` الذي يحتوي على التوقعات المحسوبة لأسعار اللايتوب باستخدام النموذج. يمكنك استخدام هذه التوقعات لتقييم أداء النموذج أو لأي غرض تحليلي آخر.

هذا الكود يستخدم مكتبة scikit-learn لحساب متوسط مربعات الخطأ (Mean Squared Error)، وهي إحدى مقاييس تقييم أداء النموذج. دعوني أشرح الكود لك:

١. استيراد المكتبة:

- `from sklearn.metrics import mean_squared_error` : يستورد دالة حساب متوسط مربعات الخطأ من مكتبة scikit-learn.

٢. توقع الأسعار باستخدام النموذج:

- `predicted_prices = model.predict(X_test)` : يقوم بتوقع أسعار اللايتوب باستخدام النموذج ومجموعة الاختبار (`X_test`).

٣. حساب متوسط مربعات الخطأ (MSE):

- `mse = mean_squared_error(y_test, predicted_prices)` : يقوم بحساب متوسط مربعات الخطأ باستخدام القيم الفعلية (`y_test`) والقيم المتوقعة (`predicted_prices`).

٤. طباعة قيمة MSE:

- `print("Mean Squared Error (MSE):", mse)` : يقوم بطباعة قيمة متوسط مربعات الخطأ.

قيمة MSE هي قياس لمدى انحراف التوقعات عن القيم الفعلية، حيث يكون المثلى هو أن تكون هذه القيمة قريبة من الصفر. يمكنك استخدام هذه القيمة لتقييم أداء النموذج، حيث أن قيمة MSE أقل تشير إلى أن النموذج يقوم بالتنبؤ بشكل أفضل بالبيانات وهنا قيمة الخطأ Mean Squared Error (MSE): 29872.91715288435

هذا الكود يقوم بتحليل البيانات باستخدام نموذج الانحدار الخطي في Python باستخدام مكتبات مثل pandas و scikit-learn. سأشرح الخطوات بشكل موجز:

١. استيراد المكتبات:

- `pandas` : للتعامل مع البيانات بشكل هيكلي.

- `train_test_split` من `sklearn.model_selection` : لتقسيم البيانات إلى مجموعات تدريب واختبار.

- `LinearRegression` من `sklearn.linear_model`: لبناء نموذج الانحدار الخطي.
- `mean_squared_error` من `sklearn.metrics`: لحساب متوسط مربعات الخطأ.

٢. قراءة البيانات:

- يتم قراءة ملف (`newlearningmasheen4.xlsx`) Excel باستخدام pandas.

٣. اختيار الميزات والهدف:

- تُختار الميزات (`features`) والهدف (`target`) من البيانات.

٤. تقسيم البيانات:

- يتم تقسيم البيانات إلى مجموعات تدريب واختبار باستخدام `train_test_split`.

٥. بناء وتدريب النموذج:

- يتم بناء نموذج الانحدار الخطي باستخدام `LinearRegression` وتدريبه على مجموعة التدريب.

٦. توقع الأسعار:

- يتم استخدام النموذج لتوقع أسعار اللابتوب باستخدام مجموعة الاختبار.

٧. حساب متوسط مربعات الخطأ (MSE):

- يتم حساب MSE باستخدام `mean_squared_error` بين الأسعار الفعلية في مجموعة الاختبار والأسعار المتوقعة.

٨. طباعة نتائج:

- يتم طباعة قيمة MSE.

هذا الكود يسمح لك بفحص أداء نموذج الانحدار الخطي على مجموعة الاختبار باستخدام متوسط مربعات الخطأ كمقياس للدقة.

هذا الكود يُعرض دالة `predict_laptop_price` التي تستخدم النموذج الذي تم تدريبه مسبقًا لتوقع سعر اللابتوب باستناد إلى المواصفات التي يتم تمريرها كمدخلات. هنا هو شرح للكود:

١. الدالة `predict_laptop_price`:

- تقوم بأخذ المدخلات التي تمثل المواصفات المختلفة لللابتوب (`core`، `GPU type`، `RAM size`، وما إلى ذلك).
- تُعد هذه المدخلات وتُمثل في شكل قائمة متداخلة `'laptop_features'`.
- يتم استخدام النموذج (`'model'`) لتوقع سعر اللابتوب باستخدام `'predict'`.
- السعر المتوقع يُسترجع كقيمة من الدالة.

٢. المواصفات المذكورة:

- تمثل متغيرات (`'laptop_gpu_type'`، `'laptop_core'`، ...) المواصفات المعينة لللابتوب الذي ترغب في توقع سعره.
- تُمرر هذه المتغيرات إلى الدالة `'predict_laptop_price'`.

٣. الطباعة:

- يتم طباعة السعر المتوقع باستخدام الدالة `'predict_laptop_price'`.

النموذج الخطي (`'model'`) الذي تم بناؤه مسبقًا يستخدم هذه المواصفات لتوقع سعر اللابتوب. يُفترض أن يتم استدعاء هذه الدالة باستخدام المواصفات الفعلية للحصول على تقدير لسعر اللابتوب.
