Paper Title\* (use style: paper title)

\*Note: Sub-titles are not captured in Xplore and should not be used

line 1: 1st Given Name Surname   
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 4th Given Name Surname  
line 2: *dept. name of organization*  
*(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 2nd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 5th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCIDline 1: 3rd Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

line 1: 6th Given Name Surname  
line 2: *dept. name of organization   
(of Affiliation)*  
line 3: *name of organization   
(of Affiliation)*line 4: City, Country  
line 5: email address or ORCID

*Abstract*—This electronic document is a “live” template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. *\*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract*. (*Abstract*)

Keywords—component, formatting, style, styling, insert (key words)

# Introduction (*Heading 1*)

Our primary responsibilities for task 1 are those related to data preprocessing, such as handling missing data, categorical data transformation, and feature extraction. In the meantime, we concentrate on feature extraction (data mining) on the auxiliary dataset to gather additional data that could enhance the model's prediction performance. A few of our data preprocessing strategies will assume that we will use tree-base models for this task. In task 2, we intend to run recommendation models and evaluate the outcomes of various solutions using the previously extracted and transformed variables from task 1.

Figure 1: raw data information

# Task I:

## Exploratory Data Analysis & Preprocessing

### Price

Figure 2.1 shows that there are two unique price records that are outside of the price distribution. For building the model, we were concerned that the results of these 2 samples would be skewed, and we assumed that such irrational property prices were unlikely to be forecast for actual use. So, we made the decision to remove these two outstanding records. The price distribution of the reset records is presented in Figure 2.2.

Graphical user interface, application, Teams

Description automatically generated

Figure 2: 1. raw data price distribution(left) 2. processed price distribution(right)

### Title

The 'title' attribute always contains the following details: “{n bed} {property type} for sale in {location}”. Even though this information may overlap with those of other variables or be irrelevant for modelling, they can still be used to impute missing values or conduct sanity check for crucial attributes like the number of beds and property type. Figure 3 compares the features taken from the title to the current fields.

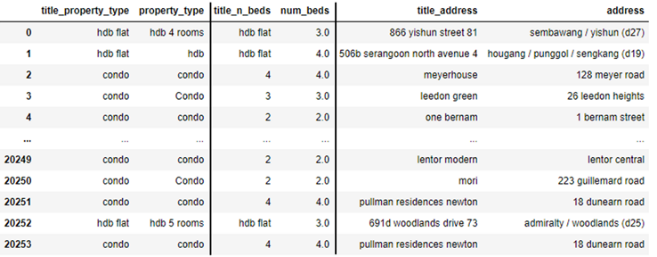


Figure 3: Features extracted from Title

### Latitude & Longitude

We primarily concentrated on the incorrect records for latitude and longitude features. Given that the goal is to forecast Singapore house property prices, the latitude and longitude ranges should be close by (1.290270, 103.851959). However, a few records are located outside the Singapore region, and further research has revealed that those records (by title, address) belong to Singapore. Since there aren't many incorrect records (Figure 4.1), we decide to use Google to find the correct location (Figure 4.2) and manually amend the entries. The results before and after the correction are shown in Figure 4.3 and Figure 4.4.

Graphical user interface, application

Description automatically generated

Figure 4: 1. unique properties with missing location (top left); 2. correct locations for imputation (bottom left) 3. before location correction (top right); 4. after location correction (bottom right)

### Property type

Transforming this property into ordinal categorical data with potential ranks is the goal of handling it. Prior to the change, we must address the issues listed below that existed in this column:

1. Uniform lower-case letters
2. Different types of HDB types/description: “hdb”, “hdb {n} rooms”, “hdb Executive”

Regarding the second issue, we chose to remove the extra "n rooms" information from the property attribute because it is already included in the num beds and num baths information. Figure 6.1 shows fewer noise property types as compared to Figure 5.

A picture containing scatter chart

Description automatically generated

Figure 5: property types and price before processing

Before converting them to a numerical representation, we decided to rank each category by its average price in consideration of improved separation and understanding by the tree-base models (Figure 6.2).

Graphical user interface, chart, application

Description automatically generated

Figure 6: 1. property types and price before processing (left); 2. average price trend given different property types (right)

### Tenure

Three main activities must be completed in order to pre-process this feature: cleaning up noisy records, handling of missing data, and categorical data conversion.

Graphical user interface, application

Description automatically generated

Figure 7: 1. conventional Singapore tenures (left); 2. tenure-price distribution before processed (right)

There are a few unrecognized tenure types in these categories when compared to the Singapore standard types of tenures (Figure 7.1) shown in Figure 7.2, which presented the raw data tenure categories. Our solution was to merge them with the neighbouring tenure categories in terms of the length of the lease.

There are 1595 missing values in this column overall, so we just placed them to a new category called "others" to handle the missing data. Additionally, we think that records falling under the "others" category can be handled by other features when utilizing a tree-based model. And the Figure 8 shows the result distribution of each category.

Since there aren't many categories in this column, we chose to apply one-hot encoding to complete the final task, categorical data conversion.

A screenshot of a computer

Description automatically generated with low confidence

Figure 8: tenure-price distribution after processed

### Built year

We only consider how to impute the missing values from this column for the "built year" attribute. With the help of other features like property type, location (latitude and longitude), and block number (extracted from title address), we used a strategy to impute the missing records as much as we could. We found a common group that might contain the unique built year and used it to impute the missing records that belong to the same group.

After the imputation, we were able to reduce the number of records missing the built year from 922 to 494. Regression algorithms like XGboost and LightBoost, which can handle the missing value by default. In the case of other regressors, we simply discard these incomplete records. The graphs in Figure 9 below demonstrate the relationship between built year and price for the top 4 types of properties by population

Graphical user interface, application

Description automatically generated

Figure 9: top 4 property types built-year-price distribution

### Num beds

The 80 missing values for the number of beds were imputed using an approach that mainly relied on two sources: the number of beds data we obtained from the title field as indicated in earlier sections and the unique value or median value of grouping by the property size in sqft. We can handle every missing value from this field using these 2 sources. Figure 10 shows that the price increases in tandem with the number of bedrooms.

Chart, box and whisker chart

Description automatically generated

Figure 10: num\_beds-price distribution

### Num baths

The "num baths" column includes 432 missing values; thus, we used a similar procedure to "num beds" to estimate the data. In order to complete this task, we decided to group data by "property type," "size sqft," and "num beds" and utilize either the unique or median value for imputation. Only 1 record remained NaN after imputation, therefore we decided to remove it. Figure 11 shows that the relationship between the number of bathrooms and the price follows a similar trend as the bedrooms.

Chart

Description automatically generated

Figure 11: num\_baths-price distribution

### Size sqft

There are multiple incorrect records under the feature "size sqft," as we discovered. To start, we use the z score to eliminate excessively big values. Two records that included "size sqft" values more than one million have been removed from Figure 12.1. In addition, some records include exceptionally small size, and we assume this is because the incorrect unit (sqm) was used. As a result, we also convert them using the ratio 10.76391 to convert from square meters to square feet (Figure 12.2). It is clear from Figure 12.3 that the size of the property has a positive relationship with the cost of the property.

Graphical user interface, chart, scatter chart

Description automatically generated

Figure 12: 1. comparison between before and after all processes (top left); 2. comparison between before and after unit correction (bottom left); 3. crrorelation bewwten size and price(right)

### Floor level

Figure 13 shows the quantity of blank values and distinct values in the "floor level" field. Since there are too many missing records, we decided to group them together under the heading "nan" for handling missing data. It's worth noting that this method is similar to how the XGboost algorithms handle Nan values by default. Additionally, we choose to separate the total level for some records from the "floor level," which will divide all values into categories in ['ground', 'low', 'mid', 'high', 'top', 'penthouse', 'nan', ‘no\_level’]. And we defined the properties such as bungalow, townhouse, and others as "no level." Then the total level specified earlier will likewise be regarded as an independent variable Figure 14.1 illustrates how the floor level affects the price of a house for the majority of properties, and Figure 14.2 demonstrates how the price of a high-floor house varies depending on the total number of building levels on the property. We choose one hot encoder for this field since it has a small number of features for category encoding.

Text

Description automatically generated

Figure 13: numbers of missing values in floor level and unique floor levels

Graphical user interface, chart, application, Teams

Description automatically generated

Figure 14: 1. overall floor\_level-price distribution (left); 2. total levels-price distribution given high floor (right)

### Furnishing

In raw data, there are total 5 categories under furnishing feature: ['unspecified' 'partial' 'unfurnished' 'fully' 'na']. After encoding, we combine the categories "unspecified" and "na" to minimize the dimensionality. Due to the small number of features after encoding and the fact that there is no obvious correlation (Figure 15) between furnishing and price, we decided to use one hot encoder for this task.

Chart, box and whisker chart

Description automatically generated

Figure 15: top 4 property types furnishing-price distribution

### available\_unit\_types

The following 4 features can be extracted from "available unit types" (Figure 16):

1. min\_br\_available: the building's smallest number of bedrooms available
2. max\_br\_available: the building's largest number of bedrooms available
3. number\_of\_types\_available: how many distinct types of units are offered by the building
4. has\_studio: Indicates whether a building has a studio-type property

Chart, box and whisker chart

Description automatically generated

Figure 16: correlation bewteen price and 4 fratures extracted from available\_unit\_types(min\_br\_available, max\_br\_available, number\_of\_types\_available, has\_studio)

### Planning area

There are 113 records in Figure 17.1 that missing planning areas, however they are all aligned with the incorrect data listed in the "Latitude & Longitude" section. As a result, we used data from earlier studies to impute the missing information, as shown in Figure 17.2.

Graphical user interface, text, application

Description automatically generated

Figure 17: 1. numbers of missing values in planning areas and unique planning areas (left); 2. Planning area correction information (right)

Chart, scatter chart

Description automatically generated

Figure 18: plainng\_area-price distribution

### Auxiliary Datasets

There are a total of 5 auxiliary datasets: shopping mall, primary school, secondary school, mrt station, and commercial center. We often employ the same technique to extract the following data from these datasets.

1. Closest interested location:

When determining the closest point of interest, we divide the area into four categories: CR(), IEBP(), BN(), and IHL(). The features for the remaining 3 datasets are, respectively, mrt station name, primary school name, secondary school name, and shopping mall name

1. Distance to the closest interested location:

The distance from the properties to the previously indicated nearest location of interest

1. Number of interested locations given radius range:

Counting the number of sites inside the circle with the hyperparameter radius R is a method similar to the DB scan method

### Others (address, subszone, Property name, property\_details\_url, elevation)

Fields like subzone and property\_details\_url are simply removed from the model building in the case of the curse of dimensionality. Additionally, we remove elevation from the training dataset because it only contains one unique value and makes no contribution to the model at all.

## Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

# Task II: Property Recommendation

In this task, we recommend similar listings from a given query listing. Such as task can be useful to users looking to purchase houses and have difficulties searching for listings that satisfy most, if not all their preferences. It would be useful if they could select certain features that are more important to them. Therefore, we aim to develop a recommendation system that suggests the most similar listings to the user’s query while providing the option of selecting certain features.

Before we dive into the implementation of our recommendation system, we first discuss the basic idea of a recommendation engine and the considerations that resulted in our implementation.

## Recommender Systems

A recommendation engine can be built in many ways. The two broad categories are content-based recommendation systems and collaborative-filtering. In content-based recommendation systems, items are recommended based on similarities with other items in the dataset. These similarities are calculated based on some concept of distance, such as cosine similarity or Manhattan distance. The limitation here is that a reference item is required. On the other hand, in collaborative filtering, recommendations for a user are made based on similarities in preferences with other users. Since only one user is of interest in this task, we use content-based filtering rather than collaborative filtering.

## Considerations

Since we are only given listings of houses in the property market and not any user information, we will work with these listings as our items. As mentioned earlier, a limitation of content-based filtering is that a reference item is required. This is not an issue for this task as a reference item will be selected from the dataset as the query item. We choose cosine similarity as the metric to calculate the similarities between the query listing and all the other listings in the dataset as it provides simplicity and accuracy [1]. The formula for cosine similarity is given by:



## Dataset

The dataset we use for this task is the pre-processed dataset with auxiliary data produced from Task 1. This dataset is appropriate for this task for several reasons. Firstly, the auxiliary data is important as data such as proximities to primary and secondary schools are very relevant considerations in Singapore’s context. Living within a 1km radius from a primary school would provide a child with higher priority in entering a primary school; and living close to secondary schools would allow for ease of commute. Secondly, the data had already converted columns with characters to ordinal encoding or one-hot encoding, which fits the requirements of this task as cosine similarity cannot be applied to characters. The only further processing done to this dataset was to fill all ‘NA’ values with -1 as null values cannot be computed in cosine similarity.

## Implementation

When generating the cosine similarity, the most naïve method would be to concatenate all the features of the query listing and calculate the cosine similarities of this vector with the concatenated features of each listing. However, doing so would lead to extremely skewed cosine similarity scores as the features with larger values would dominate the calculation. The result would be every listing having cosine similarities of 1 or 0.9999 with the query listing. Therefore, there is a need to normalize each feature. To accomplish this, we use the MinMaxScaler function from sklearn’s preprocessing library.

Once we normalize the data, the next step is to create a feature dataframe containing all the concatenated values of each row. We then use sklearn’s cosine\_similarity function to calculate the cosine\_similarities of each feature vector with the query listing’s feature vector, obtaining a column with all the cosine similarity scores.

As mentioned in III., a useful feature of this recommendation engine for users is the option to select certain features of the listing that suit their preference. For example, a user may like everything about their current query listing except for the fact that it was built in 1971. Their ideal listing would have all the same features as their query listing (i.e., same number of bedrooms, same proximity to primary schools) except that the built year of the house is later than year 2000. To implement this feature, we sieved through all the features in the dataset to come up with a selection of features that makes the most practical sense in the context of Singapore. Structuring it this way also avoids having an overly complicated argument list that confuses users. For example, it makes less sense for users to search for a house that was built before a given year since the values of older houses are significantly lower. Thus, we make it such that when the user includes ‘built\_year’ in the argument, it means that the output will contain only listings built after the given ‘built\_year’ input value. The full list of options available for selection and their input types are explained below:

* **built\_year (int):** Filters listings built during or after the input year as aforementioned.
* **max\_price (int):** Filters listing with price at most the given input. Homebuyers often have a budget in mind when searching for a house.
* **num\_beds (int):** Filters listings with at least the input number of bedrooms. If a user is looking for a house with at most 2 bedrooms, they can filter by max\_price instead since houses with fewer bedrooms tend to be cheaper.
* **pri\_sch (bool):** Default value is False. If True, filters listings with primary schools present within a radius of 1km. Parents looking to stay near primary schools are likely applying under the scheme where priority is given if they live within a 1km radius of the school.
* **sec\_sch (bool):** Default value is False. If True, filters listings with secondary schools present within a radius of 2km. There is no priority scheme for families living near secondary schools, so if users use this filter, they are likely looking for convenience in commuting to and fro the secondary schools. As such, a distance of 2km should prove to be a short enough distance that is convenient by public transport.
* **mall (bool):** Default value is False. If True, filters listings with shopping malls present within a radius of 2km. Like secondary schools, this option is more likely for convenience in accessing amenities.
* **hdb (bool):** Default value is False. If True, filters listings that are of type *hdb* or *hdb executive*, for users looking to stay in public housing.
* **private (bool):** Default value is False. If True, filters listings that are not *hdb* or *hdb executive* (i.e., private housing), for users looking to stay in private housing.

Users can customize the input options to their preferences following this given argument schema and the output data will be filtered accordingly. Once this is done, the last remaining step is to sort the data by their cosine similarity scores in descending order and select only the top *k+1* rows in the dataframe, where *k* is the number of recommendations the user wants. The top-most row is removed as this is guaranteed to be the query listing itself, with a cosine similarity of 1. This occurs as the query listing is sampled from this dataset. The resulting dataframe with *k* listings is then returned as output.

## Results

First, we select a random row from the dataset to be the query listing as shown in Fig. 1. We then do some searches to look at various results.

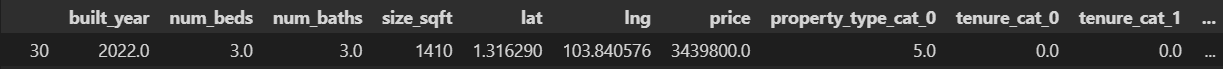
In the first search, we do a baseline search without any optional features as shown in Fig. 2. The cosine similarity scores are very high, and the features are quite similar. The full comparison can be found in the Jupyter notebook for Task 2.

In the second search, we search for similar listings that are public housing and cost less than $500,000. Public housing are *hdb* and *hdb executive*, with column *property\_type\_cat\_0* values 0 and 1. We can, therefore, deduce that the query listing is a private property. Since no private properties cost less than $500,000, the most similar listings have a considerably lower cosine similarity score compared to the baseline, as shown in Fig. 3.

In the third search, we search for similar listings that are public housing, close to primary schools and with at least 3 bedrooms, as shown in Fig. 4. Like the second search, the cosine similarity score is considerably lower than the baseline as the query listing is a private property.

In the fourth search, we search for similar listings that are private housing, close to shopping malls and cost less than $500,000, as shown in Fig. 5. There were no listings in the output as no private property costs less than $500,000.

In the last search, we search for similar listings that are public housing, built during or after 2015, close to primary and secondary schools, with at least 2 bedrooms and cost less than $1,000,000. The cosine similarity scores are the lowest compared to previous searches as newer houses tend to be more costly, thus fewer listings meet the condition of costing less than $1,000,000. Coupled with the fact that the housing type is different from the query listing, the cosine similarities are a lot lower, as shown in Fig. 6.

  
Fig. 1. Query listing

Graphical user interface, text

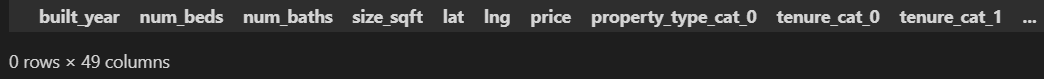
Description automatically generated  
Fig. 2. Search for the most similar listings

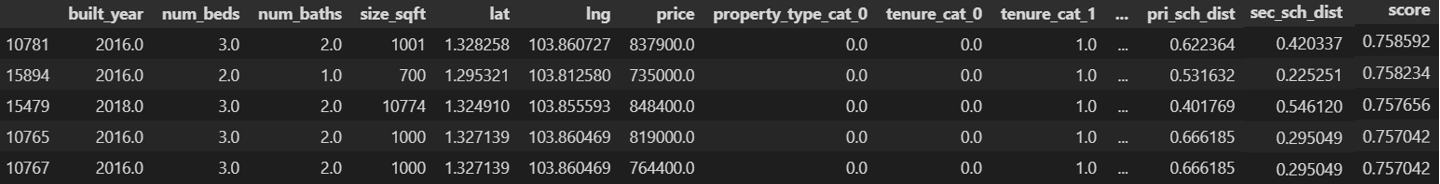
A picture containing text, scoreboard

Description automatically generated  
Fig. 3. Search for similar listings that are public housing with maximum price $500,000

Graphical user interface, text

Description automatically generated  
Fig. 4. Search for similar listings that are public housing, close to primary schools and with at least 3 bedrooms

  
Fig. 5. Search for similar listings that are private housing, close to shopping malls and cost less than $500,000

  
Fig. 6. Search for similar listings that are public housing, built during or after 2015, close to primary and secondary schools, with at least 2 bedrooms and cost less than $1,000,000

=========================================== END OF TASK 2 ==================================================

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

Identify applicable funding agency here. If none, delete this text box.

* Use a zero before decimal points: “0.25”, not “.25”. Use “cm3”, not “cc”. (*bullet list*)

## Equations

The equations are an exception to the prescribed specifications of this template. You will need to determine whether or not your equation should be typed using either the Times New Roman or the Symbol font (please no other font). To create multileveled equations, it may be necessary to treat the equation as a graphic and insert it into the text after your paper is styled.

Number equations consecutively. Equation numbers, within parentheses, are to position flush right, as in (1), using a right tab stop. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, six authors.** A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than six authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than six authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)
2. Example of a figure caption. (*figure caption*)

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. Thompson, V., Panchev, C., and Oakes, M, “Performance Evaluation of Similarity Measures on Similar and Dissimilar Text Retrieval. <https://www.scitepress.org/papers/2015/56191/56191.pdf>
2. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
3. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
4. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
5. K. Elissa, “Title of paper if known,” unpublished.
6. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
7. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
8. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published.**