Syrian Arab Republic

Lattakia - Tishreen University

Department of Communication and electrical engineering

5th , Network Programming : Homework No1

الجمهورية العربية السورية

اللاذقية –جامعـة تشريــن

كلية الهندسة الكهربائية والميكانيكية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة: مشروع برمجة شبكات

توقع سعر أسهم شركة تيسلا باستخدام تعلم الآلة في بايثون

تقديم الطالبتين:

رغد حكيم عليا عرنوق

إشراف الدكتور:

مهند عيسى

ملخص

تعتبر لغة بايثون هي اللغة الأكثر شيوعاً في التعامل مع الذكاء الاصطناعي وتكاد تصبح اللغة الوحيدة المتخصصة بهذا المجال. الذكاء الصنعي هو ثورة تكنولوجية ضخمة لما يقدمه من تطبيقات مثل تعلم الآلة، ومن تطبيقات لغة الآلة هو توقع الاسعار. في هذا المشروع سنقوم ببناء نموذج شبكة عصبية AI للتنبؤ بأسعار الأسهم. على وجه التحديد ، سنعمل مع أسهم شركة Tesla لمالكها الشهير Elon Musk ، سنستخدم مجموعة بيانات أسهم TESLA من Yahoo Finance، والتي تحتوي على بيانات الأسهم لمدة عشر سنوات. وسنستفيد من منصة Google colab لمعالجة البيانات الخاصة بنا.

Abstract: Python is the most popular language for dealing with artificial intelligence, and it almost becomes the only language specialized in this field. Artificial intelligence is a huge technological revolution because it offers applications such as machine learning, and among the applications of machine language is price prediction. In this project we will build an AI neural network model for forecasting stock prices. Specifically, we will be working with Elon Musk's famous owner Tesla stock, we will use the TESLA stock data set from Yahoo Finance, which contains stock data for ten years. We will make use of the Google colab platform to process our data.

المحتويات

قائمة الاشكال:

1. مقدمة

بحث الانسان على مر التاريخ على اختراع يمكنه أن يحاكي العقل البشري في نمط تفكيره ، فقد حاول كل من الفنانين والكتاب وصناع الأفلام ومطوري الألعاب على حد سواء إيجاد تفسير منطقي لمفهوم الذكاء الاصطناعي. فعلى سبيل المثال في عام 1872 تحدث "صموئيل بتلر" في روايته "إريوهون" 1872 عن الآلات والدور الكبير الذي ستلعبه في تطوير البشرية ونقل العالم الى التطور والإزدهار.

وعلى مر الزمن، كان الذكاء الاصطناعى حاضراً فقط في الخيال العلمي، فتارةً ما يسلط الضوء على الفوائد المحتملة للذكاء الاصطناعي على البشرية وجوانبه الإنسانية المشرقة، وتارةً أخرى يسلط الضوء على الجوانب السلبية المتوقعة منه، و يتم تصويره على أنه العدو الشرس للبشرية الذي يعتزم إغتصاب الحضارة والسيطرة عليها.

إن الكثير من الأحاديث التكنولوجية التي تدور مؤخرا باتت تتمحور حول ماهية الذكاء الاصطناعي و المفاهيم المرتبطة به . الذكاء الاصطناعي هو محاكاة للدماغ البشري في القيام ببعض وظائفه المعقدة من اكتساب للمعلومات من المحيط وربطها للتوصل لاستنتاجات محددة متعلقة بأمر معين ، حيث تتم عملية المحاكاة باستخدام أجهزة الحاسب والآلات المتنوعة ، حيث يعتبر فرعاً من فروع علم الحاسوب التي تعنى بخلق آلات وأجهزة ذكية .

يمتلك الذكاء الاصطناعي خصائص كثيرة جعلت منه استثمارا ذا فعالية في كثير من المجالات:

- تطبيق الذكاء الاصطناعي على الأجهزة والآلات يمكنها من التخطيط وتحليل المشكلات باستخدام المنطق.
- يتعرف على الأصوات والكلام والقدرة على تحريك الأشياء.
- تستطيع الأجهزة المتبنية للذكاء الاصطناعي فهم المدخلات وتحليلها جيدا لتقديم مخرجات تلبي احتياجات المستخدم بكفاءة عالية.
- يمكن من التعلم المستمر ،حيث تكون عملية التعلم آلية وذاتية دون خضوعه لإشراف ومراقبة.
- يقدر على معالجة الكم الهائل من المعلومات التي يتعرض لها
- يستطيع ملاحظة الأنماط المتشابهة في البيانات وتحليلها بفعالية أكثر من الادمغة البشرية.
- يستطيع إيجاد الحلول للمشاكل غير المألوفة باستخدام قدراته المعرفية.

2. أدوات البحث

سنستخدم لغة بايثون التي أصبحت اللغة المسيطرة في الذكاء الاصطناعي وباستخدام تعلم الآلة سنستخدم نموذج LSTM و RNN.

سنستخدم مكتبات بايثون التالية:

- NumPy: مكتبة عددية للتعامل مع المصفوفات متعددة الأبعاد. وهي مكتبة بايثون تستخدم للعمل مع المصفوفات. كما أن لديها وظائف للعمل في مجال الجبر الخطي وتحويل فورييه fourier transform والمصفوفات. NumPy هي إختصار لـ Numerical Python. في بايثون لدينا lists تخدم الغرض من المصفوفات، لكنها بطيئة. يهدف NumPy إلى توفير object مصفوفة أسرع بخمسين ضعف من قوائم Python التقليدية.
- Scipy: مكتبة علمية من أجل الخوارزميات العددية وتخصيص أدوات النطاق لمعالجة الإشارة والتسريع والتحسين أيضاً.
- matplotlib: للرسم ثلاثي وثنائي الأبعاد.
- Pandas: هياكل البيانات، تحليل وإخراج البيانات والتلاعب بطريقة عرضها بشكلها الرقمي وبشكل متسلسل مع الزمن. مكتبة الباندا Pandas يمكن استخدامها في لغة بايثون من خلال التعرف على بياناتك و تنظيفها وتحويلها وتحليلها. على سبيل المثال ، لنفترض أنك تريد استكشاف مجموعة بيانات مخزنة في ملف CSV على جهاز الكمبيوتر الخاص بك. سيقوم Pandas باستخراج البيانات من ملف CSV هذا فيجدول DataFrame
- Scikit-learn: أهم مكتبة وهي تضم خوارزميات تعلم الآلة.
- Google Colab وهي منصة على الانترنت خاصة بالتعامل مع الذكاء الاصطناعي باستخدام البايثون وتعطي قدرات معالجة كبيرة.

3. المنهجيات العلمية

1.3. تعلم الآلة:

تعلم الآلة Machine Learning :هو أحد فروع الذكاء الاصطناعي التي تهتم بتصميم وتطوير خوارزميات وتقنيات تسمح للحواسيب بامتلاك خاصية «التعلم». بشكل عام هناك مستويين من التعلم: الاستقرائي والاستنتاجي. يقوم الاستقرائي باستنتاج قواعد وأحكام عامة من البيانات الضخمة. المهمة الأساسية للتعلم الآلي هو استخراج معلومات قيمة من البيانات، بالتالي هو قريب جدا من التنقيب في البياناتdata mining والإحصاء والمعلوماتية النظرية. يستخدم التعلم الآلي في العديد من المجالات من الهندسة إلى الطب.

يتضمن التعلم الآلي عدداً كبيراً من حقول التطبيقات: معالجة اللغات الطبيعية (بالإنجليزية: natural language processing) وتمييز الأنماط (بالإنجليزية: syntactic pattern recognition) ومحركات البحث (بالإنجليزية: search engines) والتشخيص الطبي والمعلوماتية الحيوية والمعلوماتية الكيميائية، تصنيف تسلسلات الدنا، تمييز الكلام (بالإنجليزية: speech recognition) وتمييز الكتابة handwriting recognition، وحتى تمييز الأشياء (بالإنجليزية: object

recognition)، رؤية الحاسوب (بالإنجليزية: computer vision) الألعاب الإستراتيجية وتحريك الروبوت (بالإنجليزية: robot locomotion).

يتم تصنيف خورازميات تعلم الآلة إلى عدة أنواع:

التعلم بالإشراف (Supervised Learning) وهو أحد أشهر أنواع التعلم الآلي ويقوم على وجود بيانات وقرائنها الصحيحة عند وقت التعلم بحيث تشكل هذه البيانات امثلة حقيقية يمكن للنموذج التعلم منها.

شبه التعلم بالإشراف (Semi-supervised learning) وهو عند وجود بيانات مع قرائنها الصحيحة ولكنها محدودة أو غير مكتملة.

التعلم بدون إشراف (Unsupervised Learning) وهو تعلم يَنتُج عن وجود بيانات بدون قرائنها الصحيحة ومن أشهر انواع التعلم بدون إشراف هو التحليل العنقودي.Clustering

التعلم المعزز (Reinforcement learning): وهو أحد أنواع التعلم بدون إشراف، وفيه تتفاعل الآلة مع البيئة وتبني خبراتها بناءً على هذا التفاعل يعتبر التعليم المعزز من انواع التعليم الواعدة والتي قد يكون لها نصيب كبير في حل مسائل معقدة في المستقبل. يجب استخدام التعليم المعزز عندما تكون البئية غير معروفه والا فانه سوف يحتاج الكثير من المصادر الحسابية بدون جدوى فعلية لعملية التعليم.

2.3. الذاكرة الطويلة القصيرة الأمد LSTM:

Long Short-Term Memory (LSTM)هي بنية شبكة عصبية اصطناعية متكررة (RNN) تستخدم في مجال التعلم العميق. على عكس الشبكات العصبية ذات التغذية الأمامية القياسية ، فإن LSTM لديها اتصالات تغذية راجعة. لا يمكنها فقط معالجة نقاط البيانات الفردية (مثل الصور) ، ولكن أيضًا التسلسل الكامل للبيانات (مثل الكلام أو الفيديو). على سبيل المثال ، ينطبق LSTM على مهام مثل التعرف على خط اليد غير المقسم والمتصل، التعرف على الكلام واكتشاف الشذوذ في حركة مرور الشبكة أو أنظمة كشف التسلل.(IDS)

تتكون وحدة LSTM المشتركة من خلية وبوابة إدخال وبوابة إخراج وبوابة نسيان. تتذكر الخلية القيم على فترات زمنية عشوائية وتنظم البوابات الثلاثة تدفق المعلومات داخل الخلية وخارجها.

تعد شبكات LSTM مناسبة تمامًا لتصنيف ومعالجة وعمل التنبؤات بناءً على بيانات السلاسل الزمنية ، حيث يمكن أن يكون هناك فترات تأخير غير معروفة بين الأحداث المهمة في سلسلة زمنية. تم تطوير LSTMs للتعامل مع مشكلة التدرج المتلاشي التي يمكن مواجهتها عند تدريب RNNs التقليدية. تعد الحساسية النسبية لطول الفجوة ميزة لـ LSTM على شبكات RNN ونماذج ماركوف المخفية وطرق التعلم المتسلسلة الأخرى في العديد من التطبيقات.

1.2.3. بنية LSTM :

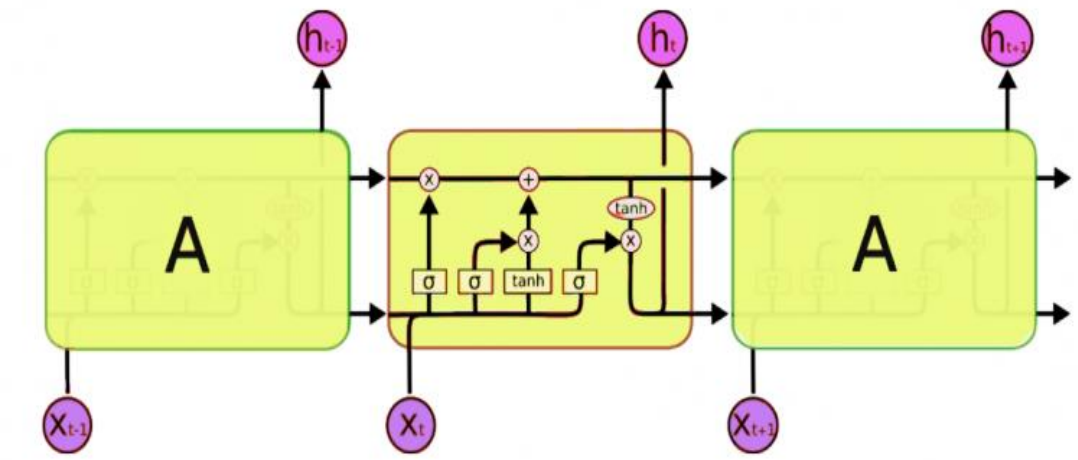
يمكن تصور أداء LSTM من خلال فهم أداء فريق القناة الإخبارية الذي يغطي قصة جريمة قتل. الآن ، قصة إخبارية مبنية على الحقائق والأدلة والبيانات من العديد من الناس. عندما يقع حدث جديد ، تتخذ أيًا من الخطوات الثلاث.

دعنا نقول ، كنا نفترض أن القتل قد تم عن طريق "تسميم" الضحية ، لكن تقرير التشريح الذي جاء للتو ذكر أن سبب الوفاة كان "تأثير على الرأس". كونك جزءًا من هذا الفريق الإخباري ، ماذا تفعل؟ تنسى على الفور سبب الوفاة السابق وجميع القصص التي تم نسجها حول هذه الحقيقة.

ماذا ، إذا تم إدخال مشتبه به جديد تمامًا في الصورة. شخص لديه ضغينة مع الضحية ويمكن أن يكون القاتل؟ قمت بإدخال هذه المعلومات في موجز الأخبار الخاص بك ، أليس كذلك؟

الآن لا يمكن تقديم كل هذه المعلومات المكسورة على وسائل الإعلام الرئيسية. لذلك ، بعد فترة زمنية معينة ، تحتاج إلى تلخيص هذه المعلومات وإخراج الأشياء ذات الصلة إلى جمهورك. ربما في شكل XYZ" تبين أنه المشتبه به الرئيسي."

ندخل الآن في تفاصيل بنية شبكة LSTM:



الشكل 1بنية LSTM

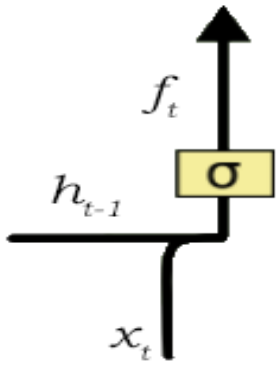
الآن ، هذا ليس قريبًا من النسخة المبسطة التي رأيناها من قبل ، لكن دعني أطلعك عليها. تتكون شبكة LSTM النموذجية من كتل ذاكرة مختلفة تسمى الخلايا (المستطيلات التي نراها في الشكل 1). هناك حالتان يتم نقلهما إلى الخلية التالية ؛ حالة الخلية والحالة المخفية. تعتبر كتل الذاكرة مسؤولة عن تذكر الأشياء ويتم التلاعب بهذه الذاكرة من خلال ثلاث آليات رئيسية تسمى البوابات. تتم مناقشة كل منهم أدناه.

- **بوابة النسيان:**

أخذ مثال مشكلة التنبؤ بالنص. لنفترض أنه تم إدخال LSTM بالجملة التالية:

Bob is a nice person. Dan on the other hand is evil.

بمجرد مواجهة أول نقطة توقف بعد "person" ، تدرك بوابة النسيان أنه قد يكون هناك تغيير في السياق في الجملة التالية. نتيجة لذلك ، يتم نسيان موضوع الجملة وإخلاء مكان الموضوع. وعندما نبدأ الحديث عن "Dan" هذا الموقف من الموضوع يتم تخصيصه لـ "Dan". عملية نسيان الموضوع هذه ناتجة عن بوابة النسيان.



الشكل 2بوابة النسيان

بوابة النسيان مسؤولة عن إزالة المعلومات من حالة الخلية. تتم إزالة المعلومات التي لم تعد مطلوبة لـ LSTM لفهم الأشياء أو المعلومات الأقل أهمية عن طريق مضاعفة عامل التصفية. هذا مطلوب لتحسين أداء شبكة. LSTM

تأخذ هذه البوابة مدخلين ؛ h_t-1 و. x_t

h_t-1 هي الحالة المخفية من الخلية السابقة أو ناتج الخلية السابقة و x_t هي الإدخال في تلك الخطوة الزمنية المحددة. يتم ضرب المدخلات المعطاة في مصفوفات الوزن ويضاف التحيز. بعد ذلك ، يتم تطبيق تابع الظل القطعي (sigmoid) على هذه القيمة. ينتج تابع الظل القطعي متجهًا ، تتراوح قيمه من 0 إلى 1 ، المقابلة لكل رقم في حالة الخلية. في الأساس، يكون تابع

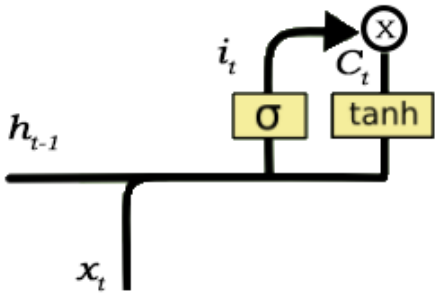
الظل القطعي مسؤول عن تحديد القيم التي يجب الاحتفاظ بها وأيها يجب التخلص منها. إذا تم إخراج "0" لقيمة معينة في حالة الخلية ، فهذا يعني أن بوابة النسيان تريد أن تنسى حالة الخلية تلك المعلومات تمامًا. وبالمثل ، يعني "1" أن بوابة النسيان تريد أن تتذكر قطعة المعلومات بأكملها. يتم ضرب هذا المتجه الناتج من تابع الظل القطعي في حالة الخلية.

- **بوابة الإدخال:**

نأخذ مثالًا آخر حيث تقوم LSTM بتحليل جملة:

Bob knows swimming. He told me over the phone that he had served the navy for 4 long years.

الآن المعلومات المهمة هنا هي أن "Bob" يعرف السباحة وأنه خدم في البحرية لمدة أربع سنوات. يمكن إضافة هذا إلى الحالة الخلوية ، ومع ذلك ، فإن حقيقة أنه أخبر كل هذا عبر الهاتف هي حقيقة أقل أهمية ويمكن تجاهلها. يمكن إجراء عملية إضافة بعض المعلومات الجديدة عبر بوابة الإدخال.



الشكل 3بوابة الإدخال

بوابة الإدخال مسؤولة عن إضافة المعلومات إلى حالة الخلية. إضافة المعلومات هذه هي في الأساس عملية من ثلاث خطوات كما يتضح من الشكل 3.

1. تنظيم القيم التي يجب إضافتها إلى حالة الخلية من خلال تضمين تابع الظل القط. هذا يشبه إلى حد كبير بوابة النسيان ويعمل كمرشح لجميع المعلومات من h_t−1 و x_t.
2. إنشاء متجه يحتوي على جميع القيم الممكنة التي يمكن إضافتها (كما يُدرك من h_t−1 و x_t) إلى حالة الخلية. يتم ذلك باستخدام تابع tanh ، والتي تنتج قيمًا من −1 إلى +1.

3. مضاعفة قيمة المرشح التنظيمي (بوابة الظل القطعي) إلى المتجه المُنشأ (تابع tanh) ثم إضافة هذه المعلومات المفيدة إلى حالة الخلية عبر عملية الإضافة.

بمجرد الانتهاء من هذه العملية المكونة من ثلاث خطوات ، نضمن إضافة هذه المعلومات فقط إلى حالة الخلية المهمة وليست زائدة عن الحاجة.

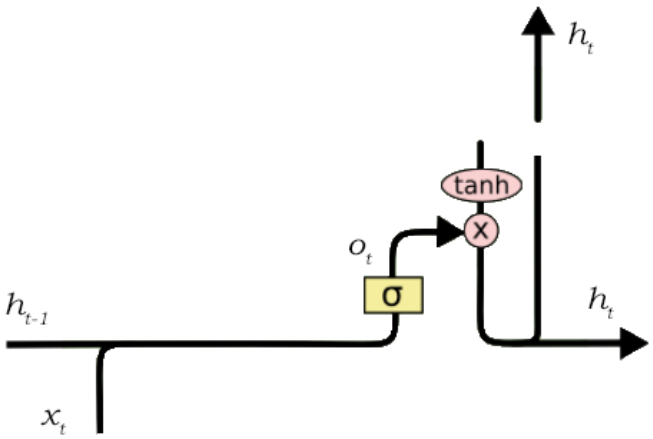
- **بوابة الخرج:**

ليست كل المعلومات التي تعمل على طول حالة الخلية مناسبة للإخراج في وقت معين. سنتخيل هذا بمثال:

Bob fought single handedly with the enemy and died for his country. For his contributions brave _____.

في هذه العبارة ، يمكن أن يكون هناك عدد من الخيارات للمساحة الفارغة. لكننا نعلم أن المدخلات الحالية لـ "brave" ، هي صفة تُستخدم لوصف اسم. وبالتالي ، أيا كانت الكلمة التالية ، لديها ميل قوي لكونها اسمًا. وبالتالي ، يمكن أن يكون Bob ناتجًا مناسبًا.

هذه المهمة هي اختيار المعلومات المفيدة من حالة الخلية الحالية وإظهارها كإخراج يتم عبر بوابة الإخراج.



الشكل 4بوابة الاخراج

يمكن تقسيم عمل بوابة الإخراج إلى ثلاث خطوات:

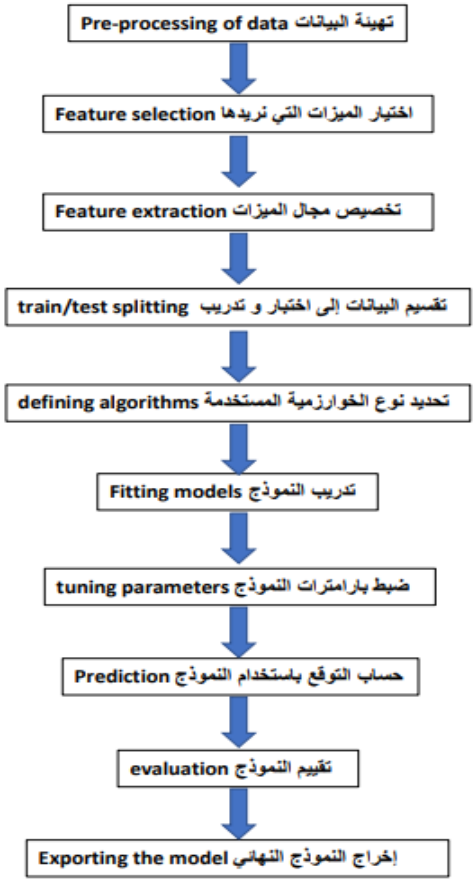
1. إنشاء متجه بعد تطبيق تابع tanh على حالة الخلية ، وبالتالي قياس القيم إلى النطاق −1 إلى +1.

2. عمل مرشح باستخدام قيم h_t−1 و x_t ، بحيث يمكنه تنظيم القيم التي يجب إخراجها من المتجه الذي تم إنشاؤه أعلاه. يستخدم هذا المرشح مرة أخرى تابع الظل القطعي.
3. مضاعفة قيمة هذا المرشح التنظيمي إلى المتجه الذي تم إنشاؤه في الخطوة 1 ، وإرساله كمخرج وأيضًا إلى الحالة المخفية للخلية التالية.

سيتأكد المرشح في المثال أعلاه من أنه يقلل من جميع القيم الأخرى باستثناء "Bob". وبالتالي ، يجب أن يُبنى المرشح على قيم الإدخال والحالة المخفية وأن يتم تطبيقه على متجه حالة الخلية.

4. المحاكاة والنتائج

1.4. سيناريو العمل



2.4. البرنامج

أولاً ، نلقي نظرة على مجموعة البيانات، العمود "open" يمثل سعر الافتتاح للسهم في ذلك اليوم في العمود "date" ، وعمود "close" هو سعر الإغلاق في ذلك اليوم . يمثل العمود "high" أعلى سعر تم الوصول إليه في ذلك اليوم ، ويمثل العمود "low" أقل سعر.

	A	B	C	D	E	F	G	H
1	Date	Open	High	Low	Close	Adj Close	Volume	
2	7/1/2010	5	5.184	4.054	4.392	4.392	41094000	
3	7/2/2010	4.6	4.62	3.742	3.84	3.84	25699000	
4	7/6/2010	4	4	3.166	3.222	3.222	34334500	
5	7/7/2010	3.28	3.326	2.996	3.16	3.16	34608500	
6	7/8/2010	3.228	3.504	3.114	3.492	3.492	38557000	
7	7/9/2010	3.516	3.58	3.31	3.48	3.48	20253000	
8	#########	3.59	3.614	3.4	3.41	3.41	11012500	
9	#########	3.478	3.728	3.38	3.628	3.628	13400500	
10	#########	3.588	4.03	3.552	3.968	3.968	20976000	
11	#########	3.988	4.3	3.8	3.978	3.978	18699000	
12	#########	4.14	4.26	4.01	4.128	4.128	13106500	
13	#########	4.274	4.45	4.184	4.382	4.382	12432500	
14	#########	4.37	4.37	4.01	4.06	4.06	9126500	
15	#########	4.132	4.18	3.9	4.044	4.044	6262500	
16	#########	4.1	4.25	4.074	4.2	4.2	4789000	
17	#########	4.238	4.312	4.212	4.258	4.258	3268000	
18	#########	4.3	4.3	4.06	4.19	4.19	4611000	
19	#########	4.182	4.236	4.052	4.11	4.11	3098500	
20	#########	4.11	4.18	4.102	4.144	4.144	2336000	
21	#########	4.154	4.176	4	4.07	4.07	3080000	
22	#########	4.04	4.088	3.91	3.988	3.988	2134500	
23	8/2/2010	4.1	4.194	4.066	4.184	4.184	3590500	

الشكل كشكل الداتا سيت

نبدأ باستيراد المكتبات:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential, load_model
from keras.layers import LSTM, Dense, Dropout
```

سنستخدم numpy للعمليات الرياضية ، وpandas لتعديل مجموعة البيانات الخاصة بنا ، و matplotlib لتصور النتائج، وتعلم كيفية توسيع نطاق بياناتنا ، و keras للعمل كغطاء في مكتبات منخفضة المستوى مثل TensorFlow أو مكتبة الشبكات العصبية عالية المستوى Theano .

نقوم بتحميل مجموعة البيانات الخاصة بنا (يتم تجاهل الكود التالي إذا كنت لا تستخدم Google Colab):

```
from google.colab import files
dataset = files.upload()
```

ثم نضغط على "choose files" التي تظهر على الشاشة عند تشغيل الكود لنقوم بتحميل مجموعة البيانات. في حال لا يتم استخدام Google Colab ، فيمكن وضع ملف البيانات في نفس مجلد الكود.

بعد تحميل بياناتنا، نحتاج إلى عمل إطار بيانات:

```
df = pd.read_csv('TSLA.csv')
```

بعد ذلك ، دعنا نحصل على عدد أيام التداول:

```
df.shape
```

سيكون خرج التعليمة (2392,7).

لتبسيط الأمر قدر الإمكان ، سنستخدم فقط متغيرًا واحدًا وهو السعر "open".

```
df = df['Open'].values
df = df.reshape(-1, 1)
```

يتيح لنا التابع reshape() إضافة أبعاد أو تغيير عدد العناصر في كل بُعد. نحن نستخدم reshape(-1,1) لأن لدينا بُعدًا واحدًا فقط في المصفوفة ، لذا سيُنشئ numby نفس عدد الأسطر ويضيف محورًا آخر 1: ليكون البعد الثاني.

الآن نقسم البيانات إلى مجموعات تدريب واختبار:

```
dataset_train = np.array(df[:int(df.shape[0]*0.8)])
dataset_test = np.array(df[int(df.shape[0]*0.8):])
```

سنستخدم MinMaxScaler لتقييس بياناتنا بين صفر وواحد.

```
scaler = MinMaxScaler(feature_range=(0,1))
dataset_train = scaler.fit_transform(dataset_train)
dataset_test = scaler.transform(dataset_test)
```

بعد ذلك ، سننشئ التابع الذي سيساعدنا في إنشاء مجموعات البيانات:

```
def create_dataset(df):
    x = []
    y = []
    for i in range(50, df.shape[0]):
        x.append(df[i-50:i, 0])
        y.append(df[i, 0])
    x = np.array(x)
    y = np.array(y)
    return x,y
```

بالنسبة للميزات (x) ، سنلحق دائمًا آخر 50 سعرًا ، وبالنسبة للجدول (y) ، سنلحق السعر التالي. ثم سنستخدم numpy لتحويله إلى مصفوفة.

سنقوم الآن بإنشاء بيانات التدريب والاختبار الخاصة بنا عن طريق استدعاء هذا التابع لكل من التدريب والاختبار(لكل واحد):

```
x_train, y_train = create_dataset(dataset_train)
x_test, y_test = create_dataset(dataset_test)
```

بعد ذلك ، نحتاج إلى إعادة تشكيل reshape() بياناتنا لجعلها مصفوفة ثلاثية الأبعاد لاستخدامها في LSTM Layer.

```
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

الخطوة التالية هي بناء النموذج :

```
model = Sequential()
model.add(LSTM(units=96, return_sequences=True, input_shape=(x_train.shape
[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=96,return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=96,return_sequences=True))
model.add(Dropout(0.2))
```

```
model.add(LSTM(units=96))
model.add(Dropout(0.2))
model.add(Dense(units=1))
```

أولاً ، قمنا بتهيئة نموذجنا كنموذج تسلسلي مكون من 96 وحدة في بعد الخرج. استخدمنا return_sequences = True لجعل طبقة LSTM بإدخال ثلاثي الأبعاد و input_shape لتشكيل مجموعة البيانات الخاصة بنا.

جعل الجزء المتسرب 0.2 أي يسقط 20٪ من الطبقات. أخيرًا ، أضفنا طبقة كثيفة بقيمة 1 لأننا نريد إخراج قيمة واحدة.

بعد ذلك ، نريد إعادة تشكيل ميزتنا لطبقة LSTM ، لأنها متسلسلة_3 3_sequentila)) والتي تتوقع 3 أبعاد ، وليس 2:

```
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))
x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))
```

الآن سنقوم تجميع نموذجنا:

```
model.compile(loss='mean_squared_error', optimizer='adam')
```

استخدمنا loss='mean_squared_error' وهو تابع حساب متوسط مربع الخطأ ، ومُحسِّن adam لتحديث أوزان الشبكة بشكل متكرر بناءً على بيانات التدريب.

نقوم بحفظ النموذج الخاص وبنا ونبدأ التدريب:

```
model.fit(x_train, y_train, epochs=50, batch_size=32)
model.save('stock_prediction.h5')
```

تشير كل فترة epoch إلى دورة واحدة من خلال مجموعة بيانات التدريب الكاملة ، ويشير حجم الدُفعة إلى عدد أمثلة التدريب المستخدمة في تكرار واحد.

نقوم بتحميل نموذجنا:

```
model = load_model('stock_prediction.h5')
```

الخطوة الأخيرة هي جعل النتائج تظهر بيانياً:

```
predictions = model.predict(x_test)
predictions = scaler.inverse_transform(predictions)
y_test_scaled = scaler.inverse_transform(y_test.reshape(-1, 1))

fig, ax = plt.subplots(figsize=(16,8))
ax.set_facecolor('#000041')
ax.plot(y_test_scaled, color='red', label='Original price')
plt.plot(predictions, color='cyan', label='Predicted price')
plt.legend()
```

3.4. النتائج

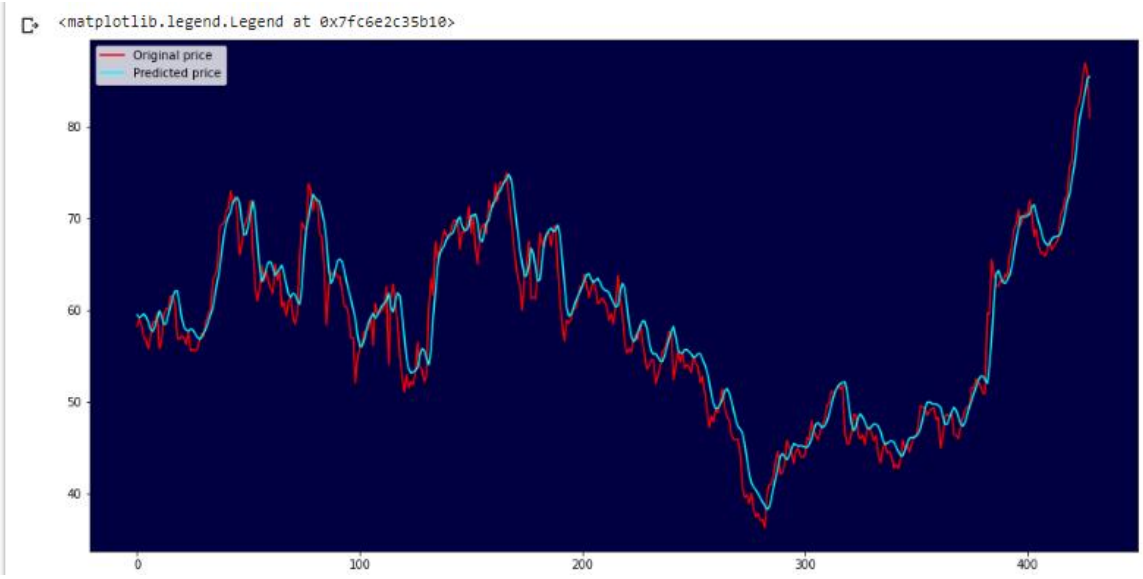
مرحلة التدريب:

```
[18] model.save('stock_prediction.h5')

Epoch 1/50
59/59 [==============================] - 19s 175ms/step - loss: 0.0144
Epoch 2/50
59/59 [==============================] - 10s 174ms/step - loss: 0.0035
Epoch 3/50
59/59 [==============================] - 10s 173ms/step - loss: 0.0031
Epoch 4/50
59/59 [==============================] - 10s 173ms/step - loss: 0.0037
Epoch 5/50
59/59 [==============================] - 10s 173ms/step - loss: 0.0028
Epoch 6/50
59/59 [==============================] - 10s 172ms/step - loss: 0.0026
Epoch 7/50
59/59 [==============================] - 10s 173ms/step - loss: 0.0028
Epoch 8/50
59/59 [==============================] - 10s 172ms/step - loss: 0.0026
Epoch 9/50
59/59 [==============================] - 10s 177ms/step - loss: 0.0024
Epoch 10/50
59/59 [==============================] - 10s 176ms/step - loss: 0.0022
Epoch 11/50
```

الشكل 6مرحلة التدريب

المخطط البياني لمقارنة السعر الحقيقي مع السعر المتوقع: يبين الشكل المخطط البياني لسعر الأسهم حيث يمثل الخط الأحمر السعر الأصلي والخط الأزرق يمثل السعر المتوقع باستخدام مشروعنا.



الشكل 7مخطط يوضح السعر الحقيقي والسعر المتوقع

5. التوصيات

تطوير البرنامج ليصبح أكثر دقة في توقع أسعار الصرف وأيضاً يوصى بوصله على منصة خاصة بالبورصة لتوقع أسعار أسهم الشركات الكبرى.

6. المراجع

1- https://blog.devgenius.io/predicting-tesla-stocks-tsla-using-python-pycaret-45af9ed47de9

2- https://www.coursera.org/projects/tesla-stock-price-prediction-facebook-prophet

3- https://www.kaggle.com/fatmakursun/tesla-stock-price-prediction

4- Learning, Aurelien Geron Hands-On Machine. "with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems." (2017).

4- https://finance.yahoo.com/quote/TSLA/history/