Projekt: Vorhersage der Tagesrendite des Goldpreises aufgrund verschiedener Faktoren

Ziel des Projekts ist es, die zukünftige prozentuale Tagesrendite des Goldpreises mit Hilfe verschiedener Regressionsverfahren vorherzusagen. Grundlage der Vorhersage sind historische Goldpreis-Daten sowie ausgewählte externe Einflussfaktoren. Ziel ist es außerdem, die wichtigsten Einflussfaktoren zu identifizieren, die den Goldpreis maßgeblich bestimmen.

1 Bestandteile der Projektabgabe

- **Datenbeschreibung und Analyse:** Durchführung in Gruppenarbeit (Lisa: Notebook, Tanja: Dokumentation, Neanad: Recherche und Codebegleitung) – enthalten im Jupyter Notebook + HTML-Datei
- **Auswertung mit Algorithmen:** Getrennte Python-Dateien und kommentiertes Jupyter Notebook
- **Ergebnisse:** Als Tabelle im Bericht integriert (Trainings-, Crossval- und Test-Scores)
- **Arbeitsbericht:** Vorliegend als Textfile (siehe unten)
- **Lessons Learned:** Von jedem Gruppenmitglied individuell erstellt

2 Inhalt des Berichtes

2.1 Teilnehmer und Arbeitsschwerpunkte

- **Lisa:** Vorbereitung der Daten im Notebook, Preprocessing
- **Tanja:** Dokumentation und Bericht, Zielwertdefinition,Visualisiewrung
- **Neanad:** Recherche zu ML-Modellen, Unterstützung bei Codierung

2.2 Klärung der Aufgabenstellung

- **Ziel:** Vorhersage der Tagesrendite des Goldpreises(prozentual)
- **Arbeitshypothese:**

Es wird angenommen, dass bestimmte makroökonomische und marktbezogene Einflussgrößen, wie beispielsweise geopolitische Risiken, Zinssätze oder Handelsvolumina, frühzeitig auf die Entwicklung des Goldpreises wirken. Daraus leitet sich die Hypothese ab, dass aus heutigen Informationen ein prädiktives Signal für die Rendite des Folgetages extrahiert werden kann.

Unsere Zielspalte ist die Differenz des Adjusted Close Preises zum vorherigen Tag in Prozent, also die tägliche Veränderung des bereinigten Schlusskurses. Das bedeutet, dass das Modell lernt, die tägliche Preisänderung vorherzusagen, statt den absoluten Preiswert.
Die Zielgröße gold_tagesrendite beschreibt die Tagesrendite von morgen (T +1)

- **Qualitätskriterien:** R2, MAE

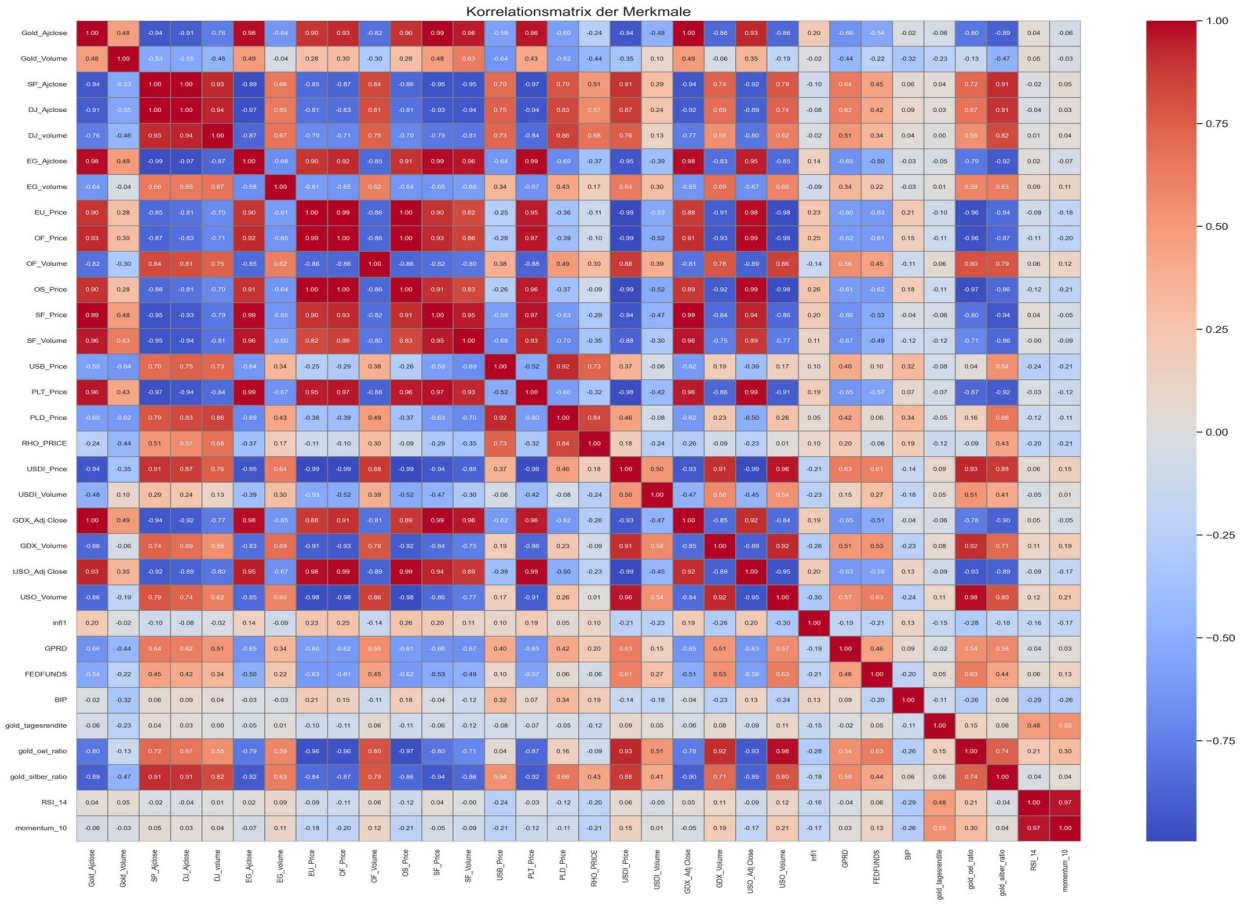
  Begründung:

- **ML-Methoden:** Die Modellierung wurde arbeitsteilig durchgeführt: **Nenad** war für die **lineare Regression** zuständig, **Lisa** übernahm die Implementierung des **Random Forest**, und **Tanja** entwickelte das Modell mit **Neuralen Netzen**.
- **Roadmap:**
  1. Daten säubern und Zielwert berechnen
  2. Modelle oberflächlich testen
  3. Crossval durchführen
  4. Gridsearch auf vielversprechende Modelle anwenden
  5. Ensemble bilden und vergleichen

2.3 Beschreibung der Daten

Die Beschreibung der einzelnen Spalten befindet sich in der Datei *Definitionen.pdf*. Weitere Informationen, insbesondere zu Einschränkungen und Besonderheiten der Datengrundlage, sind in der Datei *Einschränkungen der Daten.pdf* enthalten.

- **Datenquellen und Datensätze:**
  - **Goldpreis (Kaggle):** 1718 Zeilen, 80 Spalten. Enthält Preise, Volumen, Marktindizes, Wechselkurse und Rohstoffpreise. Grundlage der Zielwertdefinition (Tagesrendite).
  - **BIP (API-basiert) :** 1958 Zeilen, 2 Spalten. Das Bruttoinlandsprodukt wurde aus einer API abgerufen, auf Tagesebene interpoliert. Es spiegelt die wirtschaftliche Gesamtleistung wider und beeinflusst die Nachfrage nach sicheren Anlagen wie Gold.
  - **Geopolitischer Risikoindex (GPR, CSV):** 2602 Zeilen, 2 Spalten. Misst Unsicherheiten durch politische Ereignisse – steigt das Risiko, steigt oft auch der Goldpreis.
  - **Inflation (CSV):** 1614 Zeilen, mehrere Preisindizes. Zeigt die Entwicklung der Geldentwertung – hohe Inflation führt tendenziell zu einer Flucht in Sachwerte wie Gold.
  - **Zinsdaten (CSV):** 1614 Zeilen, 2–3 Spalten. Steigende Zinsen können Gold unattraktiver machen, da es keine Zinsen abwirft.
- **Beschaffung:** öffentlich zugänglich oder per API, kein Lizenzproblem
- **Datentypen:** Float-Werte, Zeitstempel, binäre Trend-Indikatoren
- **Korrelationen:** Viele Preisvarianten stark korreliert


Korrelationsmatrix der Merkmale
Gold_Ajclose
Gold_Volume
SP_Ajclose
DJ_Ajclose
DJ_volume
EG_Ajclose
EG_volume
EU_Price
OF_Price
OF_Volume
OS_Price
SF_Price
SF_Volume
USB_Price
PLT_Price
PLD_Price
RHO_PRICE
USDI_Price
USDI_Volume
GDX_Adj Close
GDX_Volume
USO_Adj Close
USO_Volume
infl1
GPRD
FEDFUNDS
BIP
gold_tagesrendite
gold_oel_ratio
gold_silber_ratio
RSI_14
momentum_10
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75

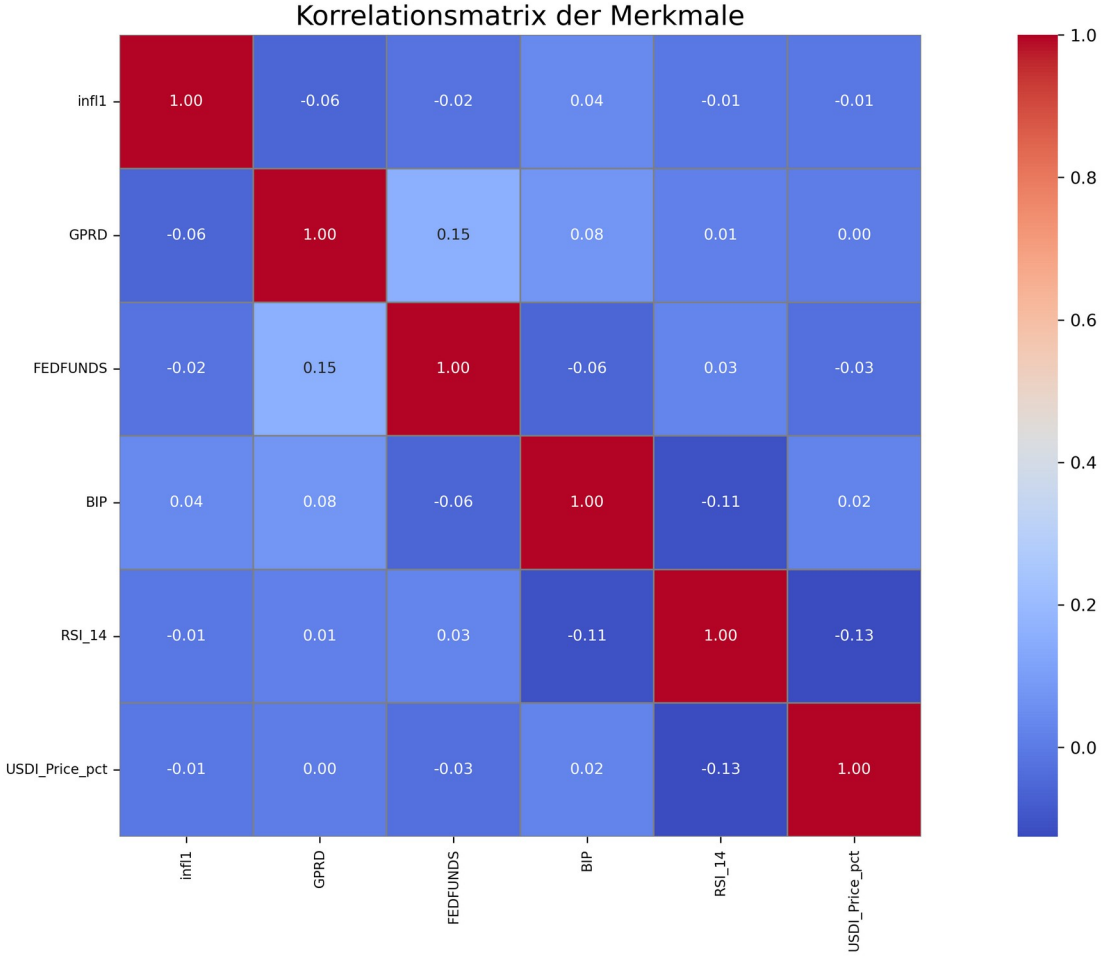

Korrelationsmatrix der Merkmale
infl1
GPRD
FEDFUNDS
BIP
RSI_14
USDI_Price_pct
1.00 -0.06 -0.02 0.04 -0.01 -0.01
-0.06 1.00 0.15 0.08 0.01 0.00
-0.02 0.15 1.00 -0.06 0.03 -0.03
0.04 0.08 -0.06 1.00 -0.11 0.02
-0.01 0.01 0.03 -0.11 1.00 -0.13
-0.01 0.00 -0.03 0.02 -0.13 1.00
1.0
0.8
0.6
0.4
0.2
0.0

Bild 1. Korrelationsmatrix post Transformation

Beobachtung:

- Viele Features redundant (z. B. Preisvarianten desselben Indexes). Bei vielen Spalten (bspw. Rohstoffpreise) sind jeweils die Werte für Open, High, Low, Close und AdjustedClose angegeben. Für jedes Asset, für das dies der Fall war, wurden Korrelationsmatrizen erstellt die zeigten, dass diese Werte stark korrelierten. Daher wurden in der Regel die Werte für AdjustedClose behalten und die restlichen Werte entfernt
- Entscheidung: nur relevante Spalten behalten, Volumen behalten, Trends teilweise entfernt
- Zusätzlich hinzugefügt wurden folgende abgeleitete Spalten:
    - **gold_tagesrendite:** Differenz der Schlusskurse zum Vortag als Zielvariable
    - **gold_silber_ratio:** Verhältnis von Gold- zu Silberpreis als Indikator für Marktverzerrungen
    - **RSI_14:** Relative Strength Index zur Identifikation überkaufter oder überverkaufter Phasen (14-Tage-Fenster)
    - **momentum_10:** Preisdynamik über ein 10-Tage-Fenster zur Einschätzung kurzfristiger Bewegungstendenzen

2.4 Beschreibung der Datenvorbereitung

- **Imputing:** NaN-Werte wurden durch den SimpleImputer(strategy='mean') ersetzt
- **Skalierung:** Die Daten wurden durch **logarithmische Skalierung** der Volumenwerte und **prozentuale Tagesveränderung** der Preiswerte transformiert, um Ausreißer zu glätten und relative Veränderungen sichtbar zu machen.
- **Zielwertberechnung:** gold_tagesrendite als Tagesrendite
- **Feature-Reduktion:** Hohe Korrelationen (r > 0.95) entfernt, keine PCA verwendet
- **Datenstruktur:** Alle Daten wurden zu täglicher Frequenz zusammengeführt und nach Datum sortiert

- *Lisa:* In dieser Transformation wird eine **logarithmische Skalierung** auf alle Spalten angewendet, die ein **Handelsvolumen** (Trading Volume) enthalten. Dazu zählt u. a. das Volumen von Gold, S&P, Dow Jones, Emerging Markets, Öl-Futures oder ETFs wie GDX und USO. Ziel ist es, **die starke Schiefe dieser Volumenwerte zu reduzieren**, da Handelsvolumina typischerweise stark rechts-schief verteilt sind. Für jede dieser Spalten ('Gold_Volume', 'SP_volume', 'DJ_volume', usw.) wird eine neue Spalte mit dem Suffix '_log' erstellt, die den natürlichen Logarithmus des ursprünglichen Werts enthält. Das Ergebnis ist ein erweiterter DataFrame merged_df, der zusätzlich zu den ursprünglichen Spalten jetzt auch deren log-transformierte Varianten enthält, was insbesondere bei der Modellierung mit linearen Algorithmen

oder neuronalen Netzen zu einer besseren Konvergenz und interpretierbaren Skalierung beitragen kann.

- *Tanja* war verantwortlich für die Entwicklung und Implementierung der Funktion **extract_price_adjclose_columns(df)**, die gezielt Preisspalten aus einem DataFrame extrahiert. Diese Funktion berücksichtigt unterschiedliche Schreibweisen und Formate von Preisinformationen, indem sie auf Begriffe wie „price“, „adj close“ und „ajclose“ prüft – unabhängig von Groß- oder Kleinschreibung sowie dem Gebrauch von Leerzeichen oder Unterstrichen. Dadurch konnte eine robuste und wiederverwendbare Lösung geschaffen werden, die auch bei uneinheitlich benannten Finanzdaten zuverlässig funktioniert.

  Ebenfalls von *Tanja* umgesetzt wurde die Funktion **add_pct_change_columns(df)**, die zu jeder numerischen Preisspalte die prozentuale Tagesveränderung berechnet und in neuen Spalten speichert. Diese Transformation bildet eine wesentliche Grundlage für finanzanalytische Anwendungen und Vorhersagemodelle. Beide Funktionen stellen damit zentrale Bausteine der Datenvorbereitung dar, die Tanja systematisch, modular und anwendungsorientiert realisiert hat.

- *Nenad:* Zur Berechnung der täglichen Rendite des Goldpreises wurde die Methode p**ct_change()** verwendet, die den prozentualen Unterschied zum Vortag ermittelt. Um jedoch eine Zielgröße zu schaffen, die für Vorhersagezwecke geeignet ist, wurde der resultierende Wert um einen Tag nach vorne verschoben **(shift(-1))**. Dadurch wird die Rendite des nächsten Tages dem aktuellen Beobachtungstag zugewiesen – was bedeutet, dass das Modell auf Basis heutiger Merkmale die Rendite von morgen vorhersagen soll. Dieses Vorgehen ist typisch für prädiktive Zeitreihenanalysen.

2.5. Anwendung verschiedener Algorithmen (alle Scores befinden sich in zwei Übersichtstabellen unter 2.8)

2.5.1. Für die Vorhersage der Zielspalte wurden die folgenden Modelle gewählt:

Neuronale Netzwerke (Tanja), Random Forest (Lisa), Ridge, Lasso und Lineare Regression (Nenad). Diese Algorithmen wurden gewählt, weil vermutet wurde, dass sie die höchste Vorhersagekraft für die Zielspalte haben.

Random Forest (RF)

Random Forest wurde als nicht-lineares Modell ausgewählt, da es komplexe Zusammenhänge und Interaktionen zwischen Merkmalen abbilden kann. Außerdem bietet es eine gewisse Robustheit gegenüber Ausreißern und Overfitting: Da jeder Baum nur mit einem zufälligen Teil des Datensatzes trainiert wird (sog. Bootstrapping), können einzelne Ausreißer nicht alle Bäume gleichzeitig beeinflussen. Durch die Aggregation vieler voneinander unabhängiger Bäume wird das Modell insgesamt stabiler**.**

Random Forest: Score Trainingsmenge

Der Trainingsscore (R2) von RF liegt bei 0.8792, was bedeutet, dass das Modell ca. 88% der Varianz der Zielvariabel im Trainingsdatensatz erklären kann. Dies spricht für ein recht gutes Fit auf den Trainingsdaten.

Der MAE des Modells ist mit 0.0023 recht klein, zudem kleiner als die Standardabweichung der gesamten Zielspalte (0.0096). Der Wert deutet darauf hin, dass das Modell die Trainingsdaten sehr gut abbildet. Keiner der beiden Werte deutet darauf hin, dass underfitting vorliegt.

Random Forest: Kreuzvalidierung

Die Kreuzvalidierung ergab folgende Ergebnisse:

- MAE: 0.0062

- R2: 0.0458

Der MAE ist mit 0.0062 recht niedrig, allerdings ist der R2 recht nahe 0 was bedeutet, dass das Modell akutell noch nicht besonders gut passt, aber immerhin etwas besser als das Mittelwert Modell ist. Aktuell liegt Overfitting vor: Der Trainingsscore ist mit 88% recht gut, das heißt, das Modell kann die Trainingsdaten gut abbilden. Allerdings kann das Modell eher schlecht mit neuen, ungesehenen Daten umgehen.

Random Forest: GridSearch

Die Grid Search hat ergeben, dass das beste Modell die folgenden Parameter besitzt:

Max_depth = 30, min_samples_leaf = 5, n_estimators = 500. Die maximale Tiefe jedes einzelnen Baumes wurde auf 30 begrenzt, was verhindert, dass Bäume zu tief wachsen und sich zu sehr an die Trainingsdaten anpassen. Jeder Blattknoten muss mindestens 5 Blätter enthalten was verhindern soll, dass das Modell zu sehr auf einzelne Ausreißer oder sehr kleine Datenmengen reagiert. Zudem wurde die Anzahl der Entscheidungsbäume auf 500 erhöht, da bei dieser Anzahl die Rechenzeit schon recht lang war, wurde auf noch größere Werte bei n_estimators verzichtet. Allerdings ist es denkbar, dass sich mit der Anzahl der Bäume auch die Genauigkeit der Voraussagen erhöht.

Der beste während der Grid Search erreichte Score ist ein negativer MAE von -0.0060. Da Scikit-Learn bei negativen Fehlermaßen wie neg_mean_absolute_error den Fehler negiert, entspricht dieser Wert einem MAE von 0.0060 — also eine leichte Verbesserung gegenüber dem ursprünglichen MAE von 0.0062.

Tanja: *Script* für Notebook → Gold_MLRegressor.html

Dokumentation & Bericht → Tanja_docs.pdf

2.8 Übersicht über alle Scores

	MAE		
	Lin. Regression	Random Forest	Neural Networks
fit	x	x	x
trainingsscore	0,0072	0,0023	0,0168
crossvalscore	0,0075	0,0062	0,0139
Suche nach optimalen Parameterwerte	x	0,0060	0,0124
Score auf Testmenge	0,0047	0,0063	0,0108

Diese Tabelle zeigt den MAE für drei verschiedene Modelle (lineare Regression (LR), Random Forest (RF)und ein neuronales Netzwerk(NN)) in jeweils drei verschiedenen Phasen: Trainingsscore, Cross-Validation, GridSearch und Testscore.

Beim Trainingsscore liegt die LR im Mittelfeld, RF weist die wenigsten Fehler auf und NN zeigt die höchsten Fehler. Auch bei der Cross-Validierung performt RF am besten. Bei der LR wurde kein GridSearch durchgeführt (x), sowohl RF als auch NN verbessern sich leicht durch den GridSearch, allerdings performt RF allgemein besser. Auf der Testmenge schneidet LR am besten ab, auch wenn RF in den anderen Phasen weniger Fehler aufwies.

Fazit:

- LR: schneidet auf der Testmenge interessanterweise am besten ab und liefert robuste Ergebnisse

- RF: sehr gute Trainings und Crossval Ergebnisse, evtl. leichtes Overfitting

- NN passt nicht gut zu diesem Datensatz – eventuell zu wenig Daten oder die Features waren zu wenig aussagekräftig, um die Zielspalte vorhersagen zu können

	R^2		
	Lin. Regression	Random Forest	Neural Networks
fit	x	x	x
trainingsscore	0,0033	0,8792	-4,6240
crossvalscore	-0,0355	0,0458	-2,6197
Suche nach optimalen Parameterwerte	x	0,2831	-2,0573
Score auf Testmenge	-0,0061	0,2185	-1,0602

Diese Tabelle zeigt die R2 Werte für die drei Modelle. Der Trainingsscore der LR bedeutet, dass dieses Modell nur 0,33 % der Varianz der Zielvariablen auf den Trainingsdaten erklären kann. RF schneidet mit fast 88% der Varianz sehr gut ab, das NN hingegen schneidet sehr schlecht ab und liegt mit einem negativen R2 deutlich schlechter als der Mittelwert. Bei der Cross-Validation schneiden LR und NN schlecht ab und erklären zum Teil deutlich weniger

als der Mittelwert. RF ist leicht positiv. Bei LR wird kein GridSearch durchgeführt, RF kann durch diese Methode deutlich besser abschneiden, NN bleibt deutlich negativ. Nur RF kann die Testmenge halbwegs sinnvoll erklären, LR liegt knapp unter dem Mittelwert und NN ist wie bei allen anderen Werten auch unter 0.

Fazit :

- LR: leider keine wirklich überzeugende Leistung
- RF: schneidet insgesamt am besten ab, das könnte mit der Funktionsweise des Algorithmus zusammenhänge: RF nimmt jede Spalte für sich, es könnte sein, dass es dadurch besser abschneidet als LR, die alle Spalten zusammen betrachtet
- NN: Extrem negativ: R^2 = –4.62 (Training), –1.06 (Test) → Modell lernt nichts Sinnvolles. Mögliche Ursachen: zu wenige Daten, schlechte Architektur, ungeeignete Features.